TESIS DOCTORAL

Reconocimiento de lugares en entornos de exterior e interior mediante técnicas de aprendizaje profundo e información multisensorial

Juan José Cabrera Mora

2025

DIRECTOR:
Luis Payá Castelló

CODIRECTOR:
Arturo Gil Aparicio

UNIVERSIDAD MIGUEL HERNÁNDEZ DE ELCHE

Programa de Doctorado en
TECNOLOGÍAS INDUSTRIALES Y DE TELECOMUNICACIÓN

La presente Tesis Doctoral, titulada “Reconocimiento de lugares en entornos de exterior e interior mediante técnicas de aprendizaje profundo e información multisensorial”, se presenta bajo la modalidad de **tesis por compendio** de las siguientes **publicaciones**:

- **An evaluation of CNN models and data augmentation techniques in hierarchical localization of mobile robots**. J.J. Cabrera, O. J. Céspedes, S. Cebollada, O. Reinoso, L. Payá. Evolving Systems (2024). Ed. Springer-Verlag. ISSN: 1868-6486.
  DOI: https://doi.org/10.1007/s12530-024-09604-6
  Factor de impacto JCR 2024: 2.7
  Posición en el ranking JCR 2024 en la categoría ``COMPUTER SCIENCE, ARTIFICIAL INTELLIGENCE'': 111/204
  Tercer cuartil (Q3)

- **An experimental evaluation of Siamese Neural Networks for robot localization using omnidirectional imaging in indoor environments.** J.J. Cabrera, V. Román, A. Gil, O. Reinoso, L. Payá. Artificial Intelligence Review (2024). Ed. Springer. ISSN: 1573-7462.
  DOI: https://doi.org/ 10.1007/s10462-024-10840-0
  Factor de impacto JCR 2024: 13.9
  Posición en el ranking JCR 2024 en la categoría ``COMPUTER SCIENCE, ARTIFICIAL INTELLIGENCE'': 7/204
  Primer cuartil (Q1)

El Dr. Luis Payá Castelló, director, y el Dr. Arturo Gil Aparicio, codirector de la tesis doctoral titulada **"Reconocimiento de lugares en entornos de exterior e interior mediante técnicas de aprendizaje profundo e información multisensorial"**

INFORMA/N:

Que D. Juan José Cabrera Mora ha realizado bajo nuestra supervisión el trabajo titulado **"Reconocimiento de lugares en entornos de exterior e interior mediante técnicas de aprendizaje profundo e información multisensorial"** conforme a los términos y condiciones definidos en su Plan de Investigación y de acuerdo al Código de Buenas Prácticas de la Universidad Miguel Hernández de Elche, cumpliendo los objetivos previstos de forma satisfactoria para su defensa pública como tesis doctoral.

Lo que firmamos para los efectos oportunos, en …… a ……. de …….. de 2025

Director de la tesis	Codirector de la tesis
Dr. Luis Payá Castelló	Dr. Arturo Gil Aparicio

El Dr. German Torregrosa Penalva, Coordinador/a del Programa de Doctorado en **Programa de Doctorado en Tecnologías Industriales y de Telecomunicación**

INFORMA:

Que D./Dña. *"nombre y apellidos del/a estudiante"* ha realizado bajo la supervisión de nuestro Programa de Doctorado el trabajo titulado **"Reconocimiento de lugares en entornos de exterior e interior mediante técnicas de aprendizaje profundo e información multisensorial"** conforme a los términos y condiciones definidos en su Plan de Investigación y de acuerdo al Código de Buenas Prácticas de la Universidad Miguel Hernández de Elche, cumpliendo los objetivos previstos de forma satisfactoria para su defensa pública como tesis doctoral.

Lo que firmo para los efectos oportunos, en …… a ……. de …….. de 2025

Prof. Dr. German Torregrosa Penalva

Coordinador del Programa de Doctorado en Tecnologías Industriales y de Telecomunicación

Abstract

This thesis addresses the problem of place recognition in mobile robotics, a fundamental task for localization, autonomous navigation and mapping in complex and dynamic environments. An integrated approach is proposed that explores and develops robust and efficient methods based on different sensory modalities: omnidirectional cameras, LiDAR, pseudo-LiDAR and cross-modal place recognition between cameras and LiDAR.

First, visual place recognition techniques using panoramic images captured by omnidirectional cameras are studied. Two approaches are presented and analyzed: a hierarchical method based on room classification followed by a fine position estimation and a global method based on Siamese neural networks and contrastive learning. The importance of data augmentation techniques specific to panoramic images is demonstrated, improving robustness against illumination variations under real operating conditions.

Subsequently, MinkUNeXt is introduced, a new neural network architecture based on sparse 3D convolutions, optimized for place recognition from LiDAR point clouds. This architecture, together with the MinkNeXt 3D residual block, sets a new milestone in the state of the art, validated on benchmark datasets such as Oxford RobotCar and In-house.

The thesis also explores the use of pseudo-LiDAR techniques in the context of visual place recognition. The proposed technique generates synthetic point clouds from panoramic images using advanced depth estimators. The Distilled Depth Variations data augmentation technique is proposed to simulate the inaccuracies in depth estimation by combining different estimators to generate the training data for the place recognition model. In this way, the model is more robust to depth inconsistencies caused by illumination changes. The results show that robust recognition can be achieved using only visual information, reducing costs and sensory complexity.

Finally, place recognition between different sensor modalities is addressed by proposing CrossPlace, a method that transforms both 360º images captured by omnidirectional fisheye cameras and LiDAR scans into a common space of intensity, depth and semantic information. This allows the use of a single network architecture for both sensor modalities, avoiding the need to recapture databases and facilitating interoperability between heterogeneous robotic platforms. Experiments on the KITTI-360 dataset demonstrate that the proposed approach outperforms existing methods in both urban and highway scenarios.

Overall, this thesis introduces novel architectures, data augmentation techniques, and sensor fusion strategies, setting new benchmarks in place recognition and paving the way for more autonomous, flexible, and adaptable robotic systems in real-world environments.

Resumen

Esta tesis aborda el problema del reconocimiento de lugares en robótica móvil, una tarea fundamental para la localización, la navegación autónoma y el mapeo en entornos complejos y cambiantes. Se propone un enfoque integral que explora y desarrolla métodos robustos y eficientes basados en diferentes modalidades sensoriales: cámaras omnidireccionales, LiDAR, pseudo-LiDAR y reconocimiento cruzado entre cámaras y LiDAR.

En primer lugar, se estudian técnicas de reconocimiento visual de lugares utilizando imágenes panorámicas capturadas por cámaras omnidireccionales. Se presentan y analizan dos enfoques: un método jerárquico basado en la clasificación de estancias y una posterior estimación fina de la posición, y un método global basado en redes neuronales siamesas y aprendizaje por contraste. Se demuestra la importancia de técnicas de aumento de datos específicas para imágenes panorámicas, mejorando la robustez ante variaciones de iluminación en condiciones reales de operación.

Posteriormente, se introduce MinkUNeXt, una nueva arquitectura de red neuronal basada en convoluciones 3D dispersas, optimizada para el reconocimiento de lugares a partir de nubes de puntos LiDAR. Esta arquitectura, junto con el bloque residual MinkNeXt 3D, establece un nuevo hito en el estado del arte, y han sido validados en conjuntos de datos de referencia como Oxford RobotCar e In-house.

La tesis explora también el uso de enfoques pseudo-LiDAR, generando nubes de puntos sintéticas a partir de imágenes panorámicas mediante estimadores de profundidad avanzados. Se propone la técnica de aumento de datos *Distilled Depth Variations* para simular las inexactitudes en las estimaciones de profundidad al combinar diferentes estimadores para generar los datos de entrenamiento del modelo de reconocimiento de lugares. De este modo, el modelo es más robusto ante las inconsistencias de profundidad debidas a los cambios de iluminación. Los resultados muestran que es posible alcanzar un reconocimiento robusto utilizando únicamente información visual, reduciendo costes y complejidad sensorial.

Finalmente, se aborda el reconocimiento de lugares entre diferentes modalidades de sensor, proponiendo CrossPlace, un método que transforma tanto las imágenes 360º capturadas por cámaras omnidireccionales *fisheye* como las lecturas LiDAR al espacio común de la intensidad, la profundidad y la información semántica. Esto permite el uso de una única arquitectura de red para ambas modalidades de sensor, evitando la recaptura de bases de datos y facilitando la interoperabilidad entre plataformas robóticas heterogéneas. Los experimentos en el conjunto KITTI-360 demuestran que el enfoque propuesto supera a los métodos existentes tanto en escenarios urbanos como de autovía.

En conjunto, la tesis contribuye con nuevas arquitecturas, técnicas de aumento de datos y estrategias de fusión sensorial, estableciendo nuevas referencias en el reconocimiento de lugares y abriendo líneas de investigación para sistemas robóticos con mayor autonomía, flexibilidad y adaptabilidad a entornos reales.

Agredecimientos

En primer lugar, quiero agradecer a mis directores de tesis, Luis Payá y Arturo Gil, por su apoyo incondicional, orientación y paciencia durante todos estos años. Ambos me impartieron clase durante el grado y el máster, y desde entonces he aprendido mucho de ellos. Luis ha confiado siempre en mí y siento que siempre ha estado ahí para ayudarme y guiarme en mi camino como investigador. A Arturo lo conocí un poco más tarde, cuando me dio la oportunidad de desarrollar un proyecto de transferencia tecnológica. Durante este periodo, me dedicó mucho tiempo y esfuerzo, bajando a mi laboratorio prácticamente todos los días para ayudarme, formarme y guiarme. Sobre todo, me quedo con su cercanía, ya que siempre me ha tratado como un amigo y así lo he sentido.

Al principio, trabajé muy estrechamente con Sergio Cebollada, quien, junto con Luis, fue mi tutor de TFG. Por ello, le tengo especial cariño a Sergio, ya que fue el primero que me inició en el mundo de la Inteligencia Artificial, por no hablar de las risas y buenos momentos que me ha dado. Por supuesto, no me puedo olvidar de Vicente y Orlando, con quienes trabajé durante mis primeros años en el grupo. Ellos, junto con Sergio, hicieron de ARVC un lugar muy acogedor y divertido, donde aprendí mucho y disfruté de cada momento. Por otro lado, María, con quien quizás no tenía tanta relación al principio y que me ha costado un poco más conocer, se ha convertido en una de mis mejores amigas. Es una de esas personas a las que acudiría para contarle cualquier problema, ya sea personal o laboral, pues siempre sabe escuchar y dar buenos consejos. Me siento muy afortunado de haberla conocido y de poder contar con ella en mi vida.

Más tarde, el grupo fue creciendo y me siento muy afortunado de haber conocido a: Marc, Álvaro, Fran Soler, Enrique, Antonio, Paula, Esther, Marcos, Miriam y Judith. Ellos han creado un ambiente de trabajo espectacular y han pasado de ser compañeros de trabajo a ser amigos. Me siento muy afortunado de haber podido compartir tantos momentos con ellos, tanto dentro como fuera del trabajo. En especial, quiero mencionar a Antonio, con quien no solo he trabajado, sino con quien he compartido una de las mejores experiencias de mi vida: la estancia en Coimbra. Antonio y yo hemos compartido buenos y malos momentos, hemos convivido, hemos discutido y hemos reído, hemos tenido conversaciones a las tantas después de llegar de fiesta y hasta nos ha tocado ir a urgencias juntos. Con todo ello, me siento muy afortunado de haber podido compartir esta experiencia contigo. Tampoco puedo terminar sin mencionar a Marcos, a quien le tengo un cariño especial porque ha seguido mis pasos desde el principio; yo fui su tutor de TFG y, a día de hoy, no sé quién ha aprendido más de quién. Marcos es una persona muy especial para mí, con su bondad y alegría siempre ha estado ahí para ayudarme y apoyarme en todo momento. Por otro lado, aunque fuera del grupo oficial de ARVC, pero no menos importante, quiero mencionar a Fran, a quien empecé a conocer en segundo de carrera y con quien he compartido este camino desde el principio. Juntos terminamos la carrera, hicimos el máster y ahora terminamos el doctorado. Es una de esas personas que siempre ha estado ahí para mí, tanto en los buenos como en los

malos momentos. Me siento muy afortunado de tenerlo en mi vida y de poder contar con él.

Para finalizar, fuera del ámbito profesional, quiero agradecer a mi familia, pareja y amigos por su apoyo incondicional. A mis padres, que siempre han estado ahí para mí y me han dado la oportunidad de que mi única preocupación en la vida fuera estudiar, apoyándome en todo lo que he hecho. A mi hermana, que siempre ha sido un ejemplo a seguir y se ha preocupado mucho por mí. A mi iaia, aunque ya no esté, que siempre quiso que estudiara y me esforzara para conseguir una vida mejor que la que ellos tuvieron, y en eso estoy. A mi tía Susi, mi tío Antonio y mis primos Juan Antonio y Alba, que siempre me han querido y apoyado, y yo a ellos. A mi pareja Andrea, que ya van cinco años juntos y haces que mi vida sea más alegre cada día. A mi grupo de amigos Luis, David, Bri, Manu, Adrito, Carmelo y Fran, que siempre han estado ahí para hacerme reír, y ojalá la vida nos junte un poco más (geográficamente) y, sobre todo, no nos separe nunca.

Financiación

La realización de la presente tesis doctoral ha tenido lugar gracias a la beca "Ayudas para la formación de profesorado universitario" (FPU) del Ministerio de Ciencia, Innovación y Universidades del Gobierno de España, con referencia FPU21/04969. Esta beca ha permitido al doctorando dedicarse a tiempo completo a la investigación y al desarrollo de los contenidos presentados en este trabajo desde el 16 de diciembre de 2022. Además, se prevé que la fecha de defensa sea antes de comenzar la última anualidad, la cual se dedicará a un Periodo de Orientación Postdoctoral (POP), al haber defendido la tesis doctoral en tres años y cumplir con los requisitos de docencia establecidos por la FPU.

Por otro lado, el doctorando ha sido beneficiario de la beca "Ayudas complementarias de movilidad destinadas a beneficiarios del programa de Formación del Profesorado Universitario (FPU)" del Ministerio de Ciencia, Innovación y Universidades del Gobierno de España, con referencia EST23/00485. Esta ayuda de movilidad ha permitido al doctorando realizar una estancia de investigación en la Universidad de Coimbra (Portugal) durante el periodo comprendido entre el 1 de septiembre de 2023 y el 30 de noviembre de 2023, con el objetivo de realizar una estancia de investigación en el ADAI Field Tech Lab, bajo la supervisión del profesor Carlos Viegas.

Además, la presente tesis doctoral se enmarca dentro de los siguientes proyectos de investigación donde el doctorando ha conformado parte del equipo de trabajo:

- **TED2021-130901B-I00:** Desarrollo de tecnologías móviles inteligentes para tareas de seguridad y vigilancia de entornos de interior y exterior. Proyecto financiado por la Agencia Estatal de Investigación, en la convocatoria Proyectos Estratégicos orientados a la transición ecológica y digital. Investigadores principales: Luis Payá, Arturo Gil. Duración: 01/12/2022 al 30/11/2024. Financiación: 110.630,00 €.
- **PID2023-149575OB-I00:** Robótica móvil para la vigilancia automática de recintos e identificación de situaciones de riesgo en condiciones desafiantes mediante técnicas de aprendizaje profundo. Proyecto financiado por la Agencia Estatal de Investigación, en la convocatoria Proyectos de Generación de Conocimiento PID2023. Investigadores principales: Luis Payá, Arturo Gil. Duración: 01/09/2024 al 31/12/2027. Financiación: 202.375,00 €.
- **NAVISROBOT:** Localización y comprensión de la escena para la NAvegación VISual de un ROBOT móvil. Proyecto financiado por la Universidad Miguel Hernández (UMH) dentro del programa "*AYUDAS 2025 PARA PROYECTOS DE INVESTIGACIÓN*".

1 Introducción

1.1 Motivación

La presente tesis aborda uno de los desafíos fundamentales en la robótica móvil: el reconocimiento de lugares. Esta capacidad permite a los robots identificar ubicaciones específicas en el entorno, a partir de la información capturada por los sensores con los que están equipados. El reconocimiento de lugares resulta esencial para tareas como la localización, la localización y mapeo simultáneos (SLAM), la navegación autónoma y la planificación de trayectorias. Pero para comprender la importancia de la tarea abordada, es necesario definir brevemente los conceptos fundamentales de la robótica móvil. En primer lugar, la localización se refiere al acto de estimar la posición y orientación (pose) exacta del robot dentro de un mapa conocido, proporcionando coordenadas métricas precisas. En segundo lugar, el SLAM es el proceso mediante el cual un robot construye un mapa del entorno mientras simultáneamente determina su pose. Por otro lado, la exploración implica planificar los movimientos del robot con el fin de observar áreas no mapeadas anteriormente. Finalmente, la planificación de trayectorias permite determinar una secuencia óptima de movimientos que lleve al robot desde una posición inicial hasta un destino, evitando obstáculos y optimizando ciertos criterios (como la distancia recorrida o el consumo energético). En la práctica, el reconocimiento de lugares constituye frecuentemente un proceso previo a la tarea de localización, ya que permite identificar un lugar (es decir, una ubicación, zona o región de un entorno) de los anteriormente visitados. Esta identificación es crucial para la localización, ya que permite al robot reducir el espacio de búsqueda y mejorar la precisión de la estimación de su pose. Asimismo, el reconocimiento de lugares es fundamental para la detección de cierres de bucle en sistemas SLAM, lo que permite corregir inconsistencias acumuladas

en el mapa. Por tanto, se trata de un componente esencial en la mayoría de los sistemas modernos de navegación autónoma.

Los sistemas de posicionamiento global (GPS) han sido y son una herramienta fundamental para la localización en robótica móvil, proporcionando información de posición global precisa en condiciones ideales. No obstante, estos sistemas presentan limitaciones significativas en múltiples escenarios. En entornos urbanos, el efecto de cañón causado por edificios altos genera sombras de señal y reflexiones múltiples que degradan considerablemente la precisión del GPS, pudiendo introducir errores de decenas de metros. En entornos interiores, las señales de los satélites son bloqueadas por las estructuras, imposibilitando el uso del GPS. Las zonas montañosas, los terrenos accidentados y las bóvedas forestales densas presentan obstrucciones naturales que interfieren con la recepción de señales satelitales. Además, en situaciones de conflictos bélicos o interferencias electromagnéticas, los sistemas GPS pueden ser deliberadamente perturbados y/o bloqueados. En entornos subterráneos como túneles, minas o aparcamientos, el uso del GPS es directamente inviable. Incluso cuando el GPS está disponible, su precisión puede ser insuficiente para ciertas aplicaciones robóticas. Mientras que el sistema GPS proporciona precisión del orden de metros, muchas tareas de navegación requieren precisión centimétrica. Para mitigar esta limitación, existen soluciones avanzadas como los sistemas GPS/RTK (*Real-Time Kinematic*) con estación base, que permiten corregir los errores del GPS convencional mediante la transmisión de correcciones diferenciales desde una base fija a los receptores móviles. Sin embargo, estos sistemas requieren una comunicación continua y fiable entre la estación base y el receptor, lo que incrementa la complejidad operativa y puede no ser viable en entornos extensos o con obstáculos. Además, incluso con GPS/RTK, pueden aparecer ruidos y errores inesperados en zonas montañosas, cerca de edificios altos o en presencia de interferencias electromagnéticas, afectando la fiabilidad de la localización. Dado que el GPS convencional y sus variantes mejoradas no son viables en entornos de interior, se han desarrollado sistemas de posicionamiento específicos conocidos como *Indoor Positioning Systems* (IPS). En concreto, existen tecnologías como *Bluetooth Low Energy* (BLE), *Radio Frequency Identification* (RFID), *Visible Light Communication* (VLC), *Ultra Wideband* (UWB) o sistemas de posicionamiento por WiFi, que proporcionan información de posición en entornos de interior. Sin embargo, estos sistemas también enfrentan limitaciones similares a las del GPS, como la interferencia de obstáculos y la necesidad de infraestructuras específicas. Por ejemplo, los sistemas basados en BLE requieren una red de balizas distribuidas en el entorno, lo que puede ser costoso y complicado de implementar. Los sistemas basados en RFID dependen de etiquetas pasivas o activas que deben estar presentes en el entorno, lo que limita su aplicabilidad. Los sistemas VLC requieren iluminación adecuada y pueden verse afectados por cambios en la iluminación ambiental y por último, los sistemas UWB ofrecen alta precisión pero requieren hardware especializado y pueden ser sensibles a interferencias electromagnéticas. Ante estas limitaciones, resulta indispensable el desarrollo de métodos de reconocimiento de lugares basados en sensores locales, como cámaras, LiDAR (*Light Detection and Ranging*) y/o RADAR (*Radio Detection and Ranging*).

Los desafíos del reconocimiento de lugares varían significativamente según el tipo

de sensor empleado, lo que ha motivado el desarrollo de enfoques especializados. Los sistemas basados en cámaras, aunque económicos y capaces de capturar gran cantidad de información del entorno, son susceptibles a cambios de iluminación, condiciones meteorológicas y variaciones estacionales. Los sensores LiDAR, por el contrario, proporcionan información geométrica precisa e invariante a la iluminación (al ser sensores autoiluminados), pero implican mayor coste económico y computacional, además de ser sensibles a partículas suspendidas en el aire como polvo o niebla. Sin embargo, los sensores RADAR ofrecen ventajas en términos de robustez ante condiciones meteorológicas adversas y pueden complementar la información proporcionada por cámaras y LiDAR. Esta disparidad entre modalidades ha impulsado la investigación hacia soluciones que maximicen las ventajas de cada tecnología mientras minimizan sus limitaciones.

La Inteligencia Artificial (IA) ha revolucionado el reconocimiento de lugares mediante el desarrollo de arquitecturas especializadas. Las Redes Neuronales Convolucionales (CNNs) como VGG y ResNet han establecido estándares en la extracción de características visuales, mientras que los Transformers Visuales (ViTs) como DINOv2 han demostrado capacidades excepcionales mediante mecanismos de atención. Para datos 3D, PointNet estableció las bases para el procesamiento de nubes de puntos, y las convoluciones 3D dispersas en bibliotecas como Minkowski Engine han demostrado efectividad particular para datos LiDAR espacialmente dispersos. En este contexto, el aprendizaje por contraste, dado por ejemplo en las arquitecturas de siamesas, ha emergido como una técnica clave para aprender funciones de similitud entre observaciones, siendo crucial para identificar lugares únicos en entornos complejos.

Las limitaciones inherentes de cada modalidad de sensor han motivado el desarrollo de enfoques innovadores. El pseudo-LiDAR emerge como una solución prometedora que combina las ventajas económicas de las cámaras con la robustez geométrica de los datos tridimensionales. Mediante estimadores de profundidad avanzados basados en ViTs, es posible generar nubes de puntos sintéticas a partir de imágenes, proporcionando una alternativa viable cuando los sensores LiDAR no son factibles por limitaciones económicas o técnicas. Esta aproximación permite aprovechar la amplia disponibilidad de las cámaras mientras se obtiene información espacial valiosa para el reconocimiento de lugares.

Un desafío particularmente relevante en aplicaciones prácticas es el reconocimiento de lugares considerando diferentes modalidades de sensor (*cross-modal place recognition*). Este problema surge cuando el tipo de sensor utilizado para capturar la base de datos difiere del empleado durante la navegación. Esta situación es común cuando se actualizan sensores de plataformas robóticas, se requiere compatibilidad entre diferentes sistemas, o las condiciones operativas impiden el uso del sensor original. Tradicionalmente, esto obligaría a recapturar toda la base de datos con el nuevo sensor, lo cual es un proceso costoso y poco práctico. El desarrollo de métodos que permitan operar entre modalidades diferentes sin recaptura de datos representa una necesidad crítica para sistemas robóticos flexibles y adaptables.

Cuando se emplean sistemas de percepción para el reconocimiento de lugares, la

robustez ante variaciones ambientales constituye un desafío fundamental. Los entornos reales experimentan cambios dinámicos: variaciones de iluminación, modificaciones estructurales, presencia de elementos móviles como vehículos y peatones, y condiciones meteorológicas cambiantes. Los avances en técnicas de aumento de datos específicas para diferentes modalidades de sensores han demostrado ser cruciales para mejorar la robustez de los sistemas de reconocimiento de lugares. Por ejemplo, técnicas como la modificación de brillo, la adición de sombras y focos de luz han mostrado efectividad particular en condiciones de iluminación variable. Para nubes de puntos, métodos como la eliminación aleatoria de puntos y las transformaciones geométricas han probado su valor para incrementar la capacidad de generalización de los modelos. Estas técnicas permiten que los sistemas aprendan a adaptarse a condiciones cambiantes, mejorando su rendimiento en entornos y condiciones de trabajo reales.

La motivación principal de esta tesis surge de la necesidad de desarrollar métodos de reconocimiento de lugares que proporcionen autonomía sensorial completa a los sistemas robóticos, independientemente de la disponibilidad de infraestructuras de posicionamiento global. Los métodos desarrollados deben ser: (1) robustos ante variaciones ambientales, (2) eficientes computacionalmente para operación en tiempo real con recursos limitados, (3) capaces de aprovechar ventajas complementarias de diferentes modalidades de sensores, y (4) flexibles para operar con diferente modalidad de sensor sin requerir recaptura de la base de datos. Estos objetivos son fundamentales para el desarrollo de sistemas de navegación autónoma verdaderamente independientes, escalables y económicamente viables.

1.2 Objetivos

Los principales objetivos de esta tesis son abordar el reconocimiento de lugares desde una perspectiva integral, desarrollando métodos que sean robustos, eficientes y capaces de operar con diferentes modalidades de sensor. Los objetivos específicos son:

1. **Desarrollar métodos de reconocimiento visual de lugares, robustos ante variaciones de iluminación y apariencia.** El reconocimiento visual de lugares constituye una tarea fundamental en robótica móvil, enfrentándose a desafíos como cambios de iluminación, oclusiones y variaciones en el punto de vista. Para abordar estos retos, se propone:
   - Estudiar arquitecturas de redes neuronales profundas especializadas en el procesamiento de imágenes y adaptadas para llevar a cabo el reconocimiento visual de lugares por medio de cámaras omnidireccionales.
   - Desarrollar técnicas de aumento de datos específicas para imágenes con amplio campo de visión que mejoren la robustez del sistema ante condiciones cambiantes del entorno.
2. **Proponer arquitecturas eficientes para reconocimiento de lugares basado en LiDAR.** Los sensores LiDAR proporcionan información geométrica tridimensional precisa e invariante a cambios de iluminación, siendo especialmente valiosos para navegación autónoma. Para maximizar su potencial, se propone:
   - Desarrollar una arquitectura de red neuronal (MinkUNeXt) basada en con-

voluciones dispersas que mejore el procesamiento de nubes de puntos para reconocimiento de lugares.

- Realizar un análisis evolutivo desde arquitecturas existentes para la segmentación semántica (MinkUNet) hacia el diseño de red propuesto para el reconocimiento de lugares a partir de nubes de puntos (MinkUNeXt), evaluando cada modificación arquitectónica de manera sistemática.
- Optimizar el balance entre precisión y eficiencia computacional mediante el estudio de diferentes configuraciones de la red propuesta.

3. **Explorar el uso de pseudo-LiDAR para reconocimiento de lugares utilizando únicamente cámaras.** Es posible combinar las ventajas de los sensores LiDAR y las cámaras generando nubes de puntos sintéticas a partir de imágenes omnidireccionales, lo que permite reducir costes sin sacrificar robustez. Para lograrlo, se propone:
   - Investigar el desempeño de estimadores de profundidad de última generación para generar nubes de puntos a partir de imágenes panorámicas.
   - Desarrollar técnicas de aumento de datos específicas para nubes de puntos pseudo-LiDAR que mejoren el rendimiento del reconocimiento de lugares.
   - Evaluar la viabilidad de utilizar únicamente información visual para generar representaciones tridimensionales efectivas para navegación autónoma.
4. **Desarrollar un enfoque de reconocimiento de lugares entre modalidades de sensores diferentes.** En aplicaciones prácticas, es común que los tipos de sensores disponibles cambien con el tiempo, requiriendo compatibilidad entre diferentes tipos de sensor sin necesidad de recapturar bases de datos completas. Para abordar este desafío, se propone:
   - Diseñar un marco unificado que permita el reconocimiento de lugares entre sensores LiDAR y cámaras *fisheye* mediante una transformación al espacio de profundidad.
   - Implementar un método que permita la utilización de una única arquitectura de red para el reconocimiento de lugares entre diferentes modalidades de sensor.
   - Demostrar la viabilidad práctica del enfoque en escenarios reales donde se utilizan sensores diferentes para la captura de bases de datos y lecturas en tiempo real.

1.3 Estructura del documento

La presente tesis se encuentra organizada de la siguiente forma:

- **Capítulo 3: Reconocimiento de lugares basado en visión.** Este capítulo aborda el reconocimiento de lugares utilizando cámaras omnidireccionales. Se presentan dos enfoques: un método jerárquico que combina (a) un paso de reconocimiento grueso basado en la clasificación de estancias con (b) estimación fina de la posición, y un método global basado en redes neuronales siamesas. Se

analiza el impacto de diferentes técnicas de aumento de datos específicas para imágenes panorámicas, con especial énfasis en efectos visuales que pueden ocurrir en condiciones reales de operación.

- **Capítulo 4: Reconocimiento de lugares basado en LiDAR.** Se presenta MinkUNeXt, una nueva arquitectura de red neuronal basada en convoluciones 3D dispersas para el reconocimiento de lugares a partir de nubes de puntos. Se describe el diseño evolutivo desde MinkUNet hasta la arquitectura propuesta, incluyendo el desarrollo del bloque residual MinkNeXt 3D. Los experimentos demuestran mejoras significativas en precisión y eficiencia comparado con el estado del arte.
- **Capítulo 5: Reconocimiento de lugares basado en pseudo-LiDAR.** Este capítulo explora el uso de nubes de puntos sintéticas generadas a partir de imágenes panorámicas utilizando estimadores de profundidad de última generación. Se presenta un enfoque que combina las ventajas de la información visual y de la geométrica, reduciendo costes mientras mantiene la robustez ante variaciones de iluminación. Se desarrollan técnicas específicas de aumento de datos para este tipo de información pseudo-LiDAR.
- **Capítulo 6: Reconocimiento cruzado de lugares entre diferentes modalidades de sensor: LiDAR y cámaras *fisheye*.** Se aborda el desafío del reconocimiento de lugares entre diferentes modalidades de sensores (LiDAR y cámaras fisheye). Se propone CrossPlace, un método que transforma ambas modalidades al espacio común de la intensidad, profundidad e información semántica, permitiendo el uso de una única arquitectura que permite el reconocimiento mediante tipos de sensor heterogéneos. Los experimentos en KITTI-360 demuestran la viabilidad del enfoque propuesto.

1.4 Estancias internacionales

Durante el desarrollo de esta tesis, se llevaron a cabo dos estancias internacionales que contribuyeron significativamente a la investigación y al intercambio de conocimientos. Estas estancias incluyeron:

- **Estancia en la Universidad de Coimbra (Portugal):** Desde el 1 de septiembre de 2023 hasta el 30 de noviembre de 2023 (3 meses), se realizó una estancia en la Universidad de Coimbra bajo la supervisión del profesor Carlos Viegas. Durante esta estancia, se trabajó en el desarrollo de técnicas avanzadas para el reconocimiento de lugares utilizando nubes de puntos y se capturó un conjunto de datos específico en entornos forestales.
- **Estancia en la Universidad de Oxford (Reino Unido):** Desde el 1 de septiembre de 2024 hasta el 30 de noviembre de 2024 (3 meses), se llevó a cabo una estancia en la Universidad de Oxford, bajo la supervisión del profesor Daniele De Martini. Esta estancia se centró en la interpretabilidad de Transformers Visuales (ViTs) y en la manipulación de sus activaciones intermedias con el objetivo de ignorar objetos dinámicos presentes en las imágenes, con vistas a su aplicación en el reconocimiento de lugares.

1.5 Resumen de materiales, métodos, evaluación y discusión de resultados

1.5.1 Materiales

En este apartado se mencionan los materiales que se han empleado para poder llevar a cabo las investigaciones del presente trabajo:

- **Sensores de captura:** Sistemas catadióptricos omnidireccionales conformado por una cámara digital MDCS2 y un espejo hiperbólico para la captura de imágenes 360º, cámaras *fisheye* con campo de visión de 185º que conforman también un sistema de visión omnidireccional 360º, y sensores LiDAR 3D como Velodyne HDL-64E y LiDAR 2D SICK LMS-151 para captura de nubes de puntos tridimensionales.
- **Conjuntos de datos:** Bases de datos públicas especializadas en reconocimiento de lugares incluyendo COLD [1] para entornos interiores, Oxford RobotCar [2] para navegación urbana, KITTI-360 [3] para evaluación *cross-modal*, y conjuntos In-house [4] para validación específica.
- **Hardware computacional:** Estaciones de trabajo equipadas con GPU NVIDIA GeForce RTX 3090 con 24 GB de memoria para entrenamiento de redes neuronales profundas, y sistemas con CPU Intel Core i7 para preprocesamiento de datos.

1.5.2 Métodos

En este apartado se van a introducir y referenciar las principales herramientas y técnicas que se han utilizado para llevar a cabo las investigaciones de la presente tesis. Estas herramientas abarcan desde arquitecturas de redes neuronales hasta técnicas de aumento de datos, pasando por métodos de procesamiento de imágenes y algoritmos de aprendizaje.

- **Redes para procesamiento 2D:** Redes Neuronales Convolucionales (CNNs) incluyendo VGG16 [5], ResNet-101 [6], ResNeXt-101 64x4d [7], MobileNet [8], EfficientNetV2 [9] y ConvNeXt [10] adaptadas al reconocimiento visual de lugares.
- **Redes para procesamiento 3D:** Redes Neuronales con convoluciones dispersas para el tratamiento de nubes de puntos como MinkUNet [11] y la arquitectura MinkUNeXt propuesta en la presente tesis para reconocimiento de lugares basado en LiDAR.
- **Redes para la estimación de la profundidad:** Arquitecturas especializadas para la estimación de profundidad a partir de imágenes, incluyendo Depth Anything V2 [12] y Distill Any Depth [13], que utilizan como *backbone* el modelo fundacional DINOv2 [14].
- **Redes para la segmentación semántica:** Modelos como MinkUNet34C [11] y SegFormer [15] para la segmentación semántica de nubes de puntos e imagen, respectivamente.
- **Redes específicas para el reconocimiento de lugares:** CosPlace [16] para

reconocimiento visual de lugares adaptado a las diferentes modalidades de sensor.

- **Procesamiento de imágenes:** Técnicas de *inpainting* (LaMa [17]), transformaciones geométricas para generación de vistas panorámicas, y algoritmos de correspondencia de características (ORB [18], RANSAC [19]).
- **Técnicas de aprendizaje:** Aprendizaje de clasificación por medio la función de pérdida de Entropía Cruzada y aprendizaje por contraste basado en la función de pérdida *Contrastive Loss* y función de pérdida *Truncated Smooth-AP* (TSAP) para optimización de ranking, y técnicas de transferencia del conocimiento desde modelos pre-entrenados.
- **Aumento de datos:** Técnicas específicas para imágenes omnidireccionales (cambios de brillo, adición de sombras, focos de luz), métodos convencionales para nubes de puntos (eliminación de puntos, rotaciones), y técnicas novedosas propuestas en la presente tesis, como *Distilled Depth Variations* para pseudo-LiDAR.

1.5.3 Métricas de evaluación

Las métricas de evaluación utilizadas en esta tesis son fundamentales para medir el rendimiento de los métodos propuestos en el reconocimiento de lugares. En general, la literatura establece como métrica principal el *Recall at* N (R@N), que mide la capacidad del sistema para recuperar correctamente las ubicaciones relevantes en función de un umbral de distancia de d metros. Esta métrica se define como el porcentaje de instancias de test para las que al menos una de las N primeras predicciones de la base de datos se encuentra dentro de un umbral de d metros medido desde la ubicación de la instancia de test. Formalmente, se expresa como:

$$\text{R@}N = \frac{|q \in Q \mid \exists\ p \in P_q^N \text{ tal que } d(p,q) \leq d|}{|Q|} \tag{1.1}$$

donde Q es el conjunto de todas las instancias de test, P_q^N representa las N ubicaciones principales predichas para la instancia de test q, y $d(p,q)$ denota la distancia euclídea entre la ubicación predicha p y la ubicación real de la instancia test q.

En concreto, se evalúan dos variantes de esta métrica: R@1 y R@1 %. Para R@1, establecemos $N = 1$, lo que significa que una predicción se considera correcta si se encuentra dentro del umbral de d metros. Para R@1 %, definimos N como el número de imágenes o nubes de puntos correspondientes al 1 % superior de la base de datos de referencia total, permitiendo la evaluación sobre un conjunto más amplio de predicciones. Además, esta métrica se complementa con otras métricas específicas según el tipo de sensor y el enfoque utilizado.

1.5.4 Principales resultados

A continuación, se resumen y exponen los resultados obtenidos en los diferentes experimentos que se han llevado a cabo en cada capítulo de la tesis:

- **Capítulo 3: Reconocimiento de lugares basado en visión.** Se han evaluado dos enfoques principales para el reconocimiento visual de lugares utilizando

imágenes omnidireccionales: un método jerárquico basado en la clasificación de estancias seguido de una estimación fina de la posición, y un método global basado en redes siamesas y aprendizaje por contraste.

  - El enfoque jerárquico ha mostrado alta precisión en la clasificación de habitaciones y una localización efectiva dentro de cada estancia, validando la utilidad de dividir el problema en pasos grueso y fino.
  - El método global ha demostrado mayor robustez ante variaciones de iluminación y cambios de apariencia, especialmente gracias al uso de redes siamesas y técnicas de aprendizaje por contraste.
  - Se ha realizado un análisis exhaustivo del impacto de diferentes técnicas de aumento de datos, como la modificación de brillo, la adición de focos de luz, sombras y rotaciones, concluyendo que tanto los cambios globales como locales en la imagen mejoran la robustez del modelo ante condiciones reales de operación.

- **Capítulo 4: Reconocimiento de lugares basado en LiDAR.** Se presenta MinkUNeXt, una arquitectura de red neuronal basada en convoluciones 3D dispersas para el reconocimiento de lugares a partir de nubes de puntos.
  - La arquitectura propuesta se ha diseñado a partir de MinkUNet [11], un modelo originalmente diseñado para la segmentación semántica. Cada cambio propuesto en este capítulo en la arquitectura de la red mejora el desempeño de la misma para la tarea de reconocimiento de lugares. Además, se demuestra la mejora significativa del bloque MinkNeXt 3D propuesto.
  - Se ha validado la importancia de cada decisión de diseño, desde la configuración macro de la red hasta el diseño micro del bloque residual, mostrando que es posible alcanzar el estado del arte sin recurrir a arquitecturas más complejas como *Transformers*.
  - MinkUNeXt ha demostrado eficiencia computacional y robustez, superando a métodos previos en términos de *recall at 1* (R@1) y *recall at 1 %* (R@1 %).
- **Capítulo 5: Reconocimiento de lugares basado en pseudo-LiDAR.** Se explora el uso de nubes de puntos pseudo-LiDAR generadas a partir de imágenes panorámicas mediante estimadores de profundidad de última generación.
  - El enfoque propuesto transforma imágenes panorámicas en mapas de profundidad y posteriormente en nubes de puntos sintéticas (pseudo-LiDAR), permitiendo el reconocimiento de lugares con una configuración sensorial basada únicamente en cámaras catadióptricas omnidireccionales.
  - Se introduce la técnica de aumento de datos *Distilled Depth Variations*, que emplea diferentes variantes destiladas de modelos de estimación de profundidad, incluyendo estimadores menos robustos, para simular inexactitudes en la predicción de profundidad y mejorar así la resiliencia del modelo de reconocimiento de lugares ante esas inconsistencias presentes en las nubes de puntos sintéticas.
  - Se demuestra que la generación de nubes de puntos 3D a partir de imágenes

contribuye a una mayor invariancia frente a cambios de iluminación, y que la combinación de características visuales basadas en gradiente mejora la generalización del sistema.

- **Capítulo 6: Reconocimiento cruzado de lugares entre diferentes modalidades de sensor: LiDAR y cámaras *fisheye*.** Se presenta CrossPlace, un método para el reconocimiento de lugares entre modalidades de sensor heterogéneas (LiDAR y cámaras *fisheye*), transformando ambas modalidades al espacio común de la intensidad, profundidad y semántica.
  - El método permite el uso de una única arquitectura de red para ambas modalidades de sensor, eliminando la necesidad de métodos de destilación complejos y costosos, y facilitando la generación de un espacio común de características.
  - Se ha desarrollado una metodología de procesamiento para convertir tanto imágenes *fisheye* como nubes de puntos LiDAR al espacio común de la intensidad, profundidad y semántica, optimizando las representaciones y aplicando técnicas de preprocesamiento como interpolación vertical e *inpainting*.
  - Se ha demostrado la importancia de la integración de información de intensidad, profundidad y semántica para mejorar la discriminación en entornos homogéneos y repetitivos, y se ha validado que la fusión tardía de descriptores proporciona el mejor rendimiento global en el reconocimiento *cross-modal*.
  - CrossPlace ha establecido una nueva referencia en el estado del arte para el reconocimiento de lugares entre sensores heterogéneos, logrando los mejores resultados tanto en entornos urbanos como de autovía en términos de R@1, R@5 y R@20.

2

Estado del arte

En la actualidad, los robots forman parte integral en diversas aplicaciones en nuestro entorno cotidiano, aunque frecuentemente su presencia pase desapercibida. Los podemos encontrar desempeñando tareas de limpieza doméstica, sirviendo en restaurantes, operando en cadenas de producción industrial, o asistiendo en hospitales y centros de atención médica. La robótica, como campo interdisciplinario que integra ingeniería mecánica, eléctrica, electrónica, telecomunicaciones e informática, ha evolucionado desde sus orígenes en los años 60 con los primeros manipuladores industriales hacia aplicaciones más sofisticadas y autónomas.

Mientras que la robótica de manipuladores puede considerarse una tecnología consolidada en el sector industrial, los robots móviles autónomos representan todavía una tecnología emergente con aplicaciones en expansión en los sectores de servicios y social. Estos sistemas deben ser capaces de realizar tareas específicas mientras navegan de forma segura por entornos dinámicos y poco estructurados, enfrentándose a desafíos como la exploración de océanos, otros planetas, o entornos peligrosos donde la presencia humana sería inviable. En los últimos años, se ha incrementado sustancialmente el despliegue de robots autónomos móviles debido a su capacidad de automatizar procesos y realizar tareas sin poner en riesgo a personas.

Un robot móvil autónomo debe ser capaz de: mapear el entorno [20], localizarse en él [21], planificar trayectorias [22], evitar obstáculos [23] y realizar tareas específicas de forma autónoma [24]. Para lograr esto, es esencial que el robot pueda percibir su entorno, interpretar la información sensorial, y tomar decisiones basadas en esa información. La percepción e interpretación del entorno es un componente crítico que permite al robot identificar lugares [25], objetos [26], personas [27], obstáculos [28] y

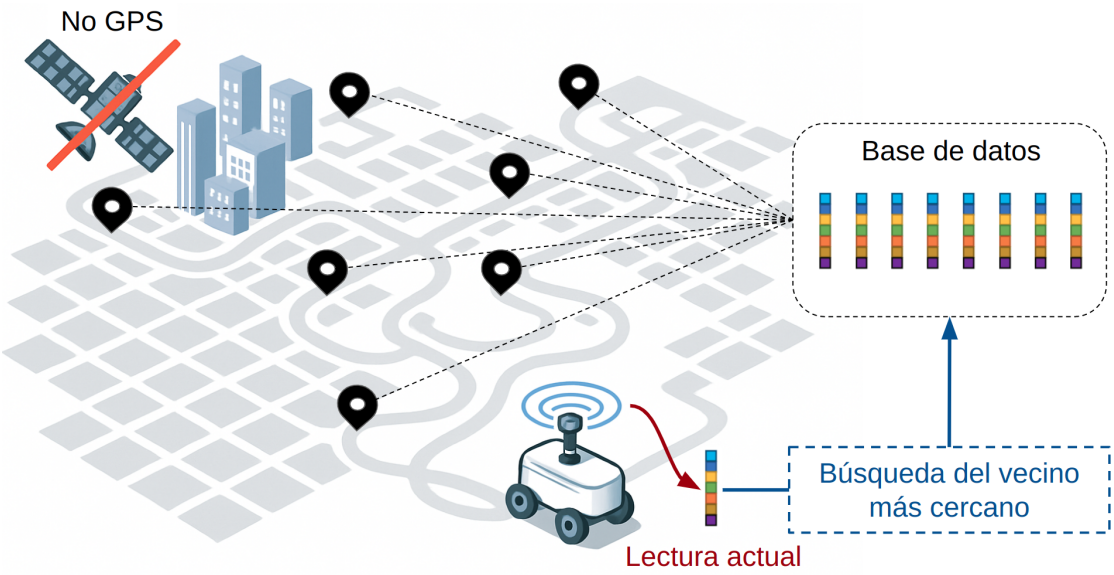


Figura 2.1: Diagrama general del proceso de reconocimiento de lugares. El robot utiliza un sensor para capturar observaciones del entorno, que son procesadas para extraer características relevantes. Estas características se comparan con una base de datos o mapa previamente almacenado para identificar si el lugar ha sido visitado anteriormente.

características relevantes del entorno [29]. El reconocimiento de lugares es una tarea fundamental en robótica móvil ya que permite a los robots identificar ubicaciones específicas en su entorno utilizando diferentes sistemas de percepción sensorial. Existen diferentes fuentes de información sensorial para realizar tareas de percepción: cámaras, sensores LiDAR, Radar y otros dispositivos de medición. A continuación, se presenta una revisión detallada de los fundamentos del reconocimiento de lugares, sus desafíos y las técnicas más relevantes en la literatura actual en función de la modalidad sensorial utilizada.

2.1 Fundamentos del reconocimiento de lugares

El reconocimiento de lugares se centra en la descripción de las observaciones obtenidas por el sensor de manera que permita al robot identificar en qué ubicación del entorno se encuentra, enfocándose en la extracción y codificación de características relevantes para comparar eficientemente la observación actual con datos almacenados previamente (base de datos o mapa). La Figura 2.1 ilustra el proceso general de reconocimiento de lugares, donde el robot captura observaciones del entorno mediante un sensor, extrae características relevantes y las compara con una base de datos o mapa previamente almacenado para identificar en qué lugar se encuentra el robot (de entre los lugares que ha visitado anteriormente). Es importante distinguir entre reconocimiento de lugares y localización: mientras que el reconocimiento de lugares tiene como objetivo identificar en qué lugar del entorno se encuentra el robot seleccionándolo de entre los lugares que ha visitado previamente, la localización estima la posición y orientación del robot dentro de un mapa conocido. En la práctica, el reconocimiento de lugares constituye frecuentemente un componente esencial del proceso de localización. Un proceso

típico de localización global consiste en dos fases: primero, identificar el lugar en el que se encuentra el robot dentro de un mapa global utilizando el reconocimiento de lugares (o también denominado localización topológica), y segundo, realizar una estimación de la posición y orientación del robot en ese lugar (localización métrica). Esta secuencia optimiza el proceso computacional, ya que el reconocimiento de lugares actúa como un filtro inicial que reduce significativamente el espacio de búsqueda. Por tanto, es común que el reconocimiento de lugares se denomine también localización topológica en la literatura, ya que permite identificar el lugar en que se encuentra el robot partiendo de un mapa habitualmente compuesto por nodos, es decir, lecturas de sensor asociadas con su ubicación. Además, el reconocimiento de lugares no sólo resulta vital para el proceso completo de localización, sino también en SLAM donde el robot debe construir un mapa del entorno mientras se localiza en él. En este caso, el reconocimiento de lugares permite identificar ubicaciones previamente visitadas (cierres de bucle) y actualizar el mapa en consecuencia. Es por ello que el reconocimiento de lugares es frecuentemente denominado con los términos localización topológica, detección de cierres de bucle o simplemente localización en la literatura.

Según el enfoque adoptado, la terminología puede variar. En la literatura, el reconocimiento de lugares aborda la tarea como una consulta a una base de datos, donde el objetivo es encontrar la entrada más similar a la observación actual. Esta propuesta se basa en la premisa de que las observaciones del entorno son representaciones de lugares específicos y que, al comparar estas representaciones, es posible identificar un lugar. En cambio, en el contexto de la localización, el enfoque se centra en estimar la pose del robot en un mapa conocido, utilizando las observaciones de los sensores. Independientemente de la terminología utilizada, el objetivo final es el mismo: conseguir que el robot disponga de información que le permita navegar de manera autónoma y eficiente en su entorno.

Estos problemas se pueden resolver con diversos tipos de sistemas de percepción, tales como cámaras [30], LiDARs [31] o Radar [32]. En la Figura 2.2 se muestran los diferentes tipos de sensores utilizados en el reconocimiento de lugares. Las cámaras, incluidas las omnidireccionales (Figura 2.2 (a) y (b)), destacan por su bajo coste y por proporcionar información visual rica y detallada, permitiendo la captura de color, textura y una visión 360º del entorno. Sin embargo, su principal limitación es la alta sensibilidad a las condiciones de iluminación, los cambios estacionales y meteorológicos, lo que puede afectar la robustez del reconocimiento. Los sensores LiDAR (Figura 2.2 (c)), por su parte, ofrecen datos tridimensionales precisos e invariantes a la iluminación al ser sensores activos, lo que los hace especialmente útiles para obtener información geométrica del entorno y mejorar la precisión en tareas de localización y mapeo. No obstante, presentan un coste económico elevado, y requieren un procesamiento más complejo de las nubes de puntos generadas. Además, son sensibles a partículas en suspensión en el aire, como polvo o humo, que pueden afectar la calidad de los datos obtenidos. Por último, los radares (Figura 2.2 (d)) son también sensores activos que emiten ondas electromagnéticas y miden el tiempo que tardan en reflejarse en los objetos del entorno. Estos sensores son menos sensibles a las condiciones de iluminación y pueden operar en condiciones climáticas adversas, lo que los hace especialmente útiles

para aplicaciones en entornos exteriores o en situaciones donde la visibilidad es limitada. Sin embargo, su resolución espacial es inferior a la de las cámaras o LiDARs, lo que limita el nivel de detalle de la representación del entorno. Además, los radares suelen proporcionar información de velocidad de los objetos detectados, lo que puede ser útil para aplicaciones de navegación y detección de obstáculos.

Además, no sólo existen enfoques basados en el uso de un único tipo de sensor, sino que también se han desarrollado enfoques multi sensor que combinan información complementaria de múltiples tipos de sensores para mayor robustez. Estos enfoques permiten aprovechar las fortalezas de cada sensor y compensar sus debilidades individuales. Por ejemplo, los sistemas que combinan cámaras y LiDAR pueden beneficiarse de la información visual rica proporcionada por las cámaras junto con la precisión geométrica de los datos LiDAR.

Por otro lado, en ocasiones se dispone de un sensor diferente para capturar la base de datos y otro para realizar las lecturas durante la navegación. A esta tarea se le denomina reconocimiento cruzado entre modalidades de sensor, que permite el reconocimiento entre diferentes tipos de información. Este enfoque es especialmente útil en aplicaciones donde se requiere flexibilidad en la elección de sensor o cuando se desea evitar la recaptura de la base de datos (o mapa) con cada cambio de sensor.

En la Sección 2.2 se revisan los enfoques más relevantes en el reconocimiento de lugares unimodal, es decir, utilizando un único tipo de sensor. En este ámbito, se detalla el estado del arte utilizando cámaras (Sección 2.2.1), LiDAR (Sección 2.2.2) y Radar (Sección 2.2.3). En la presente tesis, se dedica un capítulo completo al reconocimiento unimodal a partir de cámaras omnidireccionales (Capítulo 3) y otro capítulo al reconocimiento unimodal a partir de LiDAR (Capítulo 4). Por otro lado, en la Sección 2.3 se revisan los enfoques más relevantes en el reconocimiento de lugares enriquecido con información sintética como la estimación de la profundidad y la segmentación semántica. En esta tesis, se dedica un capítulo completo al reconocimiento de lugares enriquecido con información de profundidad a partir de imágenes omnidireccionales generando nubes de puntos pseudo-LiDAR (Capítulo 5). Además, en la Sección 2.4.1 se detallan los enfoques más relevantes en el reconocimiento de lugares multimodal, es decir, utilizando múltiples tipos de sensores. En este ámbito, se revisa el estado del arte del reconocimiento cruzado entre cámaras y LiDAR (Sección 2.4.1.1), entre cámaras y Radar (Sección 2.4.1.3) y entre LiDAR y Radar (Sección 2.4.1.2). Por último, en la Sección 2.4.2 se revisan los enfoques más relevantes en el reconocimiento cruzado entre diferentes modalidades de sensor, es decir, entre cámaras y LiDAR (Sección 2.4.2.1) o Radar y LiDAR (Sección 2.4.2.2). En esta tesis, se dedica un capítulo completo al reconocimiento cruzado entre cámaras y LiDAR (Capítulo 6).

2.2 Reconocimiento de lugares unimodal

El reconocimiento de lugares unimodal se refiere a la utilización de un único tipo de sensor para identificar ubicaciones específicas en el entorno. En la literatura se ha explorado principalmente el reconocimiento de lugares a partir de cámaras, LiDARs o

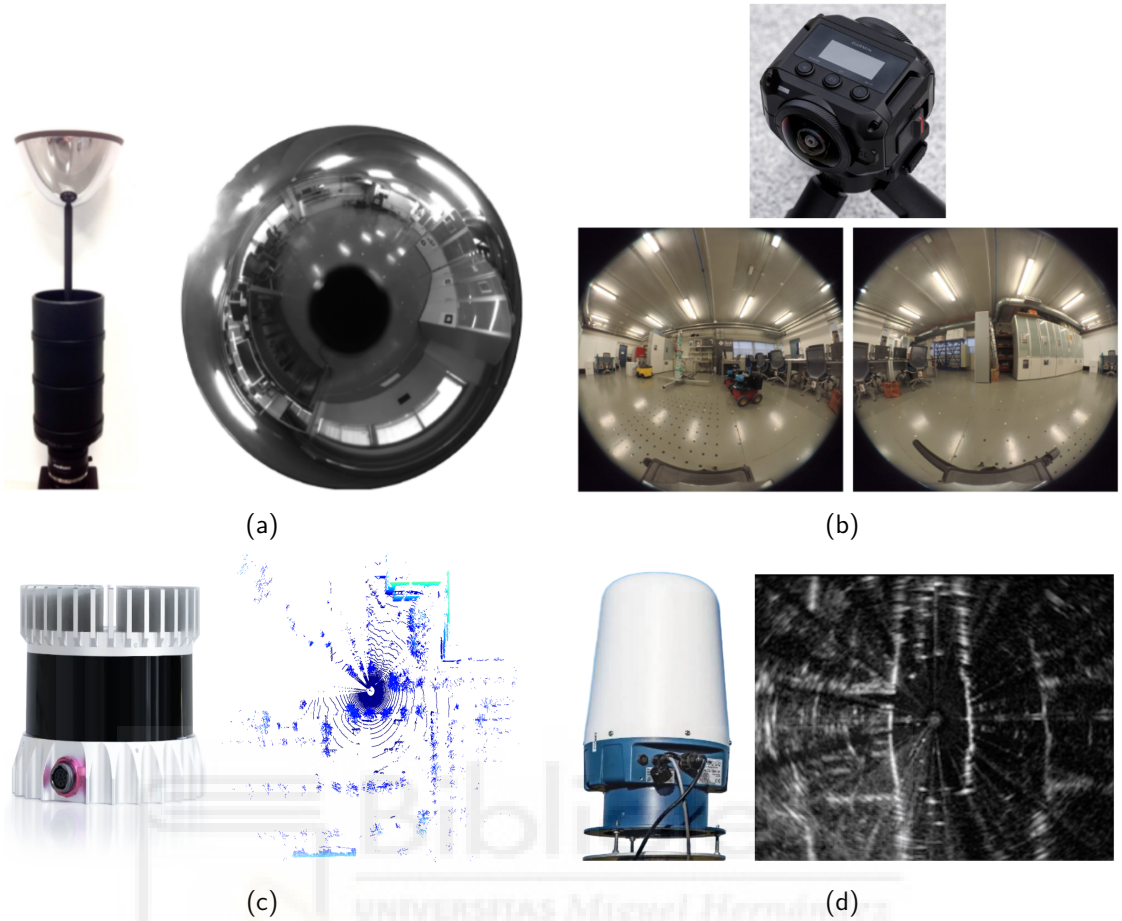


Figura 2.2: Tipos de sensores utilizados en el reconocimiento de lugares y aspecto de los datos capturados por cada uno de ellos. (a) Cámara omnidireccional catadióptrica, (b) cámara omnidireccional *fisheye*, (c) LiDAR y (d) Radar FMCW.

radares. A continuación, se presenta una revisión de los enfoques más relevantes en cada una de estas modalidades sensoriales.

2.2.1 Reconocimiento visual de lugares (VPR)

Los sistemas visuales han sido ampliamente utilizados para el reconocimiento de lugares y localización, aprovechando la información rica y detallada que proporcionan las cámaras. Estos sistemas pueden ser monoculares, estereoscópicos, omnidireccionales o fisheye, cada uno con sus propias ventajas e inconvenientes en términos de campo de visión, resolución y complejidad computacional. Además, con cada uno de ellos se han propuesto diferentes tipos de técnicas, tanto analíticas como basadas en aprendizaje automático.

2.2.1.1 Técnicas analíticas

Los primeros enfoques para el reconocimiento visual de lugares se basaron en técnicas de visión por computador clásica, utilizando descriptores locales o globales para caracterizar las imágenes. En cuanto a los descriptores locales, los métodos analíticos

SIFT (*Scale-Invariant Feature Transform*) [33] y SURF (*Speeded Up Robust Features*) [34] establecieron las bases del reconocimiento basado en características locales. Estos algoritmos detectan puntos de interés en las imágenes mediante operadores de diferencia de gaussianas o filtros de hessianas, y los describen mediante vectores de características. Estas características locales se pueden asociar entre pares de imágenes consecutivas, por ejemplo, por medio de la técnica de los cinco puntos de correspondencia [35], que permite estimar la pose relativa entre dos imágenes. Sin embargo, una solución mucho más robusta fue proporcionada por el paradigma de *Bag of Words* (BoW) [36, 37] que adaptó técnicas de procesamiento de lenguaje natural al dominio visual, revolucionando el reconocimiento de lugares en sus primeras etapas. En este enfoque, las características locales extraídas de las imágenes se cuantifican en un vocabulario visual mediante técnicas de *clustering* como *k-means*. Cada imagen se representa posteriormente como un histograma de palabras visuales, donde cada *bin* corresponde a la frecuencia de aparición de una palabra visual específica. Este enfoque que permite comparaciones eficientes entre imágenes mediante métricas de distancia estándar y ha sido ampliamente utilizado en reconocimiento de lugares, formando la base de sistemas como FAB-MAP [38]. Por otro lado, los descriptores globales, como *gist* [39], HOG (*Histogram of Oriented Gradients*) [40] o firma de Fourier, proporcionan representaciones compactas de las imágenes basadas en patrones de textura y orientación de bordes y han sido ampliamente utilizados en el contexto del reconocimiento visual de lugares [41].

2.2.1.2 Técnicas basadas en aprendizaje profundo

El reconocimiento visual de lugares basado tanto en descriptores locales como globales ha evolucionado significativamente con el surgimiento del aprendizaje profundo. En primer lugar, podemos encontrar descriptores locales tales como LIFT [42], DeLF [43], SuperPoint [44], D2Net [45] y SuperGlue [46], que utilizan redes neuronales profundas para aprender representaciones más robustas y discriminantes. Además, estos han sido ampliamente utilizados en el reconocimiento visual de lugares [43, 47]. En cuanto a la descripción global, el aprendizaje profundo ha permitido el desarrollo de arquitecturas especializadas que superan las limitaciones de los descriptores clásicos. Las primeras aproximaciones utilizaron Redes Neuronales Convolucionales (CNNs) preentrenadas para la clasificación de objetos, como AlexNet [48], adaptando sus características para reconocimiento de lugares [49]. Chen *et al.* [50] demostraron que las características aprendidas por Overfeat [51] para clasificación general podían transferirse efectivamente al reconocimiento de lugares, utilizando la técnica de *Transfer Learning*. Esta técnica resuelve dos problemas fundamentales del aprendizaje profundo: la escasez de datos de entrenamiento específicos para el reconocimiento de lugares y los altos tiempos de computación necesarios para entrenar modelos desde cero. Por ejemplo, Wozniak *et al.* [52] utilizaron los pesos de VGG-F como punto de partida para clasificar una imagen capturada por el robot entre 16 habitaciones diferentes.

Además, surgieron arquitecturas específicas para el reconocimiento de lugares, como NetVLAD [53] que marcó un hito al introducir una capa de agregación de características diferenciable, VLAD (*Vector of Locally Aggregated Descriptors*), inspirada en el enfoque de *Bag of Visual Words* (BoW). Esta técnica permite entrenar redes de

principio a fin (todas las capas son reajustadas durante el entrenamiento) produciendo descriptores globales robustos, combinando la capacidad de extracción de características de las CNNs con la agregación efectiva de las mismas. NetVLAD superó significativamente a métodos anteriores al proporcionar descriptores compactos que mantienen información discriminante para reconocimiento de lugares. Métodos más modernos han continuado esta evolución con enfoques cada vez más especializados. CosPlace [16], basado en VGG16 [5] y ResNet [6], introdujo una agregación efectiva de características y un enfoque de entrenamiento basado en la clasificación de la imagen en zonas del mapa, el cual está uniformemente dividido en regiones cuadradas. Por otro lado, EigenPlaces [54] propuso un método de entrenamiento más eficiente basado en análisis de componentes principales (PCA [55]) que reduce la dimensionalidad del espacio de características manteniendo la información más discriminante. Por otro lado, MixVPR [56] introduce una técnica de agregación holística que no requiere agrupamiento local como NetVLAD [53]. Utiliza los mapas de características extraídos de la arquitectura ResNet [6] preentrenada y aplica una cascada de bloques basados únicamente en MLPs (*Multilayer Perceptron*) para mezclar estos mapas de características, incorporando relaciones globales entre todos ellos.

Además, la adaptación de los *Transformers* (inicialmente desarrollados para el procesamiento del lenguaje natural) al dominio visual ha proporcionado nuevas capacidades para capturar dependencias entre elementos no contiguos en imágenes, a diferencia de las capas convolucionales que obtienen relaciones entre píxeles adyacentes dados por los *kernels* de convolución. En el caso de los *Vision Transformers* (ViTs) [57], la imagen de entrada se divide en "palabras" (*tokens*) y se procesan mediante mecanismos de atención, permitiendo capturar relaciones espaciales complejas. Cabe destacar que además de los *tokens* de píxeles, los ViTs también presentan un *token* global comúnmente denominado *cls token*, que representa la imagen completa. Estos modelos han demostrado un rendimiento excepcional en tareas de clasificación de imágenes y se han adaptado para reconocimiento de lugares. Además, los modelos fundacionales basados en ViTs, como DINOv2 [14], han demostrado un rendimiento sobresaliente en su aplicación al reconocimiento de lugares. Por ejemplo, AnyLoc [58] propone un enfoque universal que funciona en diversos entornos estructurados y no estructurados (urbanos, exteriores, interiores, aéreos, submarinos y subterráneos) sin necesidad de reentrenamiento. Aprovecha las características obtenidas por DINOv2 [14] y las combina con VLAD [53] y GeM (*Generalized Mean Pooling*). Este método aplica estas técnicas de agregación directamente sobre los *tokens* de los píxeles en lugar de utilizar directamente el *cls token*. Por otro lado, SALAD (*Sinkhorn Algorithm for Locally Aggregated Descriptors*) [59] reentrena DINOv2 a la vez que reformula la asignación de características locales a *clusters*, considerando tanto la similitud entre características y *clusters* como la distribución global de todas las características, logrando una agregación más informativa y robusta para el descriptor global de la imagen.

Además de los métodos basados únicamente en descriptores globales o locales, los enfoques de *re-ranking* se han vuelto fundamentales para mejorar la precisión del reconocimiento visual de lugares. El *re-ranking* es una técnica de refinamiento del proceso de reconocimiento de lugares que consta de dos etapas: primero se realiza

una búsqueda inicial utilizando descriptores globales para seleccionar un conjunto de candidatos (típicamente los *top-K* más similares), y posteriormente se utilizan unos descriptores zonales de la imagen para reordenar estos candidatos y mejorar la precisión del emparejamiento. Esta estrategia permite equilibrar eficiencia computacional con precisión, ya que el análisis detallado se aplica únicamente a un subconjunto reducido de candidatos seleccionados en lugar de toda la base de datos. Patch-NetVLAD [60] fue pionero al utilizar esta técnica de *re-ranking* al generar descriptores regionales a partir de subdivisiones de la imagen por medio de NetVLAD. Además, R2former [61] proporciona un enfoque entrenado de principio a fin para calcular puntuaciones de *re-ranking*, integrando el proceso completo en una única arquitectura. TransVPR [62] extrae conjuntamente descriptores tanto a nivel de parches como globales mediante la agregación de las capas de atención de múltiples niveles de *Vision Transformers*. Dada una imagen de entrada, primero se extraen descriptores de parches mediante una CNN de cuatro capas convolucionales y se introducen como *tokens* de entrada en un ViT. Las atenciones de las capas de entrada, intermedias y finales del *Transformer* se fusionan para generar un descriptor global de la escena por medio del *cls token*. Por otro lado, SelaVPR [63] introduce una adaptación híbrida global-local de DINOv2, un adaptador aplicado al ViT que permite extraer simultáneamente descriptores globales (para la recuperación inicial) y locales (para el *re-ranking*) sin necesidad de reentrenar el modelo completo (únicamente las capas que componen el adaptador). Así pues, la adaptación global se realiza insertando adaptadores tras las capas de atención y en paralelo a las MLPs de cada bloque Transformer, mientras que la adaptación local se implementa mediante capas convolucionales traspuestas que permiten obtener mapas de características locales detallados. Por último, Pair-VPR [64] está compuesto por una arquitectura *encoder-decoder* basada en ViTs. Este modelo se pre-entrena primero para la reconstrucción de imágenes. Para ello se seleccionan pares de imágenes y, para cada par, se enmascara fuertemente una de las imágenes. El objetivo de la red es reconstruir la imagen enmascarada. Posteriormente, se reentrenan los pesos del *encoder* y *decoder* para la tarea de reconocimiento de lugares. En específico, el *encoder* se emplea para generar un descriptor global y seleccionar a los *top-k* vecinos más cercanos mientras que el *decoder* se utiliza para refinar la selección de vecinos.

La Tabla 2.1 resume los principales métodos de reconocimiento visual de lugares que se han mencionado anteriormente, destacando su año de publicación, tipo (local, global o híbrido), arquitectura utilizada y método de agregación de características. Esta tabla proporciona una visión general de la evolución y diversidad de enfoques en el campo del reconocimiento visual de lugares.

2.2.1.3 Desafíos y robustez

Los desafíos del reconocimiento visual de lugares en entornos reales incluyen cambios de apariencia debido al transcurso del tiempo (iluminación, estación), diferencias de punto de vista, y la necesidad de generalizar a áreas desconocidas. Para abordar estos retos han surgido enfoques tales como SeqNet [65], que emplea descriptores secuenciales aprendidos para comparar secuencias de imágenes en lugar de instancias individuales, siguiendo el enfoque iniciado por SeqSLAM [66]. Estos métodos secuenciales son especialmente útiles en escenarios con cambios de apariencia extremos donde

Método	Año	Tipo	Arquitectura	Agregación
Chen *et al.* [50]	2014	Global	OverFeat	FC layers
Sunderhauf *et al.* [49]	2015	Global	AlexNet	FC layers
NetVLAD [53]	2016	Global	VGG16	VLAD
DeLF [43]	2017	Local	CNN	PCA
Wozniak *et al.* [52]	2018	Global	VGG	FC layers
CosPlace [16]	2022	Global	VGG16 / ResNet	L2 + GeM + Linear + L2
EigenPlaces [54]	2023	Global	VGG16 / ResNet	L2 + GeM + Linear + L2
MixVPR [56]	2023	Global	ResNet	MLP-Mixer
AnyLoc [58]	2023	Global	DINOv2	VLAD y GeM (sobre *tokens*)
SALAD [59]	2024	Global	DINOv2 reentrenado	Optimal Transport
Patch-NetVLAD [60]	2021	Híbrido	VGG16	VLAD sobre parches
TransVPR [62]	2022	Híbrido	CNN + ViT	*cls token*
R2Former [61]	2023	Híbrido	ViT	*cls token* + FC layer
SelaVPR [63]	2024	Híbrido	DINOv2 + *Adapter*	L2 + GeM
Pair-VPR [64]	2025	Híbrido	ViT *encoder-decoder*	FC layers

Tabla 2.1: Comparativa de métodos para reconocimiento visual de lugares (VPR).

el reconocimiento basado en imágenes individuales puede fallar.

Un aspecto crucial para aplicaciones robóticas reales es la eficiencia computacional. En este sentido, VPRTempo [67] mejora la eficiencia del reconocimiento mediante el uso de redes neuronales de impulsos, inspiradas en el sistema nervioso humano, que procesan la información de manera más eficiente que las redes neuronales tradicionales. Estas redes permiten un procesamiento rápido y de bajo consumo energético, lo que las hace adecuadas para plataformas con recursos limitados. Además, Ferrarini *et al.* [68] propusieron FloppyNet, una red neuronal binaria con reducción de profundidad y ajuste de red para VPR, optimizando el rendimiento en dispositivos con recursos limitados.

La estimación de la incertidumbre es otro componente esencial para la navegación robótica segura. Tradicionalmente, la incertidumbre en robótica se ha asociado a la estimación de la posición y orientación del robot dentro de un mapa, reflejando el grado de confianza en la localización métrica. Sin embargo, en el ámbito del reconocimiento de lugares, la incertidumbre se refiere principalmente a la confianza en la correspondencia entre la observación actual y los lugares almacenados en la base de datos. Esta información es especialmente relevante para evitar falsas correspondencias (falsos positivos) que pueden derivar en errores críticos en tareas como la detección de cierres de bucle en SLAM o la inicialización de la localización global. Por ejemplo, Cai *et al.* [69] formularon el problema de incertidumbre como la estimación de la distribución de codificaciones dentro del espacio métrico, proponiendo una red estudiante-maestro donde la red estudiante es mejorada con la varianza del maestro. Finalmente, para navegación a largo plazo y adaptación a nuevos entornos, Dasong *et al.* [70] plantearon dos funciones de pérdida de aprendizaje continuo para evitar que el modelo olvide el conocimiento previo al adaptarse a un nuevo dominio.

2.2.2 Reconocimiento de lugares basado en LiDAR

Los sensores LiDAR proporcionan nubes de puntos tridimensionales que contienen información geométrica precisa del entorno, pero requieren representaciones específicas para su procesamiento eficiente. Existen dos enfoques principales para manejar estos datos: el procesamiento directo de los puntos 3D y la proyección al plano imagen 2D.

En cuanto a las proyecciones 2D, estas incluyen múltiples variantes con características específicas según la aplicación. Las proyecciones esféricas mantienen la información angular completa del sensor [71, 72] y las proyecciones en planta (Bird's Eye View, BEV) proporcionan una vista cenital que es intuitiva para navegación [73, 74]. Cada una tiene ventajas específicas: las proyecciones esféricas conservan toda la información del sensor y las BEV simplifican la interpretación espacial y son menos sensibles a variaciones en altura.

Por otro lado, los puntos 3D se pueden tratar en crudo como una secuencia desordenada de datos para su posterior procesamiento directo mediante *Multi Layer Perceptrons* (MLPs) y funciones de agregación simétricas como *max pooling*, como se propuso en PointNet [75]. Sin embargo, este enfoque no captura adecuadamente las relaciones espaciales locales entre puntos, lo que es crucial para el reconocimiento de lugares en entornos con estructuras detalladas. Por ello, se han desarrollado técnicas más avanzadas que preservan estas relaciones espaciales y permiten el uso de arquitecturas convolucionales, como las convoluciones 3D dispersas. Para ello, las nubes de puntos se deben convertir a una representación regular, como vóxeles, que son pequeñas celdas cúbicas que dividen el espacio tridimensional en una cuadrícula uniforme y permiten representar la ocupación del espacio de forma estructurada. Esta representación voxelizada puede ser procesada eficientemente por convoluciones 3D dispersas [11] las cuales se definirán en detalle en el Capítulo 4.

2.2.2.1 Técnicas analíticas

Los primeros enfoques para el reconocimiento de lugares basado en LiDAR se centraron en técnicas analíticas que extraían características locales de las nubes de puntos [76]. En cuanto al uso de descriptores globales a partir de información 3D, se propone por primera vez en [77], donde emplean histogramas del gradiente a partir de imágenes de rango. En este contexto, Wang *et al.* [78] introdujeron LiDAR Iris, una descripción global para nubes de puntos basada en la obtención de una firma binaria. Esta representación se obtiene mediante el filtrado LoG-Gabor y operaciones de umbralización. Además, Kim *et al.* [79] diseñaron Scan Context, una representación polar que captura la distribución espacial de puntos en un formato invariante a rotaciones, permitiendo un reconocimiento eficaz incluso con cambios de orientación del sensor. Incluso también se han desarrollado descriptores análogos a BoW calculados a partir de datos LiDAR, como DBoW [80], que calculan características ORB locales a partir de la imagen de intensidad del LiDAR y las agrupan en un vocabulario visual o BoW3D [81] que utiliza descriptores locales 3D obtenidos por medio de LinK3D [82] para crear un vocabulario visual 3D, permitiendo una representación más rica de las nubes de puntos. Estos enfoques analíticos proporcionaron una base sólida para el reconocimiento de lugares

basado en LiDAR, aunque con limitaciones en términos de robustez y capacidad de generalización a entornos complejos.

2.2.2.2 Técnicas basadas en aprendizaje profundo

Tal y como se ha introducido anteriormente, PointNet [75] estableció las bases para el procesamiento directo de nubes de puntos mediante el uso de capas totalmente conectadas. Su innovación principal consistió en demostrar que las nubes de puntos pueden procesarse directamente sin necesidad de voxelización o proyección, utilizando una arquitectura de MLP seguida de una función de agregación como *max pooling*. Su capacidad para manejar datos no ordenados y su invariancia natural a permutaciones lo convirtieron en el fundamento de muchos métodos posteriores para procesamiento de nubes de puntos. Así pues, PointNetVLAD [4] representó la primera aproximación específica para reconocimiento de lugares basado en aprendizaje y tratamiento de las nubes en crudo, combinando la capacidad de extracción de características de PointNet [75] con la agregación efectiva de NetVLAD [53]. Esta arquitectura procesa nubes de puntos para extraer características locales y luego las agrega en un descriptor global mediante la capa NetVLAD adaptada a datos 3D. Aunque fue pionero en su campo, mostró limitaciones evidentes debido a la falta de información de vecindad que es crucial para la extracción de características. Estas limitaciones llevaron al desarrollo de métodos más avanzados que preservan las relaciones espaciales locales entre puntos, como PointNet++ [83], que introdujo un procesamiento jerárquico de captura de características a múltiples escalas mediante la aplicación recursiva de PointNet en regiones locales cada vez más grandes.

En relación con los métodos que parten de una nube de puntos voxelizada, Minkowski Engine [11] revolucionó el procesamiento de este tipo de datos al proporcionar una implementación eficiente con convoluciones 3D dispersas que son capaces de manejar el número irregular y estructura dispersa de las nubes de puntos. En este contexto, MinkLoc3D [84] se ha convertido en un método de referencia en reconocimiento de lugares basado en LiDAR utilizando una arquitectura *Feature Pyramid Network* (FPN) con convoluciones 3D dispersas. La arquitectura FPN permite capturar características a múltiples escalas mediante conexiones laterales entre diferentes niveles de resolución, proporcionando tanto información local detallada como contexto global.

Por otro lado, la adaptación de mecanismos de atención a datos tridimensionales ha abierto nuevas posibilidades para el reconocimiento de lugares basado en LiDAR. NDT-Transformer [85] fue pionero en aplicar *Transformers* a nubes de puntos para reconocimiento de lugares, utilizando la *Normal Distribution Transform* (NDT) como etapa de preprocesamiento para crear una representación regular que puede ser procesada efectivamente por mecanismos de atención. PPT-Net [86] introdujo una estructura piramidal que combina *Transformers* con procesamiento jerárquico para capturar características a múltiples escalas. TransLoc3D [87] estableció un nuevo estándar utilizando arquitecturas Transformer puras para reconocimiento de lugares, demostrando que los mecanismos de atención pueden capturar efectivamente las dependencias espaciales complejas en nubes de puntos.

Las técnicas basadas en la representación de nubes de puntos en 2D han ganado popularidad debido a su simplicidad y eficiencia. Las proyecciones esféricas se han utilizado ampliamente como entrada de Redes Neuronales Convolucionales (CNNs) para reconocimiento de lugares. Por ejemplo, en OverlapNet [72] se propone una red siamesa modificada para estimar la similitud entre pares de imágenes de rango LiDAR, utilizando un novedoso método de etiquetado en función del solapamiento (*overlap*) entre cada par de imágenes de rango. La arquitectura consta de dos ramas convolucionales idénticas y ligeras que procesan además de la propia imagen de rango, otros datos como las normales, la intensidad y la categoría semántica de cada punto. OverlapTransformer [88] amplía el enfoque de OverlapNet incorporando un módulo Transformer para procesar imágenes de rango LiDAR, logrando descriptores globales a rotación. La arquitectura consta de un codificador de imágenes de rango (basado en OverlapNet), seguido de un bloque Transformer que captura relaciones espaciales globales mediante mecanismos de autoatención, y una capa NetVLAD junto con MLPs para comprimir las características en un descriptor global eficiente. Además, Li *et al*. [71] introducen RangePlace, una red jerárquica basada en Transformers para el reconocimiento de lugares a partir de imágenes de rango LiDAR. Su arquitectura combina un *Swin Transformer* [89], que captura dependencias locales y globales mediante atención de ventanas desplazadas y convoluciones *depth-wise*, con una *Feature Pyramid Network* (FPN) que extrae mapas de características a múltiples escalas. Estos mapas se agregan mediante un módulo *Pyramid Feature Mix*, diseñado específicamente para fusionar información multi-escala en un descriptor global robusto. Por último, BEVPlace [73] explora el reconocimiento de lugares a partir de imágenes *Bird's Eye View* (BEV), generadas proyectando la nube de puntos LiDAR al plano del suelo. Esta representación resulta especialmente robusta frente a cambios de punto de vista y traslaciones, ya que las transformaciones en la nube de puntos se traducen en cambios mínimos en la imagen BEV. BEVPlace utiliza una red basada en convoluciones grupales [90] para extraer características locales invariantes a rotación y NetVLAD para la agregación global, logrando descriptores robustos y eficientes. Además, en BVMatch [74] se introduce el descriptor BVFT (*Bird's Eye View Feature Transform*), basado en filtros Log-Gabor, que es invariante a rotaciones e intensidades. El reconocimiento de lugares se realiza mediante un enfoque *Bag-of-Words* (BoW) sobre los descriptores BVFT, mientras que la estimación de la pose relativa 2D se lleva a cabo usando RANSAC y refinamiento posterior con ICP. BVMatch unifica así el reconocimiento de lugares y la estimación de pose.

En la Tabla 2.2 se presenta una síntesis de los principales desarrollos en reconocimiento de lugares basado en LiDAR, destacando su año de publicación, tipo de entrada (puntos 3D, proyección polar, imagen de rango o proyección BEV), arquitectura utilizada y método de agregación de características.

2.2.2.3 Desafíos y robustez

Los sensores LiDAR sufren especialmente en presencia de partículas en suspensión como polvo [92], lluvia o nieve [93], que pueden afectar la calidad de los datos y la precisión del reconocimiento de lugares. En cuanto a la navegación en entornos desconocidos, métodos como InCloud [94] han introducido enfoques de aprendizaje incremental, con funciones de pérdida innovadoras diseñadas para mantener la estruc-

Método	Año	Tipo de entrada	Arquitectura	Agregación
PointNetVLAD [4]	2018	Puntos 3D	PointNet	NetVLAD
Scan Context [79]	2018	Proyección polar	–	Descriptor 2D
LPD-Net [91]	2019	Puntos 3D	GNN (Graph Neural Net)	NetVLAD
LiDAR Iris [78]	2020	Representación polar	–	Descriptor 2D
OverlapNet [72]	2020	Imagen de rango	CNN	MLP
BVMatch [74]	2021	Proyección BEV	Descriptores BVFT	BoW
MinkLoc3D [84]	2021	Vóxeles 3D	FPN	GeM
TransLoc3D [87]	2021	Puntos 3D	Transformer	VLAD
NDT-Transformer [85]	2021	Puntos 3D (vía NDT)	Transformer	VLAD + MLP
OverlapTransformer [88]	2022	Imagen de rango	CNN + Transformer	MLP + VLAD + MLP
BoW3D [81]	2022	Puntos 3D	Descriptores LinK3D	BoW
BEVPlace [73]	2023	Proyección BEV	CNN grupal	NetVLAD
RangePlace [71]	2024	Imagen de rango	Swin Transformer + FPN	Pyramid Feature Mix

Tabla 2.2: Resumen de métodos de reconocimiento de lugares basados en LiDAR, ordenados por año

tura de *embedding* durante la adaptación a nuevos conjuntos de datos. Esto permite que los sistemas de reconocimiento mejoren continuamente con la exposición a nuevos entornos sin olvidar lo aprendido previamente, una capacidad esencial para robots que operan durante periodos prolongados en entornos cambiantes.

2.2.3 Reconocimiento de lugares basado en Radar

Los sensores de Radar (*Radio Detection and Ranging*) ofrecen una capacidad destacada para operar en condiciones adversas como niebla, lluvia intensa o nieve, donde los sensores ópticos o LiDAR pierden eficacia.

Además, existen principalmente dos modalidades de sensores radar en robótica móvil: los radar de giro mecánico (*Frequency Modulated Continuous Wave*, FMCW), que generan mapas cartesianos 2D de intensidad, y los radares de un solo chip (mmWave), que proporcionan nubes de puntos 3D con información de velocidad y *Radar Cross Section* (RCS). El RCS es una medida que representa la energía con la que un objeto refleja las ondas electromagnéticas generadas por el radar, proporcionando información sobre el material, la forma y el tamaño de este. Los radares de giro mecánico no son capaces de proporcionar información de velocidad, pero ofrecen una resolución angular más alta y son más adecuados para aplicaciones de mapeo y localización. Por otro lado, los radares mmWave proporcionan nubes de puntos 3D con información de velocidad, lo que los hace ideales para aplicaciones que requieren detección de movimiento y seguimiento de objetos.

2.2.3.1 Métodos analíticos

Los métodos analíticos para el reconocimiento de lugares con radar se centran en la extracción de características a partir de los datos de intensidad de las imágenes polares o cartesianas obtenidas a partir de los escaneos con sensores Radar FMCW. En este contexto, se han propuesto diversas técnicas para generar descriptores globales que capturan la información espacial y de intensidad de los escaneos. Por ejemplo, Scan Context [79] fue adaptado a datos radar en [95], donde se generaron histogramas pola-

res robustos frente a rotaciones, permitiendo el reconocimiento de lugares sin necesidad de aprendizaje profundo a partir de imágenes polares. Además, se han desarrollado descriptores basados en la transformada de Radon [96] y la Transformada de Fourier para obtener descriptores globales radar a partir de imágenes cartesianas. Open-RadVLAD [97] aplica la representación VLAD sobre imágenes polares de radar, junto con FFT 1-D para invariancia rotacional. Por último, ReFeree [98] introduce un descriptor global basado en el espacio libre de los escaneos radar que combina características del entorno y del espacio libre (zonas sin lecturas) de la representación polar del radar, logrando invariancia rotacional y buen rendimiento.

2.2.3.2 Enfoques de aprendizaje profundo

En cuanto a las técnicas aplicadas a radares FMCW, Saftescu *et al.* [99] y Barnes *et al.* [100] introducen CNNs sobre imágenes polares de radar, permitiendo extracción automática de características y mejora de la robustez frente a variaciones de perspectiva y ruido. Además, LookAroundYou [101] propone una red que procesa secuencias de imágenes polares de radar para mejorar la robustez frente a cambios de perspectiva y condiciones ambientales. kRadar++ [102] plantea una estrategia jerárquica basada en NetVLAD [53] e imágenes cartesianas estimando primero el lugar y después la pose, combinando eficiencia computacional con alta precisión en entornos reales. Estas CNNs consiguen adaptar exitosamente técnicas tradicionales de visión al dominio radar. Además, Komorowski *et al.* [103] proponen una *Feature Pyramid Network* (FPN) para procesar las imágenes polares del radar, inspirándose en su propio modelo MinkLoc3D [84], pero adaptado a imágenes 2D. RadarLCD [104] propone un *pipeline* supervisado para detección de cierre de bucle usando radar FMCW, integrando lo mejor de HERO (seguimiento no supervisado con U-Net) [105] y LCDNet [106]. Además, Bayesian Radar CosPlace [107] no solo identifica la ubicación más probable, sino que también estima directamente la incertidumbre asociada a dicha predicción. Aquí se adopta un modelo de clasificación que predice una distribución gaussiana para cada lugar, permitiendo rechazar consultas poco fiables. Aplicado esto sobre CosPlace [16] se demuestra una mejor estimación de la incertidumbre y una mejora en precisión.

En cuanto a los radar mmWave, AutoPlace [108] integra información Doppler para filtrar objetos móviles y una red secuencial (LSTM), mejorando significativamente la estabilidad y precisión en entornos urbanos dinámicos. Además, hacen uso de la medida *Radar Cross Section* (RCS) que proporcionan los radar mmWave que indica las propiedades de reflexión de los objetos del entorno. Por su parte, mmPlace [109] demuestra que un radar mmWave de bajo coste montado sobre una plataforma giratoria puede alcanzar reconocimiento eficaz sin la complejidad de sistemas más caros. SPR [110] propone una arquitectura basada en convoluciones de punto de kernel fijo (KPConv) diseñada específicamente para reconocimiento de lugares a partir de un único *scan* 3D de radar, sin depender de entradas adicionales de odometría. Este método también aprovecha tanto las coordenadas de los puntos como la información RCS. El método TransLoc4D [111] combina convoluciones dispersas y capas *Transformer* para datos radar 4D (x, y, z, velocidad). Proponen un *backbone* MinkLoc4D inspirado en MinkLoc3D [84] para procesar datos geométricos, de intensidad y velocidad, y posteriormente en las capas del *Transformer* aplican atención lineal para capturar dependencias espaciales,

Método	Año	Tipo de Radar	Entrada	Arquitectura/Descriptor
Radar Scan Context [95]	2020	FMCW	Imagen polar	Scan Context
Kidnapped Radar [99]	2020	FMCW	Imagen polar	VGG16 + VLAD
kRadar++ [102]	2020	FMCW	Imagen Cartesiana	NetVLAD
LookAroundYou [101]	2020	FMCW	Secuencia de imágenes polares	VGG16 + VLAD
MinkLocRadar [103]	2021	FMCW	Imagen polar	FPN
AutoPlace [108]	2022	mmWave	Nube de puntos (BEV)	LSTM
Raplace [96]	2023	FMCW	Imagen cartesiana	Radon + Discrete Fourier Transform (DFT)
Open-RadVLAD [97]	2024	FMCW	Imagen polar	VLAD + FFT
ReFeree [98]	2024	FMCW	Imagen polar	Descriptor espacio libre
mmPlace [109]	2024	mmWave	Nube de puntos	CNN
SPR [110]	2024	mmWave	Nube de puntos	KPConv + VLAD
TransLoc4D [111]	2024	mmWave	Nube de puntos	MinkLoc4D + Transformer
RadarLCD [104]	2024	FMCW	Imagen cartesiana	U-Net + NetVLAD
4D RadarPR [113]	2025	mmWave	Nube de puntos	Atención espacial + GeM + FC
Bayesian Radar CosPlace [107]	2025	FMCW	Imagen cartesiana	CosPlace
TDFANet [112]	2025	mmWave	Secuencia de nube de puntos	Deformable Pyramid Aggregation + GeM

Tabla 2.3: Comparativa de métodos para reconocimiento de lugares basados en radar.

mejorando el reconocimiento en ambientes dinámicos y adversos. TDFANet [112] es el primer modelo que procesa secuencias de escaneos 4D, eliminando puntos dinámicos, estimando la propia velocidad del sensor y generando mapas BEV alineados con la trayectoria. Además, su modelo aplica una agregación de características deformable y piramidal. Por último, 4D RadarPR [113] hace uso de datos de radar 4D para escenarios adversos como lluvia, niebla y oscuridad. El método utiliza una arquitectura de atención espacial para extraer características robustas desde nubes de puntos radar enriquecidas con velocidad, filtrando puntos dinámicos mediante la compensación de su propio movimiento.

La Tabla 2.3 recoge los métodos más relevantes para reconocimiento de lugares con sensores radar, destacando su año de publicación, tipo de radar utilizado, tipo de entrada (imagen polar, imagen cartesiana o nube de puntos) y la arquitectura o descriptor empleado.

2.2.3.3 Desafíos y robustez

Un desafío importante en el reconocimiento de lugares basado en radar es la escasez de conjuntos de datos públicos con datos radar de alta calidad, lo que limita la capacidad de entrenar y evaluar modelos robustos. A diferencia de los conjuntos de datos abundantes para cámaras y LiDARs, los datos radar son menos accesibles y requieren equipos especializados para su adquisición. Sin embargo, iniciativas como el conjunto de datos MulRan [95], Oxford Radar RobotCar Dataset [100] u Oxford Offroad Radar Dataset [114] han comenzado a abordar esta brecha, proporcionando valiosos datos para la investigación en este campo.

2.3 Enriquecimiento de la información sensorial

El enriquecimiento de la información sensorial se refiere a la incorporación de datos sintéticos generados por otros modelos de IA para mejorar la calidad y robustez del reconocimiento de lugares. Estas técnicas pueden incluir información semántica, geométrica, de profundidad y otras fuentes de enriquecimiento que complementan los datos sensoriales primarios.

La información semántica proporciona una capa adicional de comprensión sobre el entorno, permitiendo a los sistemas de reconocimiento de lugares identificar y clasificar objetos y estructuras según su significado funcional. Esta capacidad de interpretación semántica resulta especialmente valiosa para mejorar la robustez frente a cambios de apariencia y condiciones ambientales, ya que los elementos semánticos suelen mantenerse constantes a pesar de las variaciones temporales o estacionales. En el contexto del reconocimiento visual de lugares, diversos trabajos han explorado la incorporación de información semántica para mejorar el rendimiento. Garg *et al*. [115] demostraron que el uso de información semántica obtenida por RefineNet [116] permite distinguir entre elementos permanentes y transitorios del entorno, mejorando significativamente la robustez del reconocimiento en condiciones cambiantes. Naseer *et al*. [117] propusieron un enfoque que filtra píxeles específicos basándose en su categoría semántica, estimada a través de Fast-Net [118], priorizando elementos estructurales estables como edificios y eliminando objetos variables como vehículos o vegetación. Este filtrado semántico reduce el impacto de elementos dinámicos que pueden confundir al sistema de reconocimiento. Larsson *et al*. [119] desarrollaron una arquitectura de segmentación semántica denominada *Fine-Grained Segmentation Network* (FGSN) que se entrena de manera auto-supervisada para generar segmentaciones con un mayor número de clases que las disponibles en segmentaciones semánticas tradicionales. Esta red produce etiquetas consistentes a través de cambios estacionales, lo que permite utilizarla como representación invariante de la escena para mejorar significativamente el rendimiento de los algoritmos de localización existentes. Este enfoque adaptativo mejora la discriminación entre lugares similares al enfatizar características distintivas y estables. En este mismo contexto, Merrill *et al*. [120] proponen CALC2.0, donde entrenan un *Variational Autoencoder* (VAE) tanto para la descripción global de la escena como para la segmentación semántica de la imagen. Zhang *et al*. [121] y Garg *et al*. [122] proponen respectivamente MESA y SegVLAD, dos métodos que utilizan SAM [123] para segmentar imágenes y generar descriptores subglobales de cada estructura semántica del entorno. Esto permite realizar la búsqueda de lugares utilizando descriptores semánticos, mejorando la precisión y robustez del reconocimiento. Por otro lado, las recientes arquitecturas basadas en *Transformers* y modelos de lenguaje-visión han ampliado las posibilidades de incorporación semántica. Modelos como CLIP [124] permiten asociar representaciones visuales con descripciones textuales, facilitando consultas semánticas de alto nivel para reconocimiento de lugares. Trabajos como Text2SceneGraphMatcher [125] aprovechan esta capacidad para crear sistemas de reconocimiento que pueden responder a consultas en lenguaje natural, como "encuentra un lugar con un edificio rojo", combinando así las capacidades de reconocimiento visual con comprensión semántica del entorno.

En sistemas basados en LiDAR, la incorporación de información semántica ha demostrado ser igualmente beneficiosa. PSE-Match [126] extrae características separadas de las nubes de puntos según diferentes categorías semánticas (árboles, edificios, terreno, etc.) obtenidas por SqueezeSeg [127], generando descriptores más estables y consistentes entre diferentes entornos y condiciones. Esta segmentación semántica permite que el sistema pondere adecuadamente las diferentes partes del entorno según su relevancia para el reconocimiento de lugares. Por ejemplo, la estructura permanente

de los edificios recibe mayor importancia que elementos variables como vegetación o vehículos en movimiento. Pramatarov *et al.* [128] proponen BoxGraph, un sistema de reconocimiento de lugares mediante LiDAR que representa las nubes de puntos como grafos conectados por componentes segmentados semánticamente, donde cada vértice corresponde a una instancia de objeto reduciendo drásticamente el tamaño de los mapas. En cuanto a los sistemas radar, no existen bases de datos públicas con información semántica, por lo que la incorporación de información semántica en este contexto es aún un área de investigación emergente.

En cuanto a la incorporación de información geométrica y de profundidad estimada en los sistemas de reconocimiento de lugares, este tipo de información permite una comprensión más rica del entorno, mejorando la precisión y robustez del reconocimiento. Sin embargo, son pocos los trabajos que han explorado esta área en el contexto del reconocimiento visual de lugares. Milford *et al.* [129] investigaron el uso de técnicas de aprendizaje profundo para generar mapas de profundidad sintéticos [130] en escenarios con cambios extremos de apariencia. A diferencia de otros enfoques de aprendizaje profundo, no se basaron en la capacidad de las CNNs para aprender características invariantes, sino en generar imágenes de profundidad suficientemente buenas a partir de imágenes diurnas. Su enfoque mejoró significativamente la invariancia a puntos de vista del algoritmo SeqSLAM. Sizikova *et al.* [131] entrenan una CNN con datos RGB-D generados de manera sintética [132] para que pueda ser aplicada a datos RGB-D reales. Este enfoque mejora el rendimiento del reconocimiento de lugares al combinar información de intensidad y profundidad, permitiendo una mejor discriminación entre lugares similares. Por último, Garg *et al.* [133] propusieron un sistema que integra información de profundidad estimada mediante UnDEMoN [134] para resolver el problema del reconocimiento visual con cambios extremos de apariencia y puntos de vista opuestos. Este enfoque obtiene puntos característicos a partir de la capa de convolución 5 de ResNet101 [6] los cuales son filtrados por profundidad. Oertel *et al.* [135] aumentaron el reconocimiento visual de lugares con información estructural derivada de *Structure-from-Motion* (SfM). Este enfoque combina una CNN 2D para procesar imágenes con una CNN 3D que opera sobre una representación voxelizada de la nube de puntos SfM. Los experimentos demostraron que incluso una simple concatenación de características globalmente agrupadas de ambas redes mejora significativamente el rendimiento frente a métodos puramente unimodales.

La Tabla 2.4 ofrece un resumen cronológico de los principales métodos de enriquecimiento de información sensorial en el contexto del reconocimiento de lugares, destacando su año de publicación, modalidad de entrada (cámara, LiDAR o radar), tipo de enriquecimiento (profundidad, semántica o lenguaje) y la técnica utilizada para el enriquecimiento.

2.4 Reconocimiento de lugares multimodal y *cross-modal*

El reconocimiento de lugares multimodal y *cross-modal* se refiere a la integración y comparación de datos provenientes de diferentes modalidades sensoriales, como imágenes visuales, nubes de puntos LiDAR, datos de radar, y otros tipos de información.

Método	Año	Modalidad	Enriquecimiento	Técnica
Milford *et al.* [129]	2015	Cámara	Profundidad	CNN [130]
Sizikova *et al.* [131]	2016	Cámara RGB-D	Profundidad	Datos sintéticos [132]
Naseer *et al.* [117]	2017	Cámara	Semántica	Fast-Net [118]
Larsson *et al.* [119]	2019	Cámara	Semántica	FGSN [119]
CALC2.0 [120]	2019	Cámara	Semántica	*Variational Autoencoder* (VAE)
Garg *et al.* [133]	2019	Cámara	Profundidad	UnDEMoN [134]
Oertel *et al.* [135]	2020	Cámara	Nube de puntos 3D	*Structure from Motion* (SfM)
PSE-Match [126]	2021	LiDAR	Semántica	SqueezeSeg [127]
BoxGraph [128]	2022	LiDAR	Semántica	Etiquetas y RangeNet++ [136]
Garg *et al.* [115]	2022	Cámara	Semántica	RefineNet [116]
MESA [121]	2024	Cámara	Semántica	SAM [123]
SegVLAD [122]	2024	Cámara	Semántica	SAM [123]
Text2SceneGraphMatcher [125]	2024	Cámara	Lenguaje	CLIP [124]

Tabla 2.4: Comparativa de métodos para enriquecimiento de información sensorial en reconocimiento de lugares.

Estos enfoques permiten mejorar la robustez y precisión del reconocimiento de lugares al aprovechar las fortalezas de cada modalidad sensorial y compensar sus debilidades individuales.

2.4.1 Reconocimiento multimodal

El reconocimiento de lugares multimodal integra información de diferentes sensores para obtener representaciones más robustas y discriminantes del entorno. Este enfoque permite combinar las fortalezas de cada modalidad sensorial mientras compensa sus debilidades individuales, resultando en sistemas más resilientes frente a condiciones ambientales adversas y cambios de apariencia.

2.4.1.1 Cámara-LiDAR

La fusión de datos visuales y LiDAR ha demostrado ser especialmente efectiva para el reconocimiento de lugares, ya que combina la riqueza visual de las imágenes con la precisión geométrica de las nubes de puntos. Esta combinación permite superar limitaciones individuales, como la sensibilidad a cambios de iluminación en imágenes o la escasez de información semántica en nubes de puntos. Los métodos de fusión multimodal pueden clasificarse según el nivel en el que se combinan las diferentes modalidades. La fusión temprana combina los datos antes de la entrar a la red. La fusión tardía procesa cada modalidad por separado y combina sus representaciones finales. La fusión intermedia integra información en etapas intermedias del procesamiento, permitiendo interacciones más complejas entre las diferentes modalidades. A continuación se describen los trabajos más relevantes en reconocimiento de lugares con cada uno de estos tres enfoques de fusión.

En primer lugar, en cuanto a la fusión tardía, Xie *et al.* [137] propusieron una extracción de características visuales y LiDAR por medio de ResNet50 [6] y PointNetVLAD [4], respectivamente, seguido de una concatenación y una capa *Fully Connected* para generar un descriptor global. De forma similar, PIC-Net [138] implementa una arquitectura de dos ramas, una para imágenes y otra para nubes de puntos. La rama

de imagen utiliza ResNet50 [6] para extraer mapas de características, mientras que la rama de nube de puntos emplea PointNet [75] o LPD-Net [91] (eliminando la capa NetVLAD) para extraer características de cada punto. Estos descriptores por pixel y por punto pasan posteriormente por capas de atención espacial, atención de canal y atención de canal global para obtener el descriptor global final. Por otro lado, MinkLoc++ [139] utiliza MinkLoc3D [84], una mejor arquitectura para procesar la nube de puntos basada en convoluciones 3D dispersas, para LiDAR y ResNet18 [6] para imágenes, seguida de una capa GeM para generar un descriptor global. Para la fusión, prueban tanto la suma de ambos descriptores como la concatenación, obteniendo mejores resultados con la concatenación. El último enfoque de fusión tardía es AdaFusion [140], que propone un método de fusión visual-LiDAR con pesos adaptativos que aprenden la importancia relativa de las características de cada modalidad. Esta adaptabilidad permite que el sistema ajuste automáticamente la contribución de cada sensor a partir de las características aprendidas durante el entrenamiento, mejorando la robustez del reconocimiento en entornos diversos. La arquitectura de AdaFusion consta de dos ramas: una para imágenes y otra para nubes de puntos. Cada rama está compuesta por tres capas convolucionales, 2D y 3D según la modalidad, y mecanismos de atención a multiples escalas.

En segundo lugar, en cuanto a la fusión intermedia, Pan *et al*. [141] proponen CORAL, una arquitectura de fusión intermedia para reconocimiento de lugares multimodal cámara-LiDAR. Su enfoque se basa en construir mapas de elevación densos a partir de nubes de puntos 3D, que capturan la estructura geométrica del entorno en una representación tipo BEV (*Bird's Eye View*). Estos mapas se enriquecen "coloreándose" con características visuales extraídas a nivel de píxel de imágenes RGB, aprovechando la correspondencia punto-píxel proporcionada por la proyección geométrica. La red emplea dos ramas: una para extraer características visuales mediante ResNet18 [6] y FPN, y otra para extraer características estructurales de los mapas de elevación. Ambas ramas se fusionan en el espacio BEV mediante una capa de proyección y un módulo de fusión, generando un descriptor global robusto que integra información visual y geométrica de forma coherente. Por otro lado, Zhou *et al*. [142] presentan LCPR, una arquitectura que aprovecha imágenes RGB de múltiples vistas junto con nubes de puntos LiDAR para generar descriptores invariantes a rotaciones en el plano horizontal. Su módulo de fusión, denominado *Vertically Compressed Transformer Fusion* (VCTF) explota correlaciones tanto intra como inter-modales mediante mecanismos de autoatención, abordando eficazmente el problema del desequilibrio informativo causado por cámaras con campo visual limitado.

En tercer y último lugar, la fusión temprana integra las características de ambas modalidades antes de la etapa de procesamiento de la información sensorial. En este sentido, Liu *et al*. [143] presentan MFF-PR, una arquitectura de fusión temprana que combina características semánticas de nubes de puntos, características de instancia, información topológica y texturas de imagen. Esta integración permite capturar una representación más completa del entorno, mejorando la robustez del reconocimiento en escenas complejas. Por su parte, Xu *et al*. [144] introducen EINet, un enfoque innovador que permite la interacción explícita entre las modalidades visuales y LiDAR. Utiliza

datos de rango LiDAR para proporcionar supervisión de profundidad a las características visuales, mientras que emplea información RGB para enriquecer la representación de las nubes de puntos. Este intercambio mutuo genera características visuales más conscientes de la geometría y representaciones LiDAR más distintivas en términos de apariencia. Por último, Qi *et al*. [145] proponen GSPR, un enfoque de fusión temprana que combina imágenes RGB y nubes de puntos LiDAR en una representación 3D de la escena por medio de la técnica *Gaussian Splatting*. Esta técnica permite una representación más densa, rica y detallada del entorno, mejorando así la información de entrada al modelo de reconocimiento de lugares, que en este caso se trata de una *Graph Neural Network* (GNN) seguida de un bloque *Transformer* y una capa VLAD para generar el descriptor global. Además, GSPR filtra los elementos dinámicos de la escena por medio de una segmentación semántica de las imágenes RGB, lo que mejora la robustez del reconocimiento al centrarse en las características estáticas y permanentes del entorno.

2.4.1.2 Radar-LiDAR

La tarea de reconocimiento multimodal de lugares basada en radar y LiDAR continúa todavía poco explorada, pero la combinación de estas dos modalidades de sensor ha demostrado ser prometedora para mejorar la robustez y precisión en otras tareas de percepción, como en la detección de objetos. En este sentido, Wang *et al*. [146] introducen Bi-LRFusion, un enfoque bidireccional que primero enriquece las características del radar con información LiDAR y luego fusiona ambas en BEV para la detección de objetos dinámicos. En esta misma línea de investigación, Bang *et al*. [147] proponen RadarDistill, una arquitectura de destilación de conocimiento en la que las características aprendidas a partir de la información LiDAR se transfieren a un modelo basado exclusivamente en información radar, mejorando su precisión en el reconocimiento de objetos sin incrementar la carga computacional del proceso.

Más recientemente, LRFusionPR [148] es el primer y único trabajo en aplicar de forma explícita la fusión radar–LiDAR al reconocimiento de lugares. Propone una red con doble rama: la rama de fusión combina datos LiDAR y radar en una representación BEV polar usando atención cruzada; la rama de destilación es entrenada con radar únicamente para garantizar resiliencia ante fallos de sensor. El descriptor resultante es robusto y multimodal, y se ha evaluado satisfactoriamente en entornos urbanos, demostrando una clara ventaja frente a métodos unimodales.

Estos enfoques muestran que la fusión multimodal no solo aporta mejoras en tareas de detección, sino que también tiene un gran potencial para resolver los retos del reconocimiento de lugares en condiciones adversas y entornos dinámicos.

2.4.1.3 Camara-Radar

Aunque centrados en tareas de percepción más que en reconocimiento de lugares, trabajos como RadarCam-Depth [149] han sido fundamentales en establecer arquitecturas comunes, proponiendo una arquitectura de aprendizaje profundo que estima profundidad con escala métrica, a partir de cámaras y radar. Modelos como CenterFusion [150], RODNet [151] y CRF-Net [152] demuestran el potencial de esta combinación

Artículo	Año	Modalidades	Tipo de fusión	Entrada	Modelo(s)
Xie *et al.* [137]	2020	Cámara-LiDAR	Tardía	Imagen, Puntos 3D	PointNetVLAD, ResNet50
PIC-Net [138]	2020	Cámara-LiDAR	Tardía	Imagen, Puntos 3D	LPD-Net, PointNet, ResNet50
CORAL [141]	2021	Cámara-LiDAR	Intermedia	Imagen, Mapa de elevación	ResNet18
MinkLoc++ [139]	2021	Cámara-LiDAR	Tardía	Imagen, Vóxeles 3D	MinkLoc3D, ResNet18
AdaFusion [140]	2022	Cámara-LiDAR	Tardía	Imagen, Puntos 3D	Propia
MFF-PR [143]	2022	Cámara-LiDAR	Temprana	Imagen, Puntos 3D	PointNetVLAD, ResNet50, UNet
LCPR [142]	2023	Cámara-LiDAR	Intermedia	Imagen, Puntos 3D	Propia
EINet [144]	2024	Cámara-LiDAR	Temprana	Imagen, Imagen de rango	OverlapTransformer
GSPR [145]	2024	Cámara-LiDAR	Temprana	Imagen, Puntos 3D (*Gaussian Splatting*)	GNN + Transformer + VLAD
LRFusionPR [148]	2025	Radar-LiDAR	Intermedia, Tardía	BEV polar Radar, BEV polar LiDAR	ResNet18, PolarCrossAttention, VLAD
CRPlace [153]	2023	Cámara-Radar	Intermedia	Imagen, Puntos 3D Radar	Swin-T , PillarFeatureNet

Tabla 2.5: Comparativa de métodos de reconocimiento de lugares multimodal entre diferentes modalidades sensoriales.

multimodal para detección de objetos en 3D, usando representaciones BEV fusionadas con mecanismos de atención. Aunque diseñados para percepción en tiempo real, sus principios arquitectónicos son aprovechables en tareas de reconocimiento geolocalizado.

En cuanto a la fusión de cámara y radar para el reconocimiento de lugares, el único trabajo en esta temática es CRPlace [153] que fusiona imágenes multivista 360º y nubes de puntos generadas por radar. Ambos tipos de información se codifican en el espacio de características por medio de un Swin-Transformer [89] para las imágenes y PillarFeatureNet [154] para los puntos radar. Los mapas de características obtenidos de cada modalidad se proyectan a *Bird's Eye View* (BEV) y se fusionan aplicando atención cruzada bidireccional para combinar eficazmente las características BEV de ambas modalidades. Este enfoque ha demostrado ser eficaz en condiciones adversas como niebla, lluvia o nieve, donde la combinación de datos visuales y radar mejora significativamente la robustez del reconocimiento de lugares.

La Tabla 2.5 sintetiza los enfoques de fusión multimodal más destacados, indicando su año de publicación, modalidades sensoriales utilizadas, tipo de fusión (temprana, intermedia o tardía), tipo de entrada (imagen, nube de puntos, etc.) y los modelos o arquitecturas empleadas.

2.4.2 Reconocimiento cruzado entre modalidades

Los sistemas de reconocimiento cruzado entre diferentes modalidades de sensor (*cross-modal*) permiten la búsqueda y comparación de lugares utilizando datos de diferentes modalidades sensoriales, como imágenes visuales, nubes de puntos LiDAR o datos de radar. Estos enfoques son especialmente útiles en escenarios donde los datos de una modalidad no están disponibles o son insuficientes, permitiendo que robots con diferentes configuraciones sensoriales operen en el mismo entorno utilizando bases de datos compartidas. Esto es particularmente valioso en aplicaciones multi-robot donde diferentes unidades pueden tener configuraciones sensoriales distintas por razones de coste, especialización de tareas, o disponibilidad de hardware. O cuando se dispone de una base de datos con un tipo de sensor y se desea realizar la navegación con otro tipo de sensor.

2.4.2.1 Cámara-LiDAR

El reconocimiento *cross-modal* entre cámaras y LiDARs ha sido un área de investigación activa, con múltiples enfoques que buscan alinear las representaciones de ambos tipos de datos para permitir la búsqueda y comparación efectiva. Estos métodos tratan de superar las diferencias inherentes entre las modalidades por medio de técnicas de aprendizaje profundo que generan espacios de características compartidos o utilizan destilación de conocimiento para transferir capacidades entre modalidades.

Las técnicas de destilación representan un enfoque alternativo donde el conocimiento de un modelo maestro ya entrenado para una modalidad se transfiere a un modelo estudiante que opera en una modalidad diferente. DistilVPR [155] emplea destilación de conocimiento para transferir capacidades representacionales entre modalidades, permitiendo que modelos entrenados con una modalidad operen efectivamente con otra sin requerir datos emparejados durante la operación.

Los métodos que tratan de llevar ambos tipos de sensor a un mismo espacio de características utilizan comúnmente arquitecturas independientes especializadas para cada modalidad que mapean diferentes tipos de datos sensoriales mediante aprendizaje supervisado. Por ejemplo, Cattaneo *et al.* [156] propusieron utilizar VGG16 para procesamiento de imágenes y PointNet para nubes de puntos, entrenando ambas redes para producir representaciones en un espacio compartido donde la distancia refleja similaridad semántica independientemente de la modalidad original. Por otro lado, i3dLoc [157] utiliza Generative Adversarial Networks (GANs) para transformar imágenes panorámicas en proyecciones de rango que son compatibles con representaciones de rango de nubes de puntos LiDAR. El sistema alinea los dos tipos de información sensorial antes de introducirlos al modelo de reconocimiento de lugares, el cual se basa en convoluciones esféricas y una capa NetVLAD para generar descriptores globales. Este enfoque permite que las imágenes panorámicas se utilicen directamente en sistemas de reconocimiento basados en LiDAR, aunque requiere entrenamiento complejo de GANs y puede introducir artefactos que afectan la calidad del reconocimiento. En este contexto, ModaLink [158] propone un *framework* ligero y eficiente para embeber imágenes estándar y nubes de puntos en descriptores distintivos. Su principal contribución es un módulo de transformación de campo de visión que convierte nubes de puntos 360º a imágenes de rango equiparables a las imágenes estándar. El sistema también incorpora una extracción de características semánticas mutuamente consistentes entre nubes de puntos e imágenes. Por otro lado, Li *et al.* [159] proponen VXP, un método de alineación vóxel-píxel auto-supervisado que unifica modalidades en un espacio de características común sin requerir supervisión explícita. Utilizan DINO V2 [14] para procesamiento de imágenes estándar y VoxelNet [160] para nubes de puntos LiDAR, estableciendo correspondencias precisas entre modalidades mediante técnicas de atención visual que identifican regiones correspondientes en ambos tipos de datos. Por su parte, LIP-Loc [161] adapta los principios exitosos de CLIP al dominio específico del reconocimiento de lugares, utilizando aprendizaje por contraste masivo para alinear imágenes estándar y nubes de puntos en un espacio semánticamente coherente. El entrenamiento utiliza grandes cantidades de datos emparejados para aprender representaciones que capturen similaridades semánticas *cross-modal* relevantes para el reconocimiento de lugares. Fi-

nalmente, UniLoc [162] proporciona una solución capaz de procesar texto, imágenes estándar y nubes de puntos en un sistema unificado. Sin embargo, para nubes de puntos requiere información visual adicional para colorear los puntos, lo que limita su aplicabilidad en escenarios puramente *cross-modal* donde solo se dispone de información geométrica.

2.4.2.2 Radar-LiDAR

El reconocimiento cruzado Radar-LiDAR aborda un desafiante escenario debido a las diferencias fundamentales en la naturaleza de los datos: mientras que el LiDAR proporciona información geométrica precisa y densa, el radar ofrece datos más dispersos con menor resolución espacial pero con información adicional de velocidad, lo que requiere de técnicas sofisticadas para alinear estas modalidades heterogéneas en un espacio común de características. Aunque ambos tipos de información se puedan representar con nubes de puntos o imagen BEV en función del tipo de sensor radar (FMCW o mmWave). Por ejemplo, Ma *et al*. [163] proponen RoLM, un método que utiliza la representación BEV para LiDAR y Radar FMCW, inspirado en Scan Context [79] para crear un espacio de características compartido. Este enfoque permite que el sistema realice reconocimiento de lugares utilizando datos de radar y LiDAR, aprovechando la complementariedad de ambos tipos de sensor. Sin embargo, este enfoque requiere que ambos tipos de sensor estén alineados en el mismo espacio BEV, lo que puede ser un desafío en entornos dinámicos o con condiciones adversas. En la misma línea pero con aprendizaje, Nayak *et al*. [164] diseñan RaLF, una red neuronal profunda que aborda simultáneamente el reconocimiento de lugares y la localización métrica para representaciones BEV cartesianas de LiDAR y Radar FMCW. Además, emplean la misma arquitectura (RAFT [165]) para ambas modalidades pero con pesos independientes. Este enfoque ha demostrado un rendimiento destacado en el estado del arte actual, tanto en reconocimiento de lugares como en localización métrica, con capacidad para generalizar a diferentes ciudades y configuraciones de sensores no vistas durante el entrenamiento.

Otro enfoque innovador para mejorar la localización cruzada Radar-LiDAR en representación BEV cartesiana ha sido propuesto por Lisus *et al*. [166], quienes desarrollaron un método basado en aprendizaje profundo para optimizar el proceso de Iterative Closest Point (ICP) mediante pesos aprendidos. Su sistema incorpora una etapa de preprocesamiento de la imagen BEV radar basada en un modelo U-Net para filtrar puntos problemáticos relacionados con artefactos, ruido y vehículos en movimiento. Además, el método de ICP propuesto en este trabajo es diferenciable y por tanto, entrenable. Este enfoque híbrido, que combina técnicas analíticas (ICP) con componentes aprendidos (pesos), logra reducir los errores de localización y mejorar la convergencia del algoritmo ICP en datos de conducción autónoma del mundo real, acercando el rendimiento de los sistemas Radar-LiDAR al de los sistemas LiDAR-LiDAR tradicionales. Finalmente, Yin *et al*. [167] propusieron un sistema que utiliza GANs, y en concreto pix2pix [168], para transformar imágenes de radar en vista de pájaro (BEV) a representaciones similares a LiDAR en esta misma vista. El sistema completo integra esta transformación de estilo con un marco de localización Monte Carlo para un seguimiento de poses preciso, requiriendo únicamente un sensor radar FMCW y un mapa LiDAR global construido con

Método	Año	Modalidades	Tipo de entrada	Arquitectura
Cattaneo *et al.* [156]	2020	Cámara-LiDAR	Imagen, Puntos 3D	VGG16, PointNet
DistilVPR [155]	2022	Cámara-LiDAR	Imagen, Puntos 3D	ResNet, PointNet
i3dLoc [157]	2021	Cámara-LiDAR	Profundidad estimada, Imagen de rango	GAN, CNN esféricas
VXP [159]	2024	Cámara-LiDAR	Imagen, Vóxeles 3D	DINO V2, VoxelNet
LIP-Loc [161]	2024	Cámara-LiDAR	Imagen, Puntos 3D	ViT, PointNet
ModaLink [158]	2024	Cámara-LiDAR	Imagen, Imagen de rango	ResNet34, VLAD
UniLoc [162]	2024	Cámara-LiDAR	Imagen, Puntos 3D	ViT basado en CLIP
Yin *et al.* [167]	2022	Radar-LiDAR	Imagen BEV radar, Imagen BEV LiDAR	pix2pix
RoLM [163]	2023	Radar-LiDAR	Imagen BEV radar, Imagen BEV LiDAR	Scan Context
Lisus *et al.* [166]	2023	Radar-LiDAR	Imagen BEV radar, Imagen BEV LiDAR	ICP diferenciable
RaLF [164]	2024	Radar-LiDAR	Imagen BEV radar, Imagen BEV LiDAR	RAFT

Tabla 2.6: Comparativa de métodos *cross-modal* para reconocimiento de lugares.

anterioridad.

La Tabla 2.6 resume los principales métodos de reconocimiento de lugares *cross-modal*, destacando su año de publicación, modalidades sensoriales utilizadas, tipo de entrada (imagen, nube de puntos, etc.) y las arquitecturas empleadas.

3 Reconocimiento de lugares basado en visión

3.1 Introducción

El reconocimiento visual de lugares (*Visual Place Recognition*, VPR) es una tarea fundamental en el campo de la robótica móvil y la visión por computador. Consiste en identificar la ubicación de un robot o sistema de visión en un entorno previamente explorado, utilizando únicamente información visual. Esta tarea es crucial para una amplia gama de aplicaciones, como la navegación autónoma, la localización y mapeo simultáneos (SLAM) o la planificación de trayectorias. La capacidad de un robot para reconocer lugares de manera precisa y eficiente es esencial para garantizar su autonomía y funcionalidad en entornos complejos y dinámicos.

El VPR enfrenta numerosos desafíos debido a las variaciones en las condiciones de iluminación, los cambios en la apariencia del entorno, las oclusiones y las variaciones en el punto de vista de la cámara. Por ejemplo, un lugar puede parecer completamente diferente dependiendo de si es de día o de noche, si hay objetos nuevos en la escena o si el robot observa el lugar desde un ángulo distinto al registrado previamente. Estas variaciones pueden dificultar la correspondencia entre las imágenes capturadas durante la operación del robot y las imágenes almacenadas en un mapa visual. Además, en entornos de interior, factores como la presencia de personas, el movimiento de mobiliario, la iluminación artificial y la condición lumínica exterior pueden añadir complejidad al problema.

En los últimos años, los avances en el aprendizaje profundo han revolucionado el campo del VPR. Las Redes Neuronales Convolucionales (CNNs) han demostrado ser

particularmente efectivas para extraer características robustas y discriminantes de las imágenes. Estas redes están diseñadas para procesar datos visuales y aprender tanto detalles locales como patrones globales. Modelos preentrenados en grandes conjuntos de datos, como ImageNet Large Scale Visual Recognition [169], han sido adaptados para tareas de VPR mediante técnicas de transferencia de aprendizaje, lo que permite aprovechar su capacidad para generalizar a nuevas tareas y dominios. Las CNNs pueden ser utilizadas para extraer descriptores de imágenes, que son representaciones compactas y discriminantes de las características visuales de un lugar. Estos descriptores se pueden comparar entre sí para determinar si corresponden al mismo lugar.

Además de las CNNs, las Redes Neuronales Siamesas (SNNs) han ganado popularidad en el contexto del VPR debido a su capacidad para aprender funciones de similitud entre pares de imágenes. Estas arquitecturas consisten en dos ramas idénticas que comparten pesos y procesan dos imágenes de entrada para generar descriptores que luego se comparan mediante una métrica de distancia, como la distancia euclídea. Las SNNs son especialmente útiles para tareas de recuperación de imágenes y reconocimiento de lugares, ya que permiten identificar si dos imágenes corresponden al mismo lugar, incluso en presencia de variaciones significativas en las condiciones del entorno.

Otro aspecto crucial en el VPR es el preprocesamiento y aumento de datos. Dado que los conjuntos de datos disponibles para entrenar modelos de VPR suelen ser limitados, es común aplicar técnicas de aumento de datos para generar nuevas instancias a partir de las imágenes originales. Estas técnicas incluyen rotaciones, cambios de brillo y contraste, adición de ruido, oclusiones y transformaciones geométricas. El objetivo es hacer que los modelos sean más robustos frente a las condiciones dinámicas del entorno y evitar el sobreajuste a los datos de entrenamiento. De este modo, sería deseable tener un modelo que pueda generalizar bien ante los diversos cambios que se producen en entornos reales durante la operación del robot: cambios globales de iluminación, cambios direccionales en la luz (sombras, reflejos), rotaciones, pequeños desplazamientos del punto de captura y cambios en los elementos de la escena (mobiliario y personas). Las técnicas de aumento de datos (DA) intentan precisamente simular y paliar estos efectos que inevitablemente aparecerán durante la operación real del robot en entornos dinámicos.

Es importante destacar que en entornos de interior, efectos visuales como sombras, reflejos y cambios en la iluminación artificial son particularmente relevantes, mientras que en entornos exteriores, factores como las condiciones climáticas y la posición del sol pueden tener un impacto significativo en la apariencia visual de las escenas. Sin embargo, algunos de estos efectos son particularmente difíciles de reproducir de forma exacta en las imágenes durante el proceso de aumento de datos. Por ejemplo, simular con precisión la formación de sombras tal como se generarían en el entorno real resulta complejo, ya que dependen de la geometría tridimensional del espacio, la posición y naturaleza de las fuentes de luz, y la interacción entre los diferentes objetos de la escena. A pesar de estas limitaciones, las técnicas de DA suponen una aproximación eficaz para mejorar la robustez de los modelos frente a las variaciones que encontrarán en situaciones reales.

En este capítulo, se exploran los fundamentos del reconocimiento visual de lugares, destacando las técnicas más relevantes y los avances recientes en el uso de modelos de aprendizaje profundo, como las CNNs y las SNNs. Se analizan las ventajas y limitaciones de los enfoques tradicionales y modernos, así como las estrategias de preprocesamiento y aumento de datos que permiten mejorar la robustez de los modelos frente a las condiciones dinámicas del entorno. Además, se presentan estudios de caso y experimentos que ilustran la aplicación práctica de estas técnicas en escenarios reales, proporcionando una visión integral de los desafíos y oportunidades en el campo del VPR.

3.1.1 Contribuciones de este capítulo

En este capítulo se presentan las siguientes contribuciones principales:

- Se realiza una revisión exhaustiva de las técnicas más relevantes en el campo del Reconocimiento Visual de Lugares (VPR), destacando los avances recientes en el uso de modelos de aprendizaje profundo, como las Redes Neuronales Convolucionales (CNNs) y las Redes Neuronales Siamesas (SNNs).
- Se propone un enfoque jerárquico para la localización visual de robots móviles, que combina la clasificación de habitaciones mediante CNNs con la estimación precisa de la posición utilizando descriptores globales extraídos de las activaciones intermedias de la red.
- Se evalúan experimentalmente diferentes arquitecturas de CNN del estado del arte como *backbone* para abordar la tarea de localización jerárquica, analizando su desempeño bajo diversas condiciones de iluminación.
- Se exploran y analizan diversas técnicas de aumento de datos, incluyendo efectos visuales como cambios en la iluminación, contraste, saturación y rotaciones, para mejorar la robustez de los modelos frente a condiciones dinámicas del entorno.
- Se propone un enfoque global basado en Redes Neuronales Siamesas para la localización visual, evaluando su capacidad para estimar directamente la posición del robot en entornos de interior utilizando imágenes panorámicas.
- Se presentan estudios de caso y experimentos que ilustran la aplicación práctica de las técnicas propuestas en escenarios reales, utilizando diversos entornos del conjunto de datos COLD bajo diferentes condiciones de iluminación.

3.2 Trabajos relacionados

Durante los últimos años, la Inteligencia Artificial (IA) ha irrumpido en diversos campos del conocimiento, incluyendo la robótica. En el campo de la robótica móvil se emplean técnicas de IA, por ejemplo, para *mapping* [170-172], localización [173-175], navegación [176, 177] y localización y *mapping* simultáneos [178, 179]. Se puede encontrar una revisión completa del estado del arte sobre las tareas de robótica móvil basadas en el uso de la IA en [180]. Además, existen otras aplicaciones de la IA en el contexto de la robótica móvil que incluyen: navegación autónoma [181-183], detección y reconocimiento facial [184-186], reconocimiento y categorización de objetos

[187-189] y *mapping* y localización [190-192].

En particular, las Redes Neuronales Convolucionales (CNNs) han conformado una de las técnicas más populares entre las herramientas de IA. En cuanto al reconocimiento de lugares, los modelos CNN se propusieron por primera vez para abordar este problema en [50], donde se utiliza un modelo pre-entrenado llamado Overfeat [51] para extraer características de imágenes. Sünderhauf *et al*. [49] proporcionaron una investigación exhaustiva sobre el rendimiento de las características extraídas para el reconocimiento de lugares. De hecho, descubrieron que las características obtenidas de las capas convolucionales eran más robustas ante diferentes condiciones de iluminación que aquellas calculadas en las capas totalmente conectadas, mientras que estas últimas superaban en rendimiento ante los cambios de punto de vista. Bai *et al*. [193] proponen el método SeqCNNSLAM, que consiste en utilizar el modelo AlexNet pre-entrenado [48] para extraer características y alimentar el algoritmo SeqSLAM [66]. También Naseer *et al*. [194] propuso un enfoque similar, pero utilizando GoogleNet [195]. Por otro lado, Cebollada *et al*. [196] propuso la descripción holística de imágenes con CNNs para realizar la localización dentro de mapas topológicos, estudiando su robustez ante cambios de iluminación. Además, Xu *et al*. [197] y Leyva-Vallina *et al*. [198] propusieron técnicas similares para la obtención de la posición del robot. Adicionalmente, Ballesta *et al*. [199] estudió las tareas de localización utilizando CNNs y capas de regresión como descriptores de apariencia global. En este aspecto, algunas arquitecturas bien conocidas se han utilizado como estructuras básicas para desarrollar nuevas redes modificadas para fines de navegación robótica. AlexNet [48], VGG16 [5], GoogleNet [195] o NetVLAD [53] son algunas de ellas.

Las Redes Neuronales Convolucionales presentadas anteriormente se pueden combinar para formar Redes Neuronales Siamesas. Este tipo de redes permiten realizar un aprendizaje por contraste: se basa en mostrar a la red ejemplos similares y ejemplos diferentes sobre el problema en cuestión. En el campo de la robótica, se ha incrementado el uso de estas Redes Siamesas en los últimos años. Por ejemplo, Utkin *et al*. [200] utilizan una Red Neuronal Siamesa para apoyar el control de seguridad de un robot mediante la detección de anomalías en su comportamiento y Zeng *et al*. [201] presentan un sistema robótico de recogida y colocación capaz de identificar y agarrar objetos conocidos y novedosos en entornos desordenados utilizando una Red Neuronal Siamesa. Además, Li *et al*. [202] utilizan la red VGG16 para conformar una estructura Siamesa para la detección y el seguimiento de objetos. En cuanto a la tarea de reconocimiento de lugares, Leyva-Vallina *et al*. [203, 204] han propuesto el uso de este tipo de redes para abordar el problema del reconocimiento de lugares en entornos de exterior. Además, este tipo de arquitecturas siamesas no solo se emplea en el ámbito de la localización visual, sino que también se emplea en el problema de localización utilizando otros tipos de sensores (LiDAR, por ejemplo) [72, 205].

3.3 Reconocimiento visual de lugares a partir de imágenes 360º capturadas por un sistema catadióptrico omnidireccional

El reconocimiento visual tiene grandes similitudes con el problema de localización en el ámbito de la robótica móvil. Este paralelismo se abordó en la Sección 2.1. En este capítulo nos centramos en el concepto de reconocimiento visual de lugares. En concreto, esta tarea se ha abordado mediante dos enfoques diferentes. El primero, aborda el problema de manera jerárquica por medio de una arquitectura y aprendizaje de clasificación. El segundo, aborda la tarea de forma global a través de una arquitectura siamesa y aprendizaje por contraste. A continuación, se describe en qué consiste el reconocimiento visual de lugares jerárquico y global en entornos de interior:

- **Reconocimiento de lugares jerárquico.** Este método consiste en determinar el lugar en el que se encuentra el robot en dos pasos. El primero tiene como propósito determinar la estancia o zona en el que está el robot. En el segundo paso, se estiman las coordenadas de la posición del robot dentro de la habitación o zona anteriormente predicha. En la Sección 3.3.1 se realiza una descripción detallada del método propuesto, que hace uso de modelos de clasificación y activaciones intermedias de la red para extraer descriptores globales de las imágenes capturadas por el robot. Estos descriptores se comparan con los del mapa visual de la estancia predicha para estimar la posición del robot.
- **Reconocimiento de lugares global.** Este método tiene como objetivo determinar las coordenadas en las que se encuentra el robot dentro del entorno completo en una sola fase. De esta manera, se consigue el objetivo en un único paso, pero tendrá un mayor coste computacional cuanto mayor sea el escenario de operación. En la Sección 3.3.2 se realiza una descripción detallada del método propuesto con Redes Neuronales Siamesas y aprendizaje por contraste para la descripción de las imágenes capturadas por el robot. En este caso, se extraen descriptores globales de las imágenes y se comparan con los del mapa visual para estimar la posición del robot en el entorno.

3.3.1 Método jerárquico: de la clasificación de estancias a la estimación de la posición

Este estudio tiene como objetivo abordar el reconocimiento visual de lugares mediante una metodología jerárquica basada en aprendizaje profundo. El enfoque propuesto (Figura 3.1) consta de dos pasos: una etapa inicial de estimación gruesa, que consiste en identificar la habitación desde la que se ha capturado la imagen de test, y una fase posterior de reconocimiento fino, en la que la posición del robot se obtiene mediante una comparación por pares entre la imagen de test y el modelo visual que conforma la habitación preseleccionada.

El primer paso del reconocimiento visual se realiza utilizando la salida de una CNN para clasificación. La capa de salida de dicha CNN está compuesta por R neuronas, cada una correspondiente a una habitación (R es el número de habitaciones o áreas relevantes en el entorno objetivo). Luego, se aplica una función de activación SoftMax y se obtiene la predicción de la habitación. Sin embargo, antes de entrenar la CNN, se

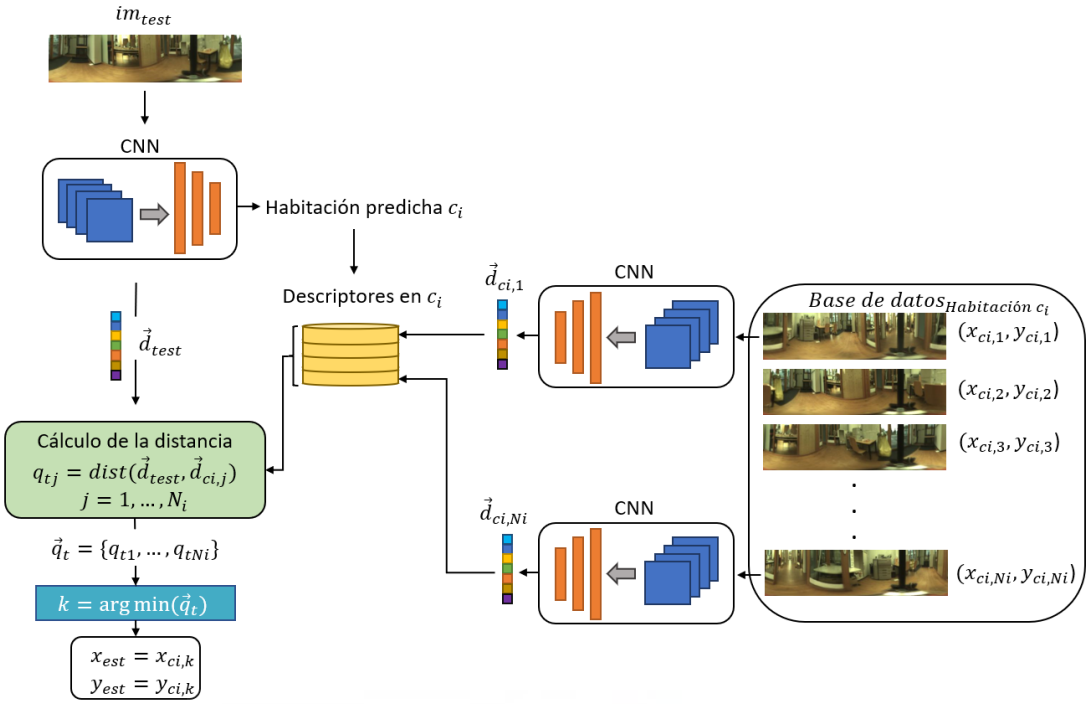


Figura 3.1: Diagrama del reconocimiento visual de lugares jerárquico propuesto. La imagen de test im_{test} es la entrada de la CNN, que predice la habitación más probable c_i y describe la imagen con un vector global $\vec{d}_{test}$ mediante la agregación del último mapa de activación. Este descriptor se compara con los descriptores del conjunto de datos de entrenamiento incluidos en la habitación recuperada mediante una búsqueda del vecino más cercano. En consecuencia, el punto de captura de la imagen que corresponde al descriptor más similar $(im_{c_i,k})$ se considera una estimación de la posición donde se capturó im_{test}. Este diagrama es una adaptación del diagrama original presentado en [206].

necesita un conjunto de datos de imágenes etiquetadas capturadas a lo largo del entorno objetivo. En este caso, cada imagen se etiqueta con la información correspondiente a la habitación dentro de la cual se ha capturado. Posteriormente, la CNN se entrena para abordar la tarea de clasificación de habitaciones empleando una función de pérdida de Entropía Cruzada (Ecuación 3.1).

$$\mathcal{L}(y, \hat{y}) = -\frac{1}{B} \sum_{i=1}^{B} \sum_{j=1}^{R} y_{ij} \log(\hat{y}_{ij}) \tag{3.1}$$

donde y es la matriz de etiquetas e $\hat{y}$ es la matriz de predicciones del modelo. Ambas matrices tienen tamaño $B \times R$, en las que B es el número de muestras (tamaño del lote) y R es el número de clases (habitaciones). y_{ij} es 1 si la muestra i pertenece a la clase j y 0 en caso contrario, e $\hat{y}_{ij}$ es la probabilidad predicha por el modelo de que la muestra i pertenezca a la clase j.

Una vez que la CNN está correctamente entrenada para la clasificación de habitaciones, se puede emplear para llevar a cabo el reconocimiento de lugares en entornos de interior a partir de las imágenes que va capturando el robot. Primero, se realiza el paso

de reconocimiento grueso: una imagen de test im_{test} se introduce en la CNN la cual predice la habitación c_i en la que se capturó la imagen. Simultáneamente, se extrae un descriptor holístico del mapa de activación de la última capa convolucional. Este descriptor $\vec{d}_{test}$ se compara con los descriptores $D_{c_i} = \{\vec{d}_{c_i,1}, \vec{d}_{c_i,2}, \ldots, \vec{d}_{c_i,N_i}\}$ del mapa visual de la habitación predicha c_i, donde N_i es el número de imágenes en la habitación c_i. Cabe destacar que los descriptores del mapa visual también se obtienen a partir del último mapa de activación de la misma CNN. Luego, se calcula la distancia entre el descriptor de test $\vec{d}_{test}$ y cada descriptor $\vec{d}_{c_i,j}\ \epsilon\ D_{c_i}$ correspondiente a la habitación c_i predicha (Ecuación 3.2).

$$q_{t_j} = dist(\vec{d}_{test}, \vec{d}_{c_i,j}), \quad j = 1, \ldots, N_i \tag{3.2}$$

donde N_i es el número de descriptores en la habitación c_i y $dist$ es la distancia euclídea (Ecuación 3.3).

$$dist(\vec{d}_{test}, \vec{d}_{c_i,j}) = \sqrt{\sum_{i=1}^{m}(d_{test,i} - d_{c_i,j,i})^2} \tag{3.3}$$

donde $\vec{d}_{test} = [d_{test,1}, d_{test,2}, \ldots, d_{test,m}]^T$ y $\vec{d}_{c_i,j} = [d_{c_i,j,1}, d_{c_i,j,2}, \ldots, d_{c_i,j,m}]^T$ son los descriptores de tamaño m, y $d_{test,i}$ y $d_{c_i,j,i}$ son los componentes i-ésimos de los vectores $\vec{d}_{test}$ y $\vec{d}_{c_i,j}$, respectivamente.

Después, se construye un conjunto $\vec{q}_t = [q_{t1}, \ldots, q_{tN_i}]^T$ con las distancias calculadas. Se encuentra el índice k que minimiza la distancia en el conjunto $\vec{q}_t$ mediante la Ecuación 3.4. Posteriormente, la posición estimada (x_{est}, y_{est}) corresponde a la posición $(x_{c_i,k}, y_{c_i,k})$ desde la que se capturó la imagen $\vec{im}_{c_i,k}$ del mapa visual (es decir, la imagen cuyo descriptor es el más cercano $\vec{d}_{c_i,k}$ (Ecuación 3.5)). Este enfoque jerárquico garantiza tanto una comprensión amplia de la escena como un reconocimiento del punto de captura preciso dentro de la habitación identificada, contribuyendo a una estrategia efectiva de reconocimiento visual de lugares. La Figura 3.1 esquematiza todo este proceso jerárquico.

$$k = \arg\min(\vec{q}_t) \tag{3.4}$$

$$x_{est} = x_{c_i,k}, \quad y_{est} = y_{c_i,k} \tag{3.5}$$

3.3.2 Método global: aprendizaje por contraste para la estimación de la posición a través de Redes Siamesas

Las Redes Neuronales Siamesas (SNNs) son unas arquitecturas de red neuronal que constan de dos ramas con estructuras idénticas y pesos compartidos. Estas ramas procesan dos imágenes de entrada y generan descriptores que se comparan mediante una métrica de distancia, como la distancia euclídea. La Figura 3.2 ilustra la arquitectura de una SNN.

Las SNNs se basan en el aprendizaje por contraste, donde se aprende a diferenciar entre pares de imágenes similares (positivas) y diferentes (negativas). Estas arquitecturas son especialmente útiles para tareas de reconocimiento visual, donde el objetivo es identificar si dos imágenes representan la misma escena. Para comenzar con este

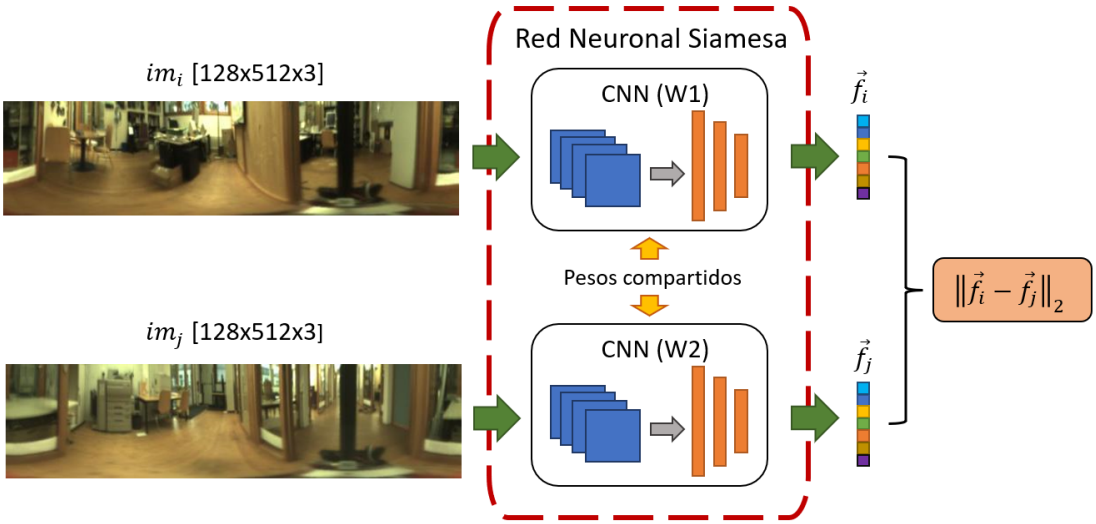


Figura 3.2: Arquitectura de una Red Neuronal Siamesa (SNN). Esta arquitectura consta de dos ramas idénticas que comparten pesos y procesan dos imágenes de entrada para generar descriptores que luego se comparan mediante una métrica de distancia, como la distancia euclídea. Este diagrama es una adaptación del diagrama original presentado en [207].

proceso de aprendizaje, se han de crear las instancias de entrenamiento. En este caso es necesario asociar pares de imágenes con una etiqueta de similitud. Así pues, es necesario seleccionar tanto pares positivos (similares), a partir de imágenes capturadas en posiciones cercanas, como pares negativos (diferentes), a partir de imágenes capturadas en posiciones alejadas (Figura 3.3).

La selección de estos pares positivos y negativos se realiza en base a la distancia euclídea entre los puntos de captura de dichas imágenes. En este caso, se considera un radio de 0.5 metros para determinar la similitud entre dos imágenes. Si estas fueron capturadas a menos de 0.5 metros de distancia, se consideran similares (etiqueta con valor 0), mientras que si fueron capturadas a más de 0.5 metros, se consideran diferentes (etiqueta con valor 1).

Por tanto, dado un par de imágenes de entrenamiento (im_i, im_j), con sus respectivas coordenadas de captura (x_i, y_i) y (x_j, y_j), se obtiene la etiqueta de similitud y_{ij} según la distancia entre ellas:

$$y_{ij} = \begin{cases} 0 & \text{si } \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} <= 0.5 \text{ m} \\ 1 & \text{si } \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} > 0.5 \text{ m} \end{cases} \tag{3.6}$$

Durante el entrenamiento, dados los descriptores $(\vec{d}_i, \vec{d}_j)$ obtenidos a partir de las imágenes de entrada (im_i, im_j), se utilizan junto con la etiqueta de similitud y_{ij} para calcular el error de la función de pérdida, que en el caso de las SNNs, se trata de la función de pérdida por contraste (Contrastive Loss), definida como:

$$\mathcal{L}(\vec{d}_i, \vec{d}_j) = \frac{1}{2}(1 - y_{ij})q_{ij}^2 + \frac{1}{2}y_{ij}\,\text{máx}(\alpha - q_{ij}, 0)^2 \tag{3.7}$$

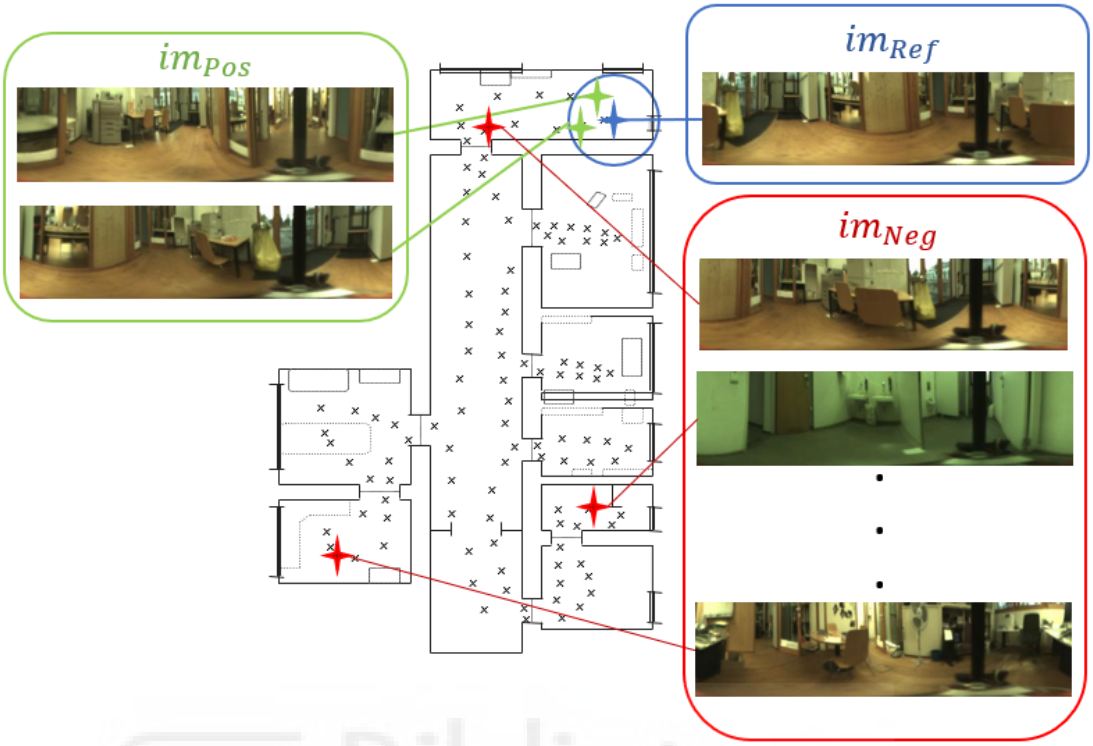


Figura 3.3: Dada una imagen de referencia im_{Ref}, se seleccionan pares positivos im_{Pos} (similares) a partir de imágenes capturadas dentro de un radio de 0.5 metros y pares negativos im_{Neg} (diferentes) a partir de imágenes capturadas a más de 0.5 metros. En este caso, se trata de una etiqueta de similitud binaria, donde 0 indica que las imágenes son similares y 1 indica que son diferentes.

donde y_{ij} es la etiqueta de similitud, q_{ij} es la distancia euclídea entre los descriptores $(\vec{d}_i, \vec{d}_j)$ y $\alpha > 0$ define el margen de separación entre los pares positivos y negativos. En este caso, la función de pérdida se minimiza para que los descriptores de imágenes similares estén lo más cerca posible entre sí, mientras que los descriptores de imágenes diferentes estén separados por al menos un margen α.

Una vez entrenada la SNN, es posible utilizarla para estimar la posición del robot en un entorno dado, sin dividirlo en zonas y en un único paso. En este caso, la imagen de test im_{test} se introduce por una de las ramas de la SNN, la cual genera un descriptor $\vec{d}_{test}$. A su vez, la otra rama de la SNN se alimenta, una a una, con todas las imágenes de la base de datos (mapa visual), generando un conjunto de descriptores $D = \{\vec{d}_1, \vec{d}_2, \ldots, \vec{d}_{N_i}\}$, donde N_i es el número total de imágenes en el mapa visual. El descriptor $\vec{d}_{test}$. se compara con todos los descriptores del mapa visual mediante la Ecuación 3.8.

$$\vec{q_{tj}} = dist(\vec{d}_{test}, \vec{d}_j), \quad j = 1, \ldots, N_i \tag{3.8}$$

donde $dist$ se define como la distancia euclídea (Ecuación 3.9) entre los descriptores

$\vec{d}_{test}$ y $\vec{d}_j$, donde j es el índice de la imagen en el mapa visual.

$$dist(\vec{d}_{test}, \vec{d}_j) = \sqrt{\sum_{k=1}^{m}(d_{test,k} - d_{j,k})^2} \tag{3.9}$$

donde $\vec{d}_{test} = [d_{test,1}, d_{test,2}, \ldots, d_{test,m}]^T$ y $\vec{d}_j = [d_{j,1}, d_{j,2}, \ldots, d_{j,m}]^T$ son los descriptores de tamaño m, y $d_{test,k}$ y $d_{j,k}$ son los componentes k-ésimos de los vectores $\vec{d}_{test}$ y $\vec{d}_j$, respectivamente.

Posteriormente, se construye un conjunto $\vec{q}_t = [q_{t1}, \ldots, q_{tN_i}]^T$ con las distancias calculadas y se encuentra el índice k que minimiza la distancia en el conjunto $\vec{q}_t$ (Ecuación 3.10). Finalmente, la posición estimada (x_{est}, y_{est}) se asigna a la posición (x_k, y_k) de la imagen recuperada im_k del mapa visual (Ecuación 3.11). La Figura 3.4 ilustra todo este proceso.

$$k = \arg\min(\vec{q}_t) \tag{3.10}$$

La posición estimada (x_{est}, y_{est}) se asigna a la posición (x_k, y_k) de la imagen recuperada im_k del mapa visual:

$$x_{est} = x_k, \quad y_{est} = y_k \tag{3.11}$$

El método global tiene la ventaja de ser más directo que el método jerárquico, ya que no requiere un paso previo de clasificación de habitaciones. Además, es aplicable a nuevos entornos tanto de interior como de exterior sin la necesidad del entrenamiento requerido para llevar a cabo dicha localización. Sin embargo, su complejidad computacional es mayor, especialmente en entornos extensos, debido a la necesidad de calcular la distancia entre la imagen de test y todas las imágenes del mapa visual.

3.4 Arquitecturas de Red

Diseñar una Red Neuronal Convolucional para abordar una tarea específica supone un gran desafío. En el presente trabajo, la CNN debe ser capaz de predecir la habitación en la que se capturó una imagen (en el caso de localización jerárquica) y embeber la imagen de entrada en un descriptor global para recuperar la posición exacta dentro de la habitación predicha y/o embeber la imagen dentro de la arquitectura de siamesa para la tarea de reconocimiento visual de lugares global. Por tanto, crear una CNN desde cero exige tanto un profundo conocimiento de las especificidades involucradas, como el acceso a un conjunto de datos suficientemente variado para un entrenamiento efectivo. Además, como se demostró previamente en [208], en términos generales, reentrenar redes diseñadas para un objetivo diferente produce resultados más precisos y fiables en la nueva tarea que entrenarlas desde cero.

Por ello, este trabajo de investigación incorpora varios modelos de CNN ampliamente reconocidos y probados, cada uno de los cuales sirve como arquitectura base para la tarea de reconocimiento de lugares. Estos modelos abarcan una amplia variedad de complejidades arquitectónicas y tamaños. Todas las arquitecturas empleadas fueron

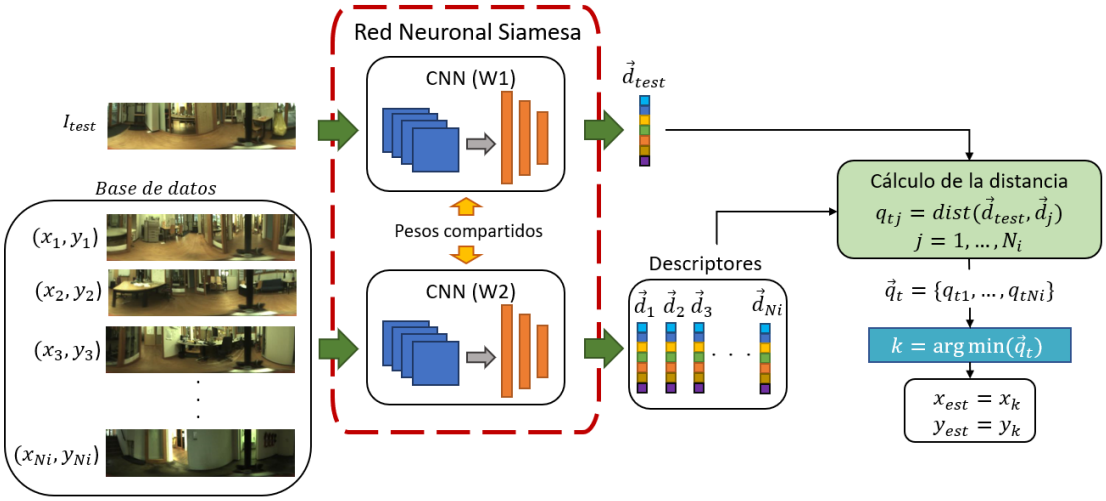


Figura 3.4: Diagrama del reconocimiento visual de lugares global propuesto. La imagen de test im_{test} es la entrada de la SNN, que genera un descriptor global $\vec{d}_{test}$. Este descriptor se compara con los descriptores del conjunto de datos de entrenamiento mediante una búsqueda del vecino más cercano. En consecuencia, el punto de captura de la imagen que corresponde al descriptor más similar (im_k) se considera una estimación de la posición donde se capturó im_{test}.

diseñadas originalmente para el reconocimiento visual de objetos. En este capítulo, la CNN se usa en el caso del reconocimiento visual jerárquico para la clasificación de habitaciones y la posterior estimación de la posición mediante las activaciones intermedias, y en el caso de la reconocimiento de lugares global, para el embeber la imagen en un único descriptor que permita estimar la posición del robot en un solo paso.

- **AlexNet [48]**: AlexNet es una arquitectura de CNN pionera conocida por su éxito en el ImageNet Large Scale Visual Recognition Challenge [169]. Consta de múltiples capas convolucionales y completamente conectadas. AlexNet sentó las bases para diseños posteriores de CNN. Esta red y las siguientes fueron entrenadas para clasificar 1.2 millones de imágenes de alta resolución en 1000 clases diferentes. Los pesos y sesgos obtenidos mediante este entrenamiento han sido tomados como punto de partida para nuestra tarea.
- **VGG16 [5]**: VGG16 es una arquitectura de CNN que se basa en la idea de utilizar capas convolucionales pequeñas (3x3) apiladas para aumentar la profundidad de la red. Esta arquitectura ha demostrado ser efectiva en tareas de clasificación de imágenes y ha sido ampliamente utilizada como base para muchas aplicaciones de visión por computador. En este trabajo, se utiliza VGG16 como una de las arquitecturas base para la tarea de reconocimiento de lugares.
- **ResNet-152 [6]**: ResNet, o Red Residual, introdujo el concepto de aprendizaje residual. Este enfoque se basa en el salto de las conexiones y permite que la CNN aprenda una función de identidad. ResNet-152 es una variante específica con 152 capas, lo que permite al modelo capturar de manera efectiva características jerárquicas complejas. Aunque es computacionalmente costosa debido a su profundidad, su precisión y robustez compensan este coste.

Tabla 3.1: Número de FLOPs y parámetros de los modelos evaluados y adaptados cuando el tamaño de la imagen de entrada es de 512x128x3 píxeles.

Modelo base	FLOPs	Número de Parámetros
AlexNet	0.9 G	57.0 M
VGG16	20.2 G	134.3 M
ResNet-152	15.2 G	58.2 M
ResNeXt-101 64X4d	20.4 G	81.4 M
MobileNetV3	0.3 G	4.2 M
EfficientNetV2	16.2 G	117.2 M
ConvNeXt Large	44.9 G	196.2 M

- **ResNeXt-101 64x4d [7]**: ResNeXt es una extensión de la arquitectura ResNet, enfatizando un parámetro de cardinalidad para mejorar la capacidad del modelo. La cardinalidad es simplemente el número de bloques paralelos, lo que permite aprender diversas representaciones de entrada. En este sentido, ResNeXt-101 64x4d tiene una cardinalidad de 64. Al aumentar la cardinalidad, la red puede capturar una mayor diversidad de características, mejorando su potencial capacidad para el reconocimiento de imágenes.
- **MobileNetV3 [8]**: MobileNetV3 está diseñada para aplicaciones de computación eficiente en dispositivos móviles y periféricos. Utiliza convoluciones separables en profundidad para construir redes neuronales profundas ligeras. Este hecho las hace especialmente adecuadas para plataformas robóticas con limitación de recursos en las que se quiera realizar el reconocimiento de lugares en tiempo real.
- **EfficientNetV2 [9]**: EfficientNetV2 se basa en la arquitectura EfficientNet y utiliza una técnica llamada coeficiente compuesto para escalar modelos de manera simple pero efectiva. Prioriza la eficiencia del modelo, logrando una precisión notable con menos parámetros en comparación con las CNN tradicionales. Esto hace que EfficientNetV2 sea una elección atractiva para aplicaciones que requieren alta precisión con recursos computacionales limitados.
- **ConvNeXt Large [209]**: ConvNeXt Large representa un reciente avance en arquitecturas de CNN. Se basa en la idea de que las CNN pueden beneficiarse de técnicas introducidas por los Transformers: convoluciones separables en profundidad, cuello de botella invertido y factorización espacial ("patchify"). De esta manera, supera a los otros modelos en el ImageNet Large Scale Visual Recognition Challenge [169].

Al evaluar estos modelos de CNN, nuestro objetivo es comprender de manera integral sus fortalezas y debilidades en el contexto del reconocimiento de escenas y localización. Finalmente, la Tabla 3.1 muestra un resumen con los modelos evaluados y su correspondiente número de FLOPs (Operaciones de Coma Flotante) y parámetros.

3.4.1 Adaptación de la CNN para la clasificación de estancias

En cuanto al reconocimiento de habitaciones, la capa "Linear" final de todas las arquitecturas necesita ser adaptada para clasificar las imágenes en R categorías correspondientes a R habitaciones (R=9 en el conjunto de entrenamiento utilizado en el presente trabajo, como se describe en la Sección 3.6). En cuanto al reconocimiento fino, el descriptor global ha sido extraído aplanando el último mapa de características de la CNN.

3.4.2 Adaptación para la arquitectura de siamesa

En el método global comparamos a su vez los modelos anteriores conformando el *backbone* de la arquitectura de siamesa. Este *backbone* constituirá la etapa de aprendizaje de características. Posteriormente, en la etapa de agregación de características se convierte el último mapa de características en un solo descriptor por medio de algún tipo de *pooling*. Para ello, se proponen 3 alternativas:

- **Capas Linear**: la capa lineal es una capa de red neuronal que aplica una transformación lineal a la entrada. Se aplican tres capas consecutivas con 500, 500, 5 neuronas respectivamente. La salida de la última capa es un vector de 5 dimensiones.
- **Capa GeM *Pooling***: la capa GeM (*Generalized Mean Pooling*) es una técnica de agrupamiento que generaliza el promedio y el máximo. Se define como:

$$\text{GeM}(x) = \left(\frac{1}{N} \sum_{i=1}^{N} x_i^p \right)^{\frac{1}{p}} \tag{3.12}$$

  donde x es el vector de activaciones, N es el número de elementos en el vector y p es un parámetro que controla la forma de la función. Cuando $p = 1$, GeM se convierte en promedio, y cuando $p \to \infty$, se convierte en máximo. Este enfoque permite capturar características más discriminantes al considerar tanto la magnitud como la distribución de las activaciones.
- **Capa convolucional 1x1 con GeM *Pooling***: esta capa combina la convolución 1x1 con GeM Pooling. La convolución 1x1 se utiliza para reducir la dimensionalidad de las activaciones, mientras que GeM *Pooling* se aplica para obtener un descriptor global. Esta combinación permite capturar características espaciales y de canal de manera efectiva.

3.5 Aumento de Datos

El entrenamiento de un modelo implica ajustar sus parámetros para realizar una tarea específica. Cuando un modelo tiene muchos parámetros, requiere de un número elevado de ejemplos para que exista un entrenamiento efectivo. Sin embargo, en la práctica, los conjuntos de entrenamiento suelen ser limitados. En estos casos, el aumento de datos (*data augmentation*) es una posible solución, ya que permite generar nuevas instancias de entrenamiento aplicando diferentes efectos visuales. Esto no solo

ayuda al modelo a evitar el sobreajuste (*overfitting*), sino que también lo hace más robusto frente a las desafiantes condiciones de los entornos reales.

En estudios previos, se han aplicado diversos efectos durante el entrenamiento como cambios de orientación, alteraciones en la iluminación, ruido y oclusiones [210], mejorando el rendimiento del modelo. Estos efectos se aplican individualmente o en combinación a cada imagen del conjunto de datos original, generando un nuevo conjunto de entrenamiento aumentado. Sin embargo, el impacto específico de cada tipo de efecto en el rendimiento final de la CNN no está completamente estudiado en la literatura. Por ello, resulta interesante cómo influye cada efecto en el rendimiento del modelo. Así pues, el presente capítulo se centra en evaluar el impacto de distintos efectos de *data augmentation* en la tarea de reconocimiento visual de lugares. Para ello, se han seleccionado y aplicado diferentes efectos visuales a las imágenes del conjunto de datos original, generando un nuevo conjunto de entrenamiento aumentado. Cada efecto se aplica individualmente para evaluar su influencia en el rendimiento del modelo.

En este trabajo se ha experimentado con la adición de dos tipos diferentes de transformaciones de las imágenes: cambios de iluminación y cambios de orientación. Para los cambios en iluminación, se consideran los siguientes efectos:

- **Focos de luz y sombras:** en interiores es común la presencia de fuentes de luz circulares, como bombillas. La estrategia propuesta consiste en aumentar los valores de los píxeles para simular una mayor intensidad lumínica en determinadas zonas puntuales (*spotlights*) y disminuirlos para simular sombras (*shadow spots*). Estos efectos se aplican por separado en diferentes opciones de aumento de datos. En nuestros experimentos, estos focos de luz y sombras se crean con diámetros que varían entre 15 y 40 píxeles. Se aplican cinco niveles de variación en la intensidad. En el primer nivel, la intensidad se modifica de forma aleatoria en ± 160, y en el quinto nivel, en ± 100.
- **Brillo y oscuridad general:** para crear imágenes más brillantes, se aumentan los valores de baja intensidad de la imagen original, simulando una mayor iluminación general (por ejemplo, un día soleado). Por otro lado, para generar imágenes más oscuras, se reducen los valores de alta intensidad, simulando una menor iluminación (por ejemplo, capturas tomadas de noche). Los efectos de brillo y oscuridad se aplican por separado, pero ambos forman parte del mismo aumento de datos.
- **Contraste:** el contraste de la imagen es un factor clave en la diferenciación de objetos dentro de una escena. Las imágenes con bajo contraste tienden a presentar un aspecto más suave, con menos sombras y reflejos. El contraste se modifica siguiendo la Ecuación 3.13:

$$I_s = 64 + c * (I - 64) \tag{3.13}$$

  donde I_s es la imagen resultante, I es la imagen original y c es el factor de contraste. Para $c > 1$, el contraste aumenta, mientras que para $c < 1$, el contraste disminuye.
- **Saturación:** la saturación del color es una medida de la "pureza" del color. Así

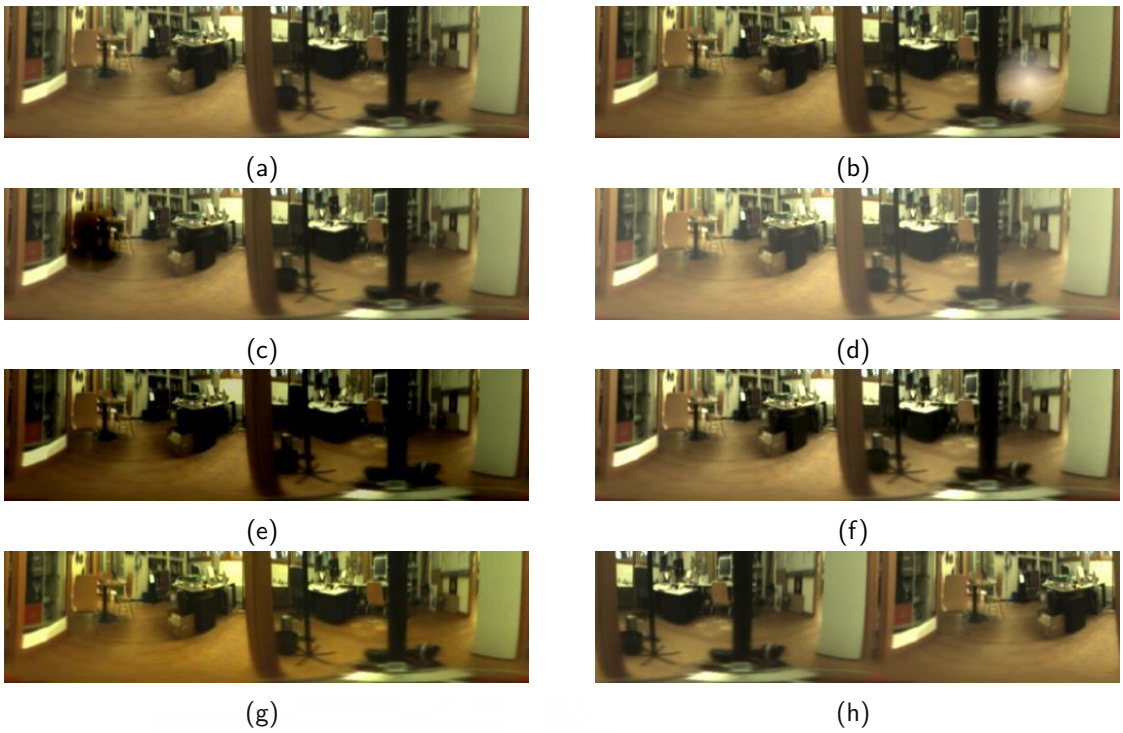


Figura 3.5: Ejemplo de *data augmentation* donde sólo se aplica un efecto por imagen. (a) Imagen original, (b) efecto de foco de luz, (c) efecto de sombra, (d) aumento de brillo general, (e) reducción de brillo general, (f) modificación de contraste, (g) modificación de saturación y (h) rotación. Esta figura ha sido extraída de [206].

pues, una luz monocromática saturada está concentrada en una única longitud de onda, mientras que una luz poco saturada se distribuye a lo largo de un espectro de longitudes de onda. Una saturación menor genera imágenes menos coloridas, y con saturaciones extremadamente bajas, pueden asemejarse a imágenes en escala de grises. Este fenómeno puede ocurrir en entornos reales bajo ciertas condiciones de iluminación y, por lo tanto, se incorpora al *data augmentation*. La saturación del color se ajusta convirtiendo la imagen de RGB a HSV y modificando directamente el canal de saturación multiplicándolo por un factor s. Si $s > 1$, los colores se vuelven más saturados; si $s < 1$, la saturación disminuye.

En cuanto a los cambios en orientación, estos pueden ocurrir cuando el robot captura imágenes desde la misma posición pero con una orientación diferente. Para esta opción de *data augmentation*, se generan nuevas imágenes aplicando rotaciones de $n = i \times 10°, i \in [1, 35]$. En la proyección panorámica de la imagen omnidireccional, una rotación equivale a una transposición de las columnas de la imagen, lo que facilita significativamente la implementación de este efecto visual sin distorsiones adicionales.

La Figura 3.5 muestra un ejemplo de los efectos aplicados a una imagen omnidireccional convertida a formato panorámico. La primera imagen corresponde a la original, mientras que las demás incluyen los distintos efectos presentados anteriormente (aplicados de manera individual).

	Entren.	Mapa	Test			Extra test		
	Nublado	Nublado	Nublado	Noche	Soleado	Nublado	Noche	Soleado
FR-A	556	556	2595	2707	2114	-	-	-
FR-B	-	560	-	-	-	2008	-	1797
SA-A	-	586	-	-	-	2774	2267	-
SA-B	-	321	-	-	-	836	870	872

Tabla 3.2: Número de imágenes de entrenamiento y evaluación de los diferentes escenarios para las tres condiciones de iluminación.

3.6 Conjunto de Datos

El presente estudio utiliza imágenes provenientes de los entornos Freiburg y Saarbrücken del conjunto de datos de COLD (COsy Localization Database) [1], que puede descargarse desde https://www.cas.kth.se/COLD/. En concreto, se han utilizado los entornos: Freiburg A y B (FR-A, FR-B) y Saarbrücken A y B (SA-A, SA-B). Cada conjunto contiene imágenes omnidireccionales capturadas por un robot que sigue distintas trayectorias dentro de un edificio de la Universidad de Freiburg y otro de la Universidad de Saarbrücken, respectivamente. En particular, los conjuntos de FR-A, FR-B, SA-A y SA-B contienen imágenes a lo largo de nueve, cinco, ocho y cinco habitaciones, respectivamente. Entre las cuales se encuentran cocinas, baños, áreas de impresión, zona de escaleras, pasillos y oficinas. La captura de imágenes se realizó bajo condiciones de operación reales, incluyendo cambios en la disposición del mobiliario, la presencia dinámica de personas en las escenas y variaciones en las condiciones de iluminación: nublado, soleado y noche. En este sentido, el robot adquiere imágenes mientras se mueve, lo que introduce posibles efectos de desenfoque o alteraciones dinámicas. Además, estos entornos contienen zonas con amplias ventanas y paredes de cristal, lo que hace que la localización visual sea un problema particularmente desafiante bajo diferentes condiciones de iluminación. En consecuencia, este conjunto de datos proporciona condiciones ideales para evaluar los métodos de localización propuestos en escenarios y condiciones reales de operación.

El conjunto ofrece además de las imágenes omnidireccionales, información de los puntos de captura de las mismas. Esta información resulta vital para llevar a cabo el presente estudio, pues se emplea tanto para el etiquetado como evaluación de los métodos propuestos. La información de los puntos de captura fue obtenida mediante un sensor láser y una técnica de SLAM basada en mapas de ocupación 2D, la cual está descrita en [211] y [212]. El error medio de posición en la trayectoria se estima entre 5 y 10 centímetros, pues corresponde con la resolución de los mapas empleados durante la fase de SLAM. Sin embargo, existen zonas donde el error puede ser mayor, como en los pasillos.

En cuanto a la división de las secuencias en entrenamiento, mapa y test, se ha utilizado únicamente un subconjunto de imágenes nubladas pertenecientes a FR-A para el entrenamiento. Este subconjunto se ha obtenido tras muestrear la secuencia de nublado (*seq2_cloudy3*) en intervalos de 20 cm. Como resultado, se han obtenido

556 imágenes en este entorno, los cuales serán utilizados como la única información disponible para el entrenamiento de la red. Además, para obtener el mapa visual de cada uno de los entornos de FR-A, FR-B, SA-A y SA-B, se ha seguido el mismo procedimiento que en los datos de entrenamiento. Es decir, el mapa visual contendrá imágenes capturadas en condiciones nubladas muestreadas cada 20 cm, mientras que la evaluación se realizará con imágenes capturadas en condiciones nubladas, soleadas y nocturnas. Para los diferentes estudios, se ha empleado como evaluación el conjunto de datos de test de FR-A en condiciones nubladas (*seq2_cloudy2*), soleadas (*seq2_sunny2*) y nocturnas (*seq2_night2*). Cabe destacar que el conjunto de test de nublado difiere tanto del conjunto de entrenamiento como del mapa visual.

Los otros escenarios se han empleado para evaluar la robustez de los métodos propuestos ante entornos nunca vistos antes. En este sentido, se han utilizado los conjuntos de datos de test de FR-B (*seq3_cloudy2*, *seq3_sunny2*), SA-A (*seq2_cloudy2*, *seq2_night1*) y SA-B (*seq4_cloudy2*, *seq4_night2*, *seq4_sunny1*). El mapa visual de estos entornos se ha obtenido siguiendo el mismo procedimiento que para FR-A, muestreando cada 20 cm imágenes capturadas en condiciones nubladas de una secuencia diferente a la de test. En concreto, las secuencias específicas elegidas para los mapas visuales son: FR-A (*seq2_cloudy3*), FR-B (*seq3_cloudy1*), SA-A (*seq2_cloudy3*) y SA-B (*seq4_cloudy1*).

Asimismo, el conjunto de entrenamiento se ha sometido a una aumento de datos (*data augmentation*), como se describe en la Sección 3.5, generando seis conjuntos de entrenamiento adicionales. Estos conjuntos se emplearán individualmente para entrenar las CNN, permitiendo explorar el impacto de cada efecto visual en el rendimiento de la red. La Tabla 3.2 muestra un resumen con el número de imágenes que componen los conjuntos de entrenamiento, base de datos, test y extra test (en entornos no conocidos).

3.7 Experimentos del método jerárquico

3.7.1 Estudio sobre la influencia de la arquitectura de clasificación

En esta sección, se evalúa experimentalmente el desempeño de los diferentes modelos de CNN utilizados como *backbone*, presentados en la Sección 3.4, tanto para la localización gruesa como para la fina.

Como se mencionó anteriormente, la localización jerárquica propuesta en este estudio consta de dos etapas. La primera, denominada localización gruesa, implica el reentrenamiento de un modelo para llevar a cabo la tarea de clasificación de habitaciones. Posteriormente, en la etapa de localización fina, se emplea la CNN previamente entrenada para generar descriptores holísticos. A partir de ellos se realiza la búsqueda de vecinos más cercanos entre la imagen de test y el mapa para estimar con precisión la posición en la que se capturó dicha imagen.

3.7.1.1 Reconocimiento grueso de lugares

En esta sección se presentan los resultados obtenidos al utilizar diferentes CNNs para la ejecución de la etapa de localización gruesa o clasificación de habitaciones. Como se describe en la Sección 3.4, los modelos de CNN evaluados en este estudio son AlexNet [48], ResNet-152 [6], ResNeXt-101 64x4d [7], MobileNetV3 [8], EfficientNetV2 [9] y ConvNeXt Large [209]. La razón detrás de la selección de estos modelos es cubrir un amplio rango de arquitecturas propuestas para la clasificación de imágenes en los últimos diez años.

Modelo *backbone*	**Exactitud en la clasificación de estancias (%)**			
	Nublado	**Noche**	**Soleado**	**Global**
AlexNet	97.61	97.60	70.67	89.93
VGG16	**98.92**	97.08	**89.64**	**95.21**
ResNet-152	96.76	96.64	64.95	87.63
ResNeXt-101 64X4d	98.11	95.16	72.47	89.71
MobileNetV3	98.50	96.93	77.29	91.88
EfficientNetV2	98.81	97.16	75.73	91.63
ConvNeXt Large	98.77	**97.64**	86.28	94.80

Tabla 3.3: Estudio sobre la influencia de diferentes arquitecturas para la clasificación de habitaciones, evaluadas bajo tres condiciones de iluminación distintas: nublado, noche, soleado y en conjunto.

Los resultados de la Tabla 3.3 muestran el desempeño de los seis modelos propuestos como *backbone* en el contexto de la clasificación de habitaciones bajo diversas condiciones lumínicas (nublado, noche y soleado), lo que proporciona una comprensión integral de su robustez y adaptabilidad a los cambios de iluminación en el entorno. AlexNet presenta un excelente rendimiento general, especialmente en condiciones nubladas, con una exactitud del 97.61 %. Por otro lado, VGG16 destaca por ser el modelo más preciso en condiciones nubladas y soleadas, alcanzando los valores 98.92 % y 89.64 %, respectivamente. Sin embargo, ya se puede apreciar como el rendimiento general disminuye en condiciones soleadas. En cuanto a ResNet, muestra un rendimiento ligeramente inferior al de AlexNet y VGG16 y además, se trata del modelo que más dificultades ha presentado en condiciones soleadas. Por otro lado, aunque ResNeXt presenta buenos resultados en condiciones nubladas, su exactitud es inferior al resto de modelos en escenarios nocturnos. En cuanto a MobileNet, destaca por su consistencia a lo largo de todas las condiciones. Además, EfficientNet se posiciona como uno de los modelos con mejor rendimiento, obteniendo el segundo y el tercer mejor resultado para escenarios nublados y nocturnos, respectivamente. Finalmente, ConvNeXt logra resultados consistentes en todos los escenarios, destacando en condiciones de iluminación nubladas, pero sin llegar a superar a VGG16, que destaca en condiciones nubladas y soleadas, lo que refleja su robustez y capacidad de generalización.

Modelo *backbone*	Nublado		Noche		Soleado		Global	
	R@1	R@1 %	R@1	R@1 %	R@1	R@1 %	R@1	R@1 %
AlexNet	89.06	94.91	92.91	99.22	55.82	72.37	79.26	88.83
VGG16	**90.06**	95.65	**92.39**	98.97	**77.25**	91.11	**86.57**	95.24
ResNet-152	88.82	97.42	90.10	98.82	38.22	70.77	72.38	89.00
ResNeXt-101 64X4d	83.82	95.72	89.84	99.19	37.09	62.87	70.25	85.93
MobileNetV3	89.40	98.27	92.17	99.34	55.82	81.32	79.13	92.98
EfficientNetV2	80.34	94.57	84.74	98.12	39.36	62.11	68.15	84.93
ConvNeXt Large	**90.06**	**98.77**	92.02	**99.30**	65.14	**91.91**	82.41	**96.66**

Tabla 3.4: Estudio sobre la influencia de diferentes arquitecturas para la tarea completa de reconocimiento de lugares, evaluadas con un umbral de 0.5 m para R@1 y R@1 % y bajo tres condiciones de iluminación distintas: nublado, noche, soleado y en conjunto.

3.7.1.2 Reconocimiento fino de lugares

Una vez entrenado el modelo de CNN para la etapa de clasificación de habitaciones, se puede utilizar para embeber la imagen de entrada en un descriptor global. Esto facilita la resolución de la etapa del reconocimiento de la posición más cercana mediante un proceso de recuperación de imágenes, en el que el descriptor de la imagen de test se compara con los descriptores del mapa visual de la habitación previamente predicha. Al igual que en la subsección anterior, evaluamos el desempeño de los diferentes modelos de CNN para abordar la etapa de reconocimiento fino. La Tabla 3.4 muestra los resultados en términos de *Recall at 1* (R@1) y *Recall at 1 %* (R@1 %) para el reconocimiento de lugares jerárquico a partir de los diferentes *backbones* (AlexNet, VGG16, ResNet-152, ResNeXt-101, MobileNetV3, EfficientNetV2 y ConvNeXt Large) y bajo diversas condiciones de iluminación (nublado, noche, soleado).

En el reconocimiento jerárquico de lugares, cada modelo ha mostrado un comportamiento similar al previamente evaluado en la clasificación de estancias, pues ambas tareas están relacionadas. Los resultados presentados en la Tabla 3.4 revelan diferencias significativas en el rendimiento de los diversos modelos *backbone* evaluados para la tarea de reconocimiento visual de lugares bajo distintas condiciones lumínicas. Nuevamente, VGG16 destacó como el modelo con mejor rendimiento global, alcanzando un R@1 del 86.57 % considerando todas las condiciones. Este modelo demostró una robustez notable en condiciones soleadas, logrando un 77.25 % de R@1, superando ampliamente a otras arquitecturas más modernas. Estos resultados sugieren que la arquitectura VGG16 es relativamente simple pero proporciona características visuales altamente discriminantes para el reconocimiento de lugares, incluso bajo condiciones de iluminación desafiantes. Por su parte, ConvNeXt Large presentó el mejor rendimiento en términos de R@1 % en todos los escenarios, alcanzando un 96.66 % a nivel global. No obstante, su precisión absoluta en términos de R@1 fue inferior a la de VGG16, especialmente en condiciones soleadas donde alcanzó tan solo un 65.14 %. Además, arquitecturas como ResNet-152, ResNeXt-101 64X4d y EfficientNetV2 mostraron un comportamiento inesperadamente deficiente en condiciones soleadas, con valores de R@1 del 38.22 %, 37.09 % y 39.36 %, respectivamente. Por último, MobileNetV3 y AlexNet presentaron un equilibrio interesante entre rendimiento y eficiencia computacional, manteniendo resultados consistentes en los diferentes escenarios (R@1 global

del 79.13 % y 79.26 %, respectivamente).

3.7.2 Estudio sobre la influencia del Aumento de Datos

En este estudio, se evalúa la influencia de los efectos del aumento de datos (cambios en la iluminación y en la orientación) en el rendimiento de la CNN. Con el fin de asegurar que el modelo sea capaz de generalizar a diferentes condiciones de iluminación y variaciones en la orientación, se han realizado experimentos con diferentes conjuntos de datos aumentados. Estos conjuntos se han creado aplicando los efectos de aumento de datos propuestos en la Sección 3.5.

Además, la probabilidad con la que pueden aparecer estos efectos durante el proceso de entrenamiento se ha fijado en un 40 %. Esto significa que el 40 % de las imágenes de entrenamiento se verán afectadas por alguno de los efectos de aumento de datos, mientras que el 60 % restante permanecerá sin cambios. Esta estrategia busca garantizar que el modelo no dependa exclusivamente de los efectos aplicados, sino que también aprenda a reconocer patrones en imágenes sin alteraciones.

Por otro lado, con el fin de evitar sesgos entre los diferentes conjuntos de datos, cada conjunto de entrenamiento cuenta únicamente con 556 imágenes, que corresponden a las imágenes de entrenamiento del conjunto de datos base. Este enfoque asegura que cada conjunto de datos aumentados contenga la misma cantidad de imágenes, lo que permite una comparación justa entre los diferentes efectos aplicados.

Como resultado de la aplicación de estos efectos, se han obtenido seis conjuntos de datos de entrenamiento adicionales: Conjunto de Datos Aumentado 1 (focos de luz), Conjunto de Datos Aumentado 2 (sombras), Conjunto de Datos Aumentado 3 (brillo/oscuridad general), Conjunto de Datos Aumentado 4 (contraste), Conjunto de Datos Aumentado 5 (saturación) y Conjunto de Datos Aumentado 6 (rotaciones). Como en experimentos previos, el mapa visual no contiene ningún tipo de efecto y para la evaluación del modelo, se consideran tres condiciones de iluminación: nublado, soleado y noche.

3.7.2.1 Reconocimiento grueso de lugares

En esta subsección utilizamos la mejor arquitectura de CNN obtenida en la Sección 3.7.1.1, que es VGG16. De aquí en adelante, se empleará este modelo para el reconocimiento jerárquico de lugares y se referirá a él como Single VGG16. Siguiendo un enfoque similar, partimos de los pesos preentrenados en ImageNet Large Scale Visual Recognition Challenge [169] y reentrenamos el modelo con los diferentes conjuntos de datos obtenidos mediante el aumento de datos propuesto.

La Tabla 3.5 presenta un análisis comparativo de diferentes técnicas de *data augmentation* aplicadas al modelo Single VGG16 para la tarea de clasificación de habitaciones. Los resultados muestran claramente que todas las estrategias de aumento de datos mejoraron el rendimiento general del modelo respecto a entrenar con el conjunto base. El aumento mediante rotaciones (Aumentado 6) demostró ser la técnica más

Conjunto de Entrenamiento	Exactitud en la clasificación de estancias (%)			
	Nublado	Noche	Soleado	Global
Base	98.92	97.08	89.64	95.21
Aumentado 1 (Focos de luz)	99.08	97.08	93.00	96.39
Aumentado 2 (Sombras)	99.04	97.23	93.80	96.69
Aumentado 3 (Brillo/Oscuridad)	**99.27**	96.97	90.02	95.42
Aumentado 4 (Contraste)	99.00	**97.34**	94.89	97.08
Aumentado 5 (Saturación)	98.57	97.16	91.30	95.68
Aumentado 6 (Rotaciones)	98.96	97.30	**95.22**	**97.16**

Tabla 3.5: Estudio sobre la influencia de diferentes efectos de aumento de datos utilizados en el entrenamiento de Single VGG16 para la clasificación de habitaciones, evaluados bajo tres condiciones de iluminación distintas: nublado, noche, soleado y en conjunto.

efectiva a nivel global, alcanzando una precisión del 97.16 %. Este resultado subraya la importancia de la invariancia a la orientación en sistemas de reconocimiento visual de lugares. Particularmente, las rotaciones proporcionaron la mayor mejora en condiciones soleadas (95.22 %), que representan el escenario más desafiante debido a las fuertes variaciones de iluminación. Cabe destacar que las imágenes de test no tienen por qué haberse capturado con la misma orientación que las del mapa (sobre todo cuando el robot está describiendo curvas), de ahí que este efecto sea especialmente beneficioso en todas las condiciones de iluminación. Las modificaciones en el contraste (Aumentado 4) también mostraron resultados prometedores, con una precisión global del 97.08 % y el mejor rendimiento en condiciones nocturnas (97.34 %). Esto sugiere que entrenar el modelo para reconocer características visuales bajo diferentes niveles de contraste mejora su capacidad para identificar lugares en entornos con variaciones de iluminación. La modificación de brillo/oscuridad general (Aumentado 3) resultó especialmente beneficiosa para el reconocimiento en condiciones nubladas, logrando la mayor precisión en este escenario (99.27 %). Sin embargo, mostró un rendimiento limitado en condiciones soleadas (90.02 %), lo que sugiere que los cambios globales de iluminación podrían no ser suficientes para simular adecuadamente las complejas variaciones lumínicas producidas por la luz solar directa. Por otro lado, la adición de focos de luz (Aumentado 1) y sombras (Aumentado 2) mostraron también buenos resultados generales (96.39 % y 96.69 % respectivamente), lo que indica que los cambios tanto globales como locales en la iluminación son efectivos para mejorar el rendimiento del modelo. En particular, las sombras artificiales demostraron ser especialmente útiles para el reconocimiento en condiciones soleadas (93.80 %), probablemente porque simulan de manera más realista los efectos de sombras producidos por la luz solar en entornos interiores. Por último, la modificación de saturación (Aumentado 5) mostró un rendimiento intermedio (95.68 % de exactitud global), destacando su capacidad para mejorar el reconocimiento en condiciones soleadas (91.30 %) respecto al modelo base.

3.7.2.2 Reconocimiento fino de lugares

Una vez que el modelo Single VGG16 ha sido entrenado para la etapa de clasificación de habitaciones, puede utilizarse para codificar la imagen de entrada en un

Conjunto de entrenamiento	Nublado		Noche		Soleado		Global	
	R@1	R@1 %	R@1	R@1 %	R@1	R@1 %	R@1	R@1 %
Base	**90.06**	**95.65**	92.39	98.97	77.25	91.11	86.57	95.24
Aumentado 1 (Focos de luz)	88.94	95.18	93.24	99.04	76.58	93.14	86.25	95.79
Aumentado 2 (Sombras)	89.90	95.53	92.91	99.08	79.85	94.18	87.55	96.26
Aumentado 3 (Brillo/Oscuridad)	89.67	95.41	**93.31**	98.85	76.58	91.44	86.52	95.23
Aumentado 4 (Contraste)	89.67	95.03	93.02	98.82	**81.55**	94.75	**88.08**	96.20
Aumentado 5 (Saturación)	89.75	95.26	93.06	98.89	78.67	92.01	87.16	95.39
Aumentado 6 (Rotaciones)	89.44	95.57	92.83	**99.04**	78.05	**95.88**	86.77	**96.83**

Tabla 3.6: Estudio sobre la influencia de diferentes efectos de aumento de datos utilizados en el entrenamiento de Single VGG16 para la tarea completa de reconocimiento de lugares, evaluados con un umbral de 0.5 m para R@1 y R@1 % y bajo tres condiciones de iluminación distintas: nublado, noche, soleado y en conjunto.

descriptor global. Esto permite resolver el reconocimiento fino de lugares a través de un proceso de recuperación de imágenes, en el cual el descriptor de la imagen de test se compara con los descriptores del mapa visual. Como en la subsección anterior, vamos a evaluar el rendimiento de los diferentes efectos de aumento de datos para abordar la etapa de localización fina.

La Tabla 3.6 presenta los resultados finales de la localización jerárquica tras aplicar diferentes técnicas de aumento de datos. El análisis revela patrones interesantes sobre la efectividad de cada técnica bajo distintas condiciones ambientales. El modelo entrenado con variaciones en el contraste (Aumentado 4) logró el mejor rendimiento global en términos de *recall at 1* (R@1) con un 88.08 %, superando al modelo base por 1.51 %. Esta mejora fue particularmente significativa en condiciones soleadas, donde alcanzó un 81.55 % frente al 77.25 % del modelo base. Estos resultados sugieren que el entrenamiento con imágenes de diferente contraste permite al modelo identificar características más robustas ante las variaciones de iluminación típicas de escenarios interiores con iluminación natural y artificial. Por otra parte, el aumento mediante rotaciones (Aumentado 6) mostró el mejor desempeño en términos de R@1 % a nivel global (96.83 %), así como en condiciones soleadas (95.88 %). Esto indica que la invariancia a la orientación proporcionada por este tipo de aumento mejora significativamente la capacidad del modelo para identificar correctamente un lugar entre sus candidatos más probables, incluso si no lo posiciona siempre como primera opción. El aumento mediante sombras artificiales (Aumentado 2) también presentó resultados notables, con un R@1 global del 87.55 % y buenos resultados en condiciones soleadas. Esto refuerza la idea de que entrenar el modelo con efectos que simulan sombras puede ayudar a mejorar su capacidad para reconocer lugares en entornos con iluminación variable. Curiosamente, el modelo base obtuvo los mejores resultados en condiciones nubladas en términos de R@1 (90.06 %) y R@1 % (95.65 %), mientras que la modificación de brillo/oscuridad general (Aumentado 3) resultó más efectiva para escenarios nocturnos (R@1 de 93.31 %). Por otro lado, el aumento de saturación (Aumentado 5) mostró un rendimiento intermedio, con un R@1 global del 87.16 % y un R@1 % del 95.39 %. Aunque no alcanzó los mejores resultados, su desempeño en condiciones soleadas (78.67 %) fue notablemente mejor que el del modelo base. Para finalizar, el aumento mediante focos de luz (Aumentado 1) mostró un rendimiento ligeramente inferior al del modelo

base en términos de R@1 (86.25 % frente a 86.57 %), pero logró un R@1 % del 95.79 %, lo que indica que este tipo de aumento también puede ser beneficioso para mejorar la robustez del modelo ante variaciones de iluminación.

En resumen, los resultados indican que la elección de la técnica de *data augmentation* es crucial para mejorar el rendimiento del modelo en la tarea de reconocimiento visual de lugares. Las técnicas que introducen variaciones en el contraste y la orientación son especialmente efectivas para aumentar la robustez del modelo ante las variaciones de iluminación y orientación típicas en entornos interiores.

3.8 Experimentos del método global

3.8.1 Estudio sobre la influencia de la arquitectura de extracción de características

En esta sección, se presenta un estudio sobre la influencia de la arquitectura de extracción de características. Se evalúan diferentes arquitecturas de CNN como AlexNet, VGG16, ResNet-152, ResNeXt-101 64X4d, MobileNetV3, EfficientNetV2 y ConvNeXt Large. Como punto de partida, todos estos modelos se inicializan con pesos preentrenados en el conjunto de datos ImageNet Large Scale Visual Recognition Challenge [169]. Para la agregación de características se utilizan inicialmente las capas completamente conectadas (FC) definidas en la Sección 3.4.2. Estas arquitecturas se reentrenan utilizando el conjunto sin aumento de datos, con una probabilidad del 50 % de escoger un par de ejemplos cercano y lejano.

La Tabla 3.7 presenta una evaluación exhaustiva de diferentes arquitecturas para la tarea de localización visual global, empleando métricas de *Recall at 1* (R@1) y *Recall at 1 %* (R@1 %) bajo diversas condiciones ambientales. De nuevo, VGG16 emerge como la arquitectura más efectiva, superando consistentemente al resto de modelos en todas las condiciones ambientales y métricas evaluadas. Alcanza un R@1 global del 61.99 % y R@1 % del 82.88 %. Este rendimiento superior de VGG16 es particularmente notable considerando que es una arquitectura comparativamente más antigua que la mayoría de los modelos evaluados. Además, las condiciones lumínicas impactan significativamente el rendimiento de todos los modelos, observándose una degradación sustancial en condiciones soleadas. Por ejemplo, mientras VGG16 mantiene un R@1 de 71.95 % y 75.25 % en condiciones nubladas y nocturnas respectivamente, este valor cae drásticamente a un 38.79 % en condiciones soleadas. Este patrón se repite en todas las arquitecturas, indicando que las variaciones abruptas de iluminación representan un desafío considerable para los sistemas de localización visual. Sorprendentemente, arquitecturas más modernas y complejas que VGG16, como ResNet-152, ResNeXt-101 y MobileNetV3 muestran resultados marcadamente inferiores (36.60 %, 36.68 % y 37.69 % de R@1 global, respectivamente). Esto sugiere que la profundidad y complejidad adicionales de estas redes no necesariamente se traducen en mejor rendimiento para tareas de localización global basada en características visuales. AlexNet, aunque es una arquitectura más antigua, muestra un rendimiento relativamente bueno con un R@1 global de 54.33 %, superando a ResNet-152, ResNeXt-101 y MobileNetV3, pero aún por debajo de VGG16. ConvNeXt Large, a pesar de ser una arquitectura mucho

Modelo *backbone*	Nublado		Noche		Soleado		Global	
	R@1	R@1 %	R@1	R@1 %	R@1	R@1 %	R@1	R@1 %
AlexNet	70.44	87.55	66.57	87.51	25.97	40.35	54.33	71.81
VGG16	71.95	**90.44**	75.25	93.20	**38.79**	**65.00**	**61.99**	**82.88**
ResNet-152	48.75	68.25	50.13	71.19	10.93	19.63	36.60	53.02
ResNeXt-101 64X4d	48.82	73.14	49.02	75.84	12.20	30.75	36.68	59.91
MobileNetV3 Large	48.86	71.64	53.27	74.47	10.93	20.01	37.69	55.37
EfficientNetV2 L	72.41	89.21	**77.84**	**96.53**	31.46	51.75	60.57	79.16
ConvNeXt Large	**75.41**	90.06	75.14	94.09	23.23	41.11	57.93	75.08

Tabla 3.7: Estudio sobre la influencia de diferentes arquitecturas de extracción de características para el reconocimiento global de lugares, evaluadas con un umbral de 0.5 m para R@1 y R@1 % y bajo tres condiciones de iluminación distintas: nublado, noche, soleado y en conjunto.

más reciente, presenta un R@1 global de 57.93 %, lo que indica que, aunque es más robusto que ResNet-152, ResNeXt-101, MobileNetV3 y AlexNet, aún no alcanza el rendimiento de VGG16. Finalmente, EfficientNetV2 ocupa el segundo lugar en términos de R@1 y R@1 % a nivel global, con un 60.57 % y 79.16 % respectivamente. Sin embargo, su rendimiento es notablemente inferior al de VGG16 en condiciones soleadas, donde alcanza un R@1 de 31.46 % con respecto al 38.79 % de VGG16. Esto sugiere que, aunque EfficientNetV2 es una arquitectura eficiente y efectiva, aún no logra superar a VGG16 en términos de localización visual global.

3.8.2 Estudio sobre la influencia de la arquitectura de agregación de características

En esta sección se presenta un análisis de la arquitectura de agregación de características a partir del modelo VGG16. En concreto, se evalúan las técnicas de agregación de características presentadas en la Sección 3.4.2:

- **VGG16-FC:** VGG16 como red de extracción de características más tres capas completamente conectadas (FC) de tamaño 500, 500 y 5 neuronas.
- **VGG16-GeM:** VGG16 como red de extracción de características más una capa GeM (*Generalized Mean Pooling*) con 512 neuronas.
- **VGG16-1x1GeM:** VGG16 como red de extracción de características más una capa convolucional 1x1 con 512 neuronas y una capa GeM (*Generalized Mean Pooling*).

Al igual que antes, estas arquitecturas se reentrenan sin utilizar el aumento de datos. Además, durante el entrenamiento, la probabilidad de escoger un par de ejemplos cercano y lejano es del 50 %. La Tabla 3.8 presenta una evaluación exhaustiva de diferentes capas de agregación de características para la tarea de reconocimiento global, empleando métricas de *Recall at 1* (R@1) y *Recall at 1 %* (R@1 %) bajo diversas condiciones de iluminación. Los resultados revelan diferencias significativas en el rendimiento según el método de agregación empleado. En dichos resultados, la arquitectura VGG16-1x1GeM destaca como la más efectiva en todas las métricas y condiciones evaluadas, alcanzando unos resultados globales de R@1 del 81.37 % y R@1 % del 95.28 %. Este

Modelo Agregación	Nublado		Noche		Soleado		Global	
	R@1	R@1 %	R@1	R@1 %	R@1	R@1 %	R@1	R@1 %
VGG16-FC	71.95	90.44	75.25	93.20	38.79	65.00	61.99	82.88
VGG16-GeM	84.81	96.72	89.80	98.26	41.91	63.43	72.17	86.14
VGG16-1x1GeM	**90.21**	**97.84**	**91.87**	**99.26**	**62.02**	**88.74**	**81.37**	**95.28**

Tabla 3.8: Estudio sobre la influencia de diferentes capas de agregación de características para el reconocimiento global de lugares, evaluadas con un umbral de 0.5 m para R@1 y R@1 % y bajo tres condiciones de iluminación distintas: nublado, noche, soleado y en conjunto.

desempeño excepcional puede atribuirse a la combinación de una capa convolucional 1x1, que reduce la dimensionalidad manteniendo la relevancia espacial y favoreciendo la interacción entre canales. La superioridad de esta arquitectura es particularmente notable en condiciones soleadas, donde logra un R@1 del 62.02 %, superando ampliamente a las otras configuraciones. Por otro lado, la arquitectura VGG16-GeM que utiliza directamente GeM *Pooling* sin la convolución 1x1, muestra un rendimiento intermedio con un R@1 global del 72.17 % y un R@1 % global del 86.14 %. Aunque presenta buenos resultados en condiciones nubladas y nocturnas (84.81 % y 89.80 % de R@1, respectivamente), su rendimiento decae considerablemente en condiciones soleadas (41.91 %), lo que indica una menor robustez frente a variaciones de iluminación intensas. Finalmente, la arquitectura VGG16-FC, basada en capas completamente conectadas, muestra el rendimiento más bajo con un R@1 global del 61.99 % y un R@1 % global del 82.88 %. Esta arquitectura es particularmente sensible a las condiciones soleadas, donde sólo alcanza un R@1 del 38.79 %, lo que sugiere una limitada capacidad para generalizar ante cambios significativos en la iluminación.

Estos resultados evidencian que la incorporación de una capa convolucional 1x1 seguida de GeM *Pooling* proporciona la mejor estrategia para agregar características en el contexto del reconocimiento visual de lugares, ofreciendo descriptores más robustos y discriminantes que las otras alternativas evaluadas. De aquí en adelante, se empleará el modelo VGG16-1x1GeM para el reconocimiento global de lugares y se referirá a él como Siamese VGG16.

3.8.3 Estudio sobre la influencia del balance de ejemplos positivos y negativos durante el entrenamiento

El análisis presentado en esta sección evalúa el impacto del balance entre ejemplos positivos y negativos durante el entrenamiento de una Red Neuronal Siamesa para la tarea de reconocimiento visual de lugares. Se analizan nueve configuraciones diferentes, variando la proporción de ejemplos positivos desde un 10 % (s10) hasta un 90 % (s90):

- **s10:** 10 % de ejemplos positivos y 90 % de ejemplos negativos.
- **s20:** 20 % de ejemplos positivos y 80 % de ejemplos negativos.
- **s30:** 30 % de ejemplos positivos y 70 % de ejemplos negativos.
- **s40:** 40 % de ejemplos positivos y 60 % de ejemplos negativos.
- **s50:** 50 % de ejemplos positivos y 50 % de ejemplos negativos.

- **s60:** 60 % de ejemplos positivos y 40 % de ejemplos negativos.
- **s70:** 70 % de ejemplos positivos y 30 % de ejemplos negativos.
- **s80:** 80 % de ejemplos positivos y 20 % de ejemplos negativos.
- **s90:** 90 % de ejemplos positivos y 10 % de ejemplos negativos.

Al igual que antes, estas arquitecturas se reentrenan utilizando el conjunto sin aumento de datos. Dado que en este caso el entrenamiento funciona con parejas de imágenes, exploramos si es conveniente que durante el entrenamiento la red disponga de un mayor número de pares disimilares, para capturar mejor las diferencias entre las regiones que componen el entorno, o de pares similares. Lo esperable es que sea conveniente un mayor número de pares disimilares, puesto que hay muchas más posibilidades de pares disimilares tanto entre habitaciones distintas como entre zonas más o menos alejadas de una misma habitación, y parece conveniente que la red tenga información de todas estas posibles diferencias para resolver adecuadamente el problema de reconocimiento de lugares. La Tabla 3.9 presenta un análisis detallado del impacto del balance entre ejemplos positivos y negativos durante el entrenamiento de las Redes Neuronales Siamesas para el reconocimiento global de lugares. Los resultados muestran patrones significativos que merecen un análisis minucioso.

Los resultados de la Tabla 3.9 revelan una tendencia clara: un menor porcentaje de ejemplos positivos (y consecuentemente un mayor porcentaje de ejemplos negativos) durante el entrenamiento conduce a un mejor rendimiento general. El balance s10, con sólo un 10 % de ejemplos positivos y un 90 % de negativos, alcanza los mejores resultados globales tanto en R@1 (82.88 %) como en R@1 % (95.96 %). Esta configuración demuestra ser particularmente efectiva en condiciones soleadas, logrando un R@1 del 65.33 % y un R@1 % del 90.63 %, superando significativamente a las demás configuraciones en este escenario desafiante. En condiciones nocturnas, s10 también destaca con un R@1 del 93.02 % y un R@1 % del 99.37 %, lo que indica que un mayor énfasis en ejemplos negativos durante el entrenamiento mejora la capacidad discriminativa de la red para identificar correctamente lugares en entornos con baja iluminación. Por otro lado, en condiciones nubladas, aunque s60 logra el mejor R@1 (90.33 %) y s90 el mejor R@1 % (98.11 %), las diferencias son marginales respecto a s10, lo que sugiere que el balance tiene un impacto menos significativo en este escenario. Es notable observar la degradación gradual del rendimiento a medida que aumenta la proporción de ejemplos positivos, particularmente en condiciones soleadas, donde el R@1 disminuye consistentemente desde 65.33 % con s10 hasta 60.93 % con s90. Este patrón sugiere que un entrenamiento con mayor proporción de ejemplos negativos favorece la robustez del modelo frente a las variaciones de iluminación, probablemente porque incentiva a la red a aprender características más discriminantes y menos dependientes de condiciones específicas de iluminación.

En resumen, la tendencia observada puede explicarse por la naturaleza del problema de reconocimiento de lugares: al entrenar con más ejemplos negativos, la red aprende a distinguir mejor entre lugares diferentes, incluso cuando comparten características visuales similares. Esta capacidad discriminativa resulta crucial en entornos reales donde pequeños cambios en la iluminación pueden modificar significativamente la apariencia

Balance	Nublado		Noche		Soleado		Global	
	R@1	R@1 %	R@1	R@1 %	R@1	R@1 %	R@1	R@1 %
s10	90.29	97.88	**93.02**	**99.37**	**65.33**	**90.63**	**82.88**	**95.96**
s20	90.29	**98.03**	92.57	99.34	64.47	90.26	82.45	95.88
s30	90.10	98.00	92.21	99.30	63.58	89.36	81.96	95.55
s40	90.13	97.88	92.06	99.26	62.68	89.03	81.62	95.39
s50	90.21	97.84	91.87	99.26	62.02	88.74	81.37	95.28
s60	**90.33**	97.96	91.50	99.26	61.31	88.79	81.05	95.34
s70	90.48	97.96	91.13	99.26	61.02	88.74	80.88	95.32
s80	90.25	97.84	91.13	99.26	61.02	88.79	80.80	95.30
s90	90.13	98.11	91.69	99.19	60.93	87.61	80.92	94.97

Tabla 3.9: Estudio sobre la influencia del balance de ejemplos positivos y negativos utilizados en el entrenamiento de Siamese VGG16 para el reconocimiento global de lugares, evaluando cada configuración con un umbral de 0.5 m para R@1 y R@1 % y bajo tres condiciones de iluminación distintas: nublado, noche, soleado y en conjunto.

Aumento	Nublado		Noche		Soleado		Global	
	R@1	R@1 %	R@1	R@1 %	R@1	R@1 %	R@1	R@1 %
Base	**90.29**	**97.88**	93.02	**99.37**	65.33	90.63	82.88	95.96
Aumento 1 (Focos de luz)	88.67	96.84	93.68	99.34	**71.24**	**95.03**	**84.53**	**97.07**
Aumento 2 (Sombras)	86.44	97.53	**93.90**	99.00	60.50	86.75	80.28	94.43
Aumento 3 (Brillo/Oscuridad)	86.47	97.15	93.79	98.85	56.53	80.13	78.93	92.05
Aumento 4 (Contraste)	88.02	97.42	93.57	98.82	57.10	80.27	79.56	92.17
Aumento 5 (Saturación)	88.09	97.30	93.54	98.67	53.93	76.16	78.52	90.71
Aumento 6 (Rotaciones)	87.71	97.19	93.65	98.71	54.82	76.49	78.73	90.79

Tabla 3.10: Estudio sobre la influencia de los diferentes efectos de aumento de datos utilizados en el entrenamiento de Siamese VGG16 para el reconocimiento global de lugares, evaluados con un umbral de 0.5 m para R@1 y R@1 % y bajo tres condiciones de iluminación distintas: nublado, noche, soleado y en conjunto.

visual de un mismo lugar.

3.8.4 Estudio sobre la influencia del aumento de datos

Finalmente, se evalúa de nuevo el impacto del aumento de datos, pero en este caso, al entrenar una Red Neuronal Siamesa para la tarea de reconocimiento global de lugares. Ahora, la probabilidad de ejemplos similares es del 10 % y la de ejemplos diferentes es del 90 %, puesto que es la mejor configuración del estudio anterior.

La Tabla 3.10 presenta un análisis exhaustivo del efecto de diferentes técnicas de aumento de datos en el rendimiento de la arquitectura siamesa para localización visual. En este sentido, el uso de focos de luz artificiales (Aumento 1) emerge como la estrategia de aumento más efectiva a nivel global, alcanzando un R@1 del 84.53 % y un R@1 % del 97.07 %. Este rendimiento superior representa una mejora significativa respecto al modelo sin aumento de datos (Base), particularmente en condiciones soleadas donde logra un 71.24 % de R@1, superando al modelo base por casi 6 puntos porcentuales.

Curiosamente, mientras que la adición de sombras artificiales (Aumento 2) muestra el mejor desempeño en términos de R@1 % para condiciones nubladas (97.53 %), su rendimiento global (80.28 % R@1) es inferior al del modelo base. Este comportamiento indica que, si bien las sombras artificiales ayudan a mejorar la robustez bajo ciertas condiciones específicas, pueden introducir sesgos que afectan negativamente al rendimiento en otros escenarios. Las técnicas de aumento basadas en modificaciones globales de la imagen, como brillo/oscuridad (Aumento 3), contraste (Aumento 4) y saturación (Aumento 5), muestran un rendimiento similar entre sí pero considerablemente inferior al modelo base en condiciones soleadas. Este fenómeno sugiere que estas transformaciones no capturan adecuadamente la complejidad de las variaciones de iluminación que ocurren en entornos con luz natural y artificial. Sorprendentemente, el aumento mediante rotaciones (Aumento 6), que suele ser muy efectivo en tareas de clasificación de imágenes, presenta un rendimiento inferior al esperado (78.73 % R@1 global).

3.9 Comparación del método jerárquico mediante arquitecturas de clasificación con el método global con arquitecturas de red siamesa

En esta sección se presenta una comparación entre el método jerárquico basado en arquitecturas de clasificación y el método global basado en arquitecturas siamesas. La Tabla 3.11 muestra los resultados de ambos enfoques en términos de *Recall at 1* (R@1) y *Recall at 1 %* (R@1 %) bajo diferentes condiciones ambientales y por primera vez en este capítulo, en entornos nunca vistos antes. Es decir, las imágenes de estos entornos no han sido utilizadas durante el entrenamiento de la red. Esta comparación es crucial para evaluar la efectividad de cada enfoque en la tarea de reconocimiento visual de lugares.

Cabe destacar que el método jerárquico por medio de arquitecturas de clasificación no se puede evaluar directamente en los escenarios no presentados durante el entrenamiento, ya que la arquitectura de clasificación no está diseñada para clasificar estancias en entornos no entrenados. Por lo tanto, los resultados de este método se presentan con un guion (-) en la tabla (Single VGG16 Jerárquico). Sin embargo, como el método jerárquico se basa en dos etapas, la primera de clasificación y la segunda de descripción, se puede eludir esta limitación al evaluar el modelo sin pasar por la etapa de clasificación. Es decir, los entornos desconocidos se consideran en sí mismos como una sola clase (estancia), y el modelo jerárquico se utiliza como un método global basándose únicamente en la etapa de descripción (Single VGG16 Global). Esto permite obtener resultados comparables al método global basado en arquitecturas siamesas (Siamese VGG16 Global), que sí se puede evaluar directamente en entornos desconocidos.

De acuerdo con los resultados de la Tabla 3.11, el método jerárquico (Single VGG16 Jerárquico) muestra un rendimiento notablemente robusto en el entorno Freiburg-A, particularmente en condiciones soleadas, donde alcanza un R@1 del 81.55 %, superando ampliamente a los otros métodos. Esta ventaja reside en el conocimiento aprendido de las habitaciones para restringir el espacio de búsqueda, lo que resulta particularmente beneficioso cuando la condición lumínica del mapa visual difiere en gran medida con respecto a la condición de iluminación de test. En cambio, cuando se emplea este

Método	Métrica	Freiburg-A			Freiburg-B		Saarbrücken-A		Saarbrücken-B			Global
		Nublado	Noche	Soleado	Nublado	Soleado	Nublado	Noche	Nublado	Noche	Soleado	
Single VGG16 †	R@1	89.67	93.02	**81.55**	-	-	-	-	-	-	-	-
	R@1 %	95.03	98.82	94.75	-	-	-	-	-	-	-	-
Single VGG16 ‡	R@1	**90.88**	91.21	55.16	84.46	**82.86**	74.27	**51.87**	**84.93**	**75.98**	81.88	77.35
	R@1 %	**99.50**	99.74	82.69	**91.48**	**94.21**	98.34	**82.27**	**95.22**	88.62	97.36	92.94
Siamese VGG16 ‡	R@1	89.29	**94.10**	71.24	**86.40**	82.75	**77.79**	49.25	81.10	73.79	**84.63**	**79.03**
	R@1 %	97.52	**99.78**	**95.03**	91.33	94.05	**98.53**	78.80	92.70	**89.08**	**97.82**	**93.46**

† Método jerárquico.
‡ Método global.

Tabla 3.11: Comparación de los métodos propuestos para el reconocimiento visual de lugares en diferentes entornos en términos de R@1 y R@1 %.

mismo modelo con una metodología global (Single VGG16 Global), su rendimiento disminuye drásticamente en condiciones soleadas a un R@1 del 55.16 %. Este hecho justifica la necesidad de utilizar un enfoque jerárquico para manejar variaciones de iluminación y condiciones ambientales, aprovechando el conocimiento previo sobre la estructura del entorno. Por su parte, el método global basado en arquitecturas siamesas (Siamese VGG16 Global) destaca como el más efectivo en términos globales, con un R@1 del 79.03 % y un R@1 % del 93.46 %, superando al método Single VGG16 Global por 1.68 % y 0.52 %, respectivamente. Esta superioridad se manifiesta especialmente en Freiburg-A condiciones soleadas (71.24 % R@1) y en la mayoría de los escenarios de Saarbrücken-B, donde la arquitectura siamesa logra un R@1 del 84.63 %, superando al método jerárquico por 3.08 %. Este rendimiento superior de la red siamesa sugiere que su capacidad para aprender representaciones discriminantes a partir de pares de imágenes es particularmente efectiva en entornos variados y desafiantes.

En conclusión, mientras que el método jerárquico demuestra un rendimiento superior en entornos conocidos con condiciones de iluminación variables, los métodos globales, especialmente el basado en arquitecturas siamesas, ofrecen una mayor flexibilidad y capacidad de generalización para entornos nuevos. Esta complementariedad sugiere que la elección entre un enfoque jerárquico o global debe considerar tanto el conocimiento previo disponible sobre el entorno como los requisitos específicos de la aplicación en términos de precisión y adaptabilidad.

3.10 Resultados cualitativos de la tarea de reconocimiento de lugares

En esta sección se presentan ejemplos visuales de los resultados obtenidos por el método Single VGG16 (Global) y el método Siamese VGG16 (Global). Estos ejemplos ilustran la capacidad de ambos métodos bajo diferentes condiciones ambientales y entornos nunca vistos antes.

Para evaluar visualmente el método de activaciones intermedias adaptado a global (Single VGG16 Global), se presentan ejemplos en las Figuras 3.6, 3.7, 3.8, 3.9, 3.10 y 3.11. En cuanto al método global basado en SNNs (Siamese VGG16 Global), se presentan ejemplos en las Figuras 3.12, 3.13, 3.14, 3.15, 3.16 y 3.17. En cada figura, dada una imagen de test, se predice su correspondiente imagen del mapa más cercana

en el espacio del descriptor y se comprueba si coincide con la imagen más cercana del mapa en el espacio métrico de la posición. Las posiciones del mapa se representan con puntos azules, la posición de la imagen de test con una cruz roja, la posición predicha con un círculo amarillo y la posición más cercana del mapa (es decir, la mejor predicción posible) con un anillo verde.

Respecto al método Single VGG16 (Global), en las Figuras 3.6, 3.7 y 3.8 se presentan ejemplos obtenidos en el entorno FR-A, mientras que en las Figuras 3.9, 3.10 y 3.11 se muestran ejemplos en los entornos FR-B, SA-A y SA-B, respectivamente. En cada figura se muestra un caso de éxito o fracaso del método Single VGG16 (Global). Este método funciona especialmente bien cuando las condiciones de iluminación de la imagen de test son similares a las del mapa, como se observa en las Figuras 3.6, 3.7 y 3.11. En estos casos, el método logra realizar predicciones correctas, aunque existan ligeros cambios de apariencia. Sin embargo, se observa que el método sufre para generalizar a condiciones de iluminación diferentes a las del entrenamiento, como en el caso de las imágenes capturadas en condiciones soleadas (Figuras 3.8 y 3.9) o nocturnas (Figura 3.10). En estos casos, el método presenta un rendimiento inferior, como se observa en la Figura 3.10, donde la predicción es totalmente errónea debido a un cambio de apariencia de la escena.

En cuanto al método Siamese VGG16 (Global), en las Figuras 3.12, 3.13 y 3.14 se presentan ejemplos obtenidos en el entorno FR-A, mientras que en las Figuras 3.15, 3.16 y 3.17 se muestran ejemplos de los entornos FR-B, SA-A y SA-B, respectivamente. En cada figura se muestra un caso de éxito o fracaso del método global basado en SNNs. En particular, se observa que el método es capaz de generalizar a condiciones de iluminación diferentes a las del entrenamiento, como en el caso de la imágenes capturadas en condiciones soleadas (Figuras 3.14 y 3.15), donde la predicción es correcta a pesar de la diferencia de iluminación entre la imagen de test y el mapa. En la Figura 3.14, el cambio lumínico se debe a la baja visibilidad de la escena de las imágenes que componen el mapa, ya que la iluminación artificial de la habitación no está encendida. En cambio, en la Figura 3.15 se presenta una imagen de test en condiciones soleadas, donde la iluminación de la estancia es diferente debido a que en la imagen de test la luz del sol entra por la ventana, mientras que en el mapa la iluminación es artificial. A pesar de esta diferencia, el método global basado en SNNs logra realizar una predicción correcta, lo que indica su capacidad para generalizar a condiciones de iluminación diferentes a las del entrenamiento. Por otro lado, el método también presenta casos de fallo, incluso cuando las condiciones de iluminación son similares a las del entrenamiento. Por ejemplo, en la Figura 3.12 se muestra un caso de fallo en el que la imagen de test presenta una ligera diferencia de orientación respecto a la imagen del mapa, lo que provoca una predicción errónea. Además, en otros entornos como SA-A y SA-B, los cambios lumínicos de nublado a noche son mucho más drásticos debido a la presencia de grandes cristaleras y ventanales, y a la limitada iluminación artificial (Figuras 3.16 y 3.17). En estos casos específicos, el método global basado en SNNs presenta un rendimiento inferior, como se observa en la Figura 3.17 de SA-B, donde la predicción es totalmente errónea debido a la baja visibilidad.

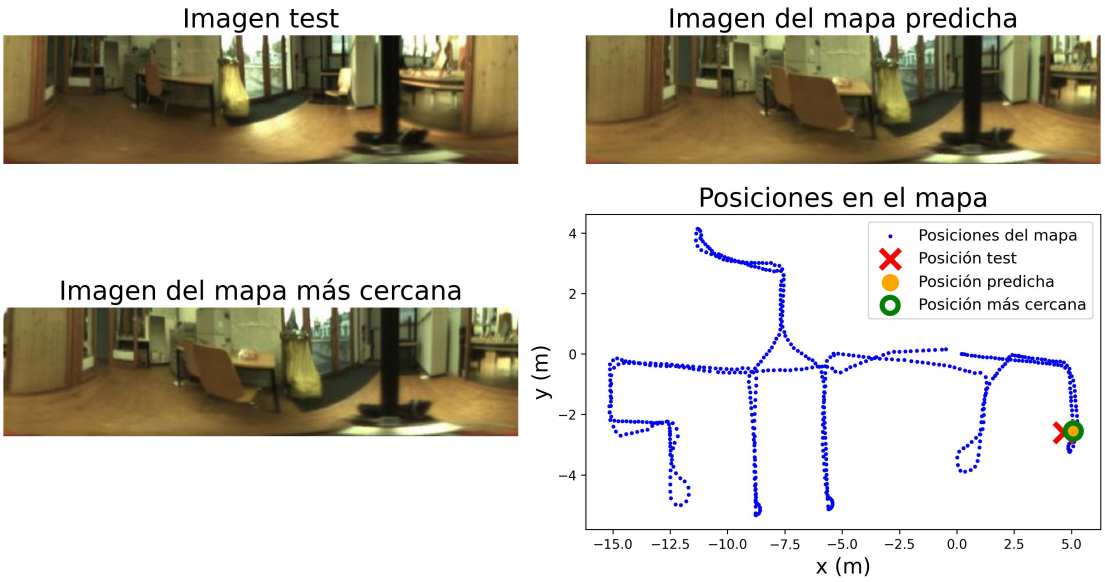


Figura 3.6: Ejemplo de predicción exitosa en condiciones nubladas, con el método Single VGG16 (Global) para FR-A.

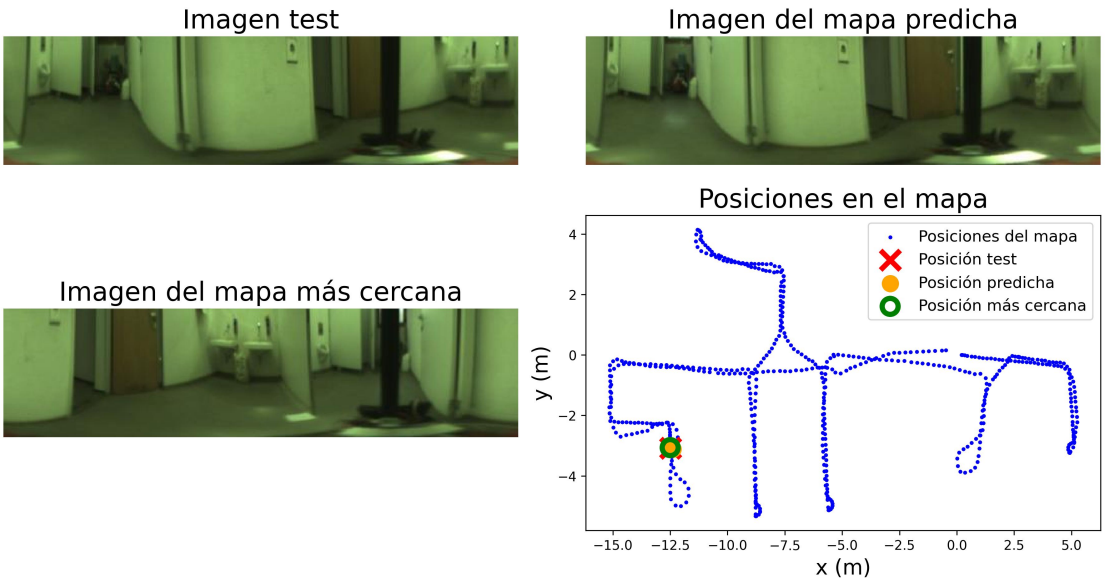


Figura 3.7: Ejemplo de predicción exitosa en condiciones nocturnas con iluminación artificial y sin perturbaciones lumínicas del exterior, con el método Single VGG16 (Global) para FR-A.

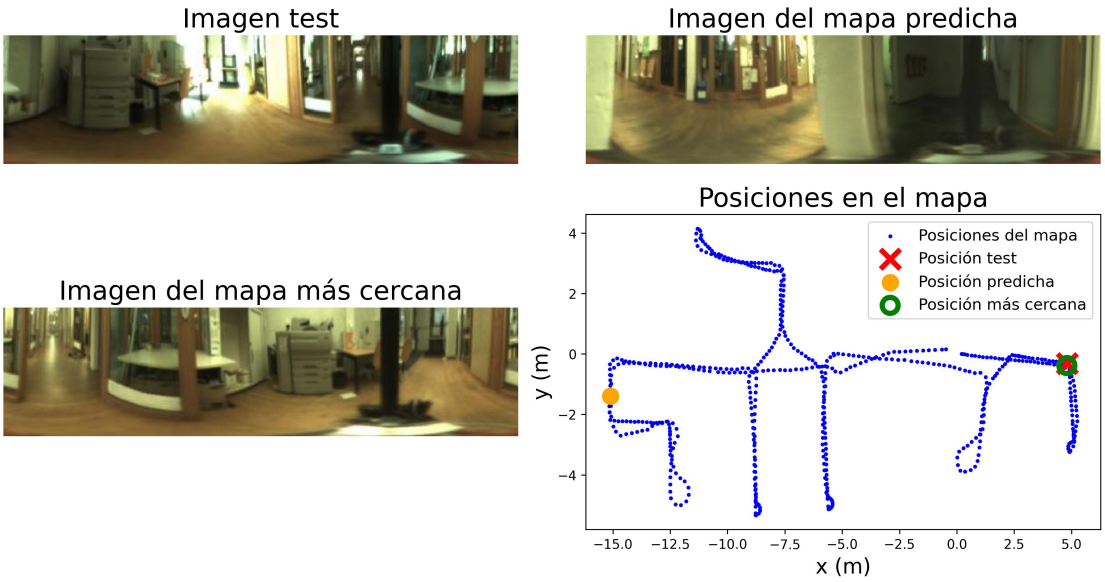


Figura 3.8: Ejemplo de predicción totalmente errónea en condiciones soleadas debido a grandes cristaleras y ventanales que permiten la entrada de luz solar, con el método Single VGG16 (Global) para FR-A.

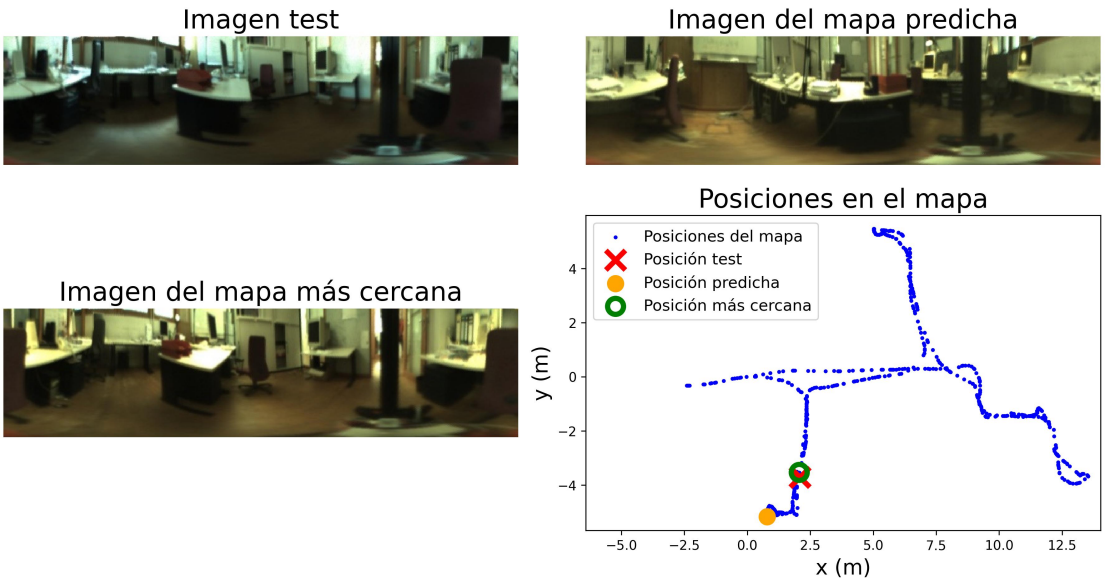


Figura 3.9: Ejemplo de predicción ligeramente errónea en condiciones soleadas debido al cambio de apariencia, con el método Single VGG16 (Global) para FR-B.

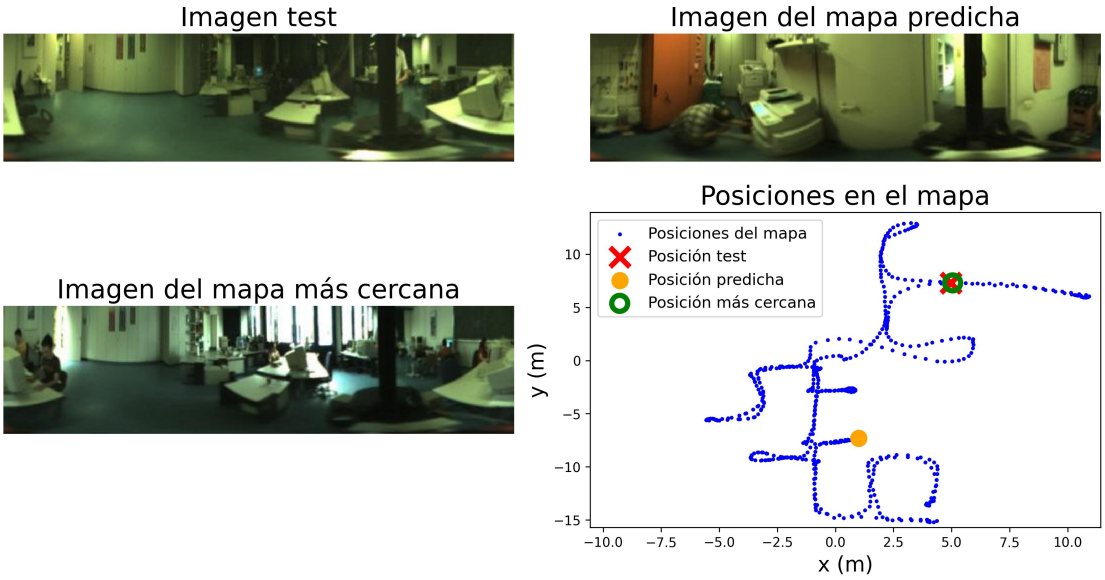


Figura 3.10: Ejemplo de predicción totalmente errónea en condiciones nocturnas, con el método Single VGG16 (Global) para SA-A.

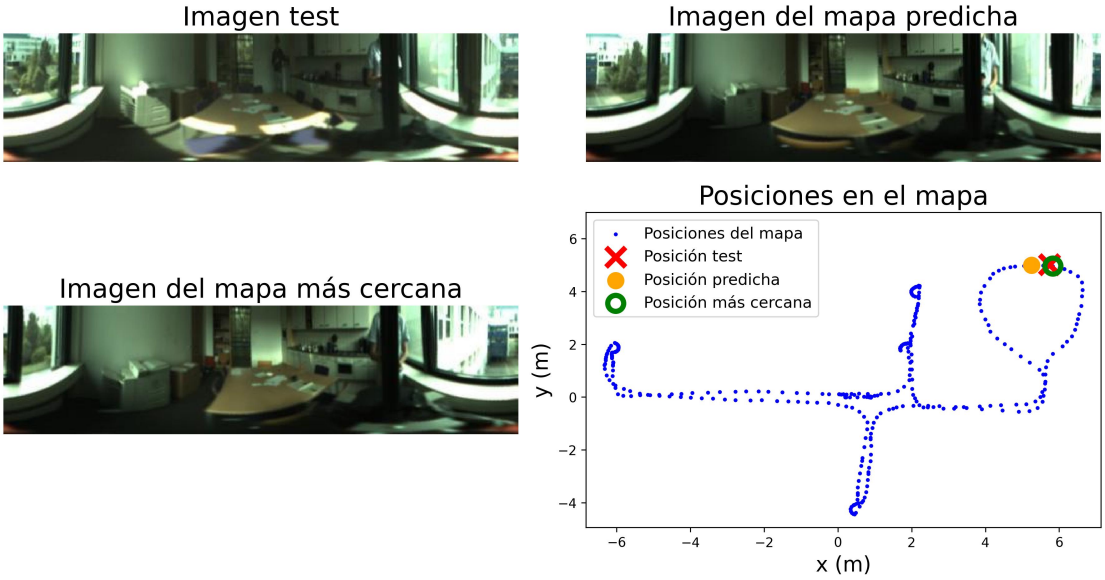


Figura 3.11: Ejemplo de predicción correcta en condiciones soleadas, con el método Single VGG16 (Global) para SA-B.

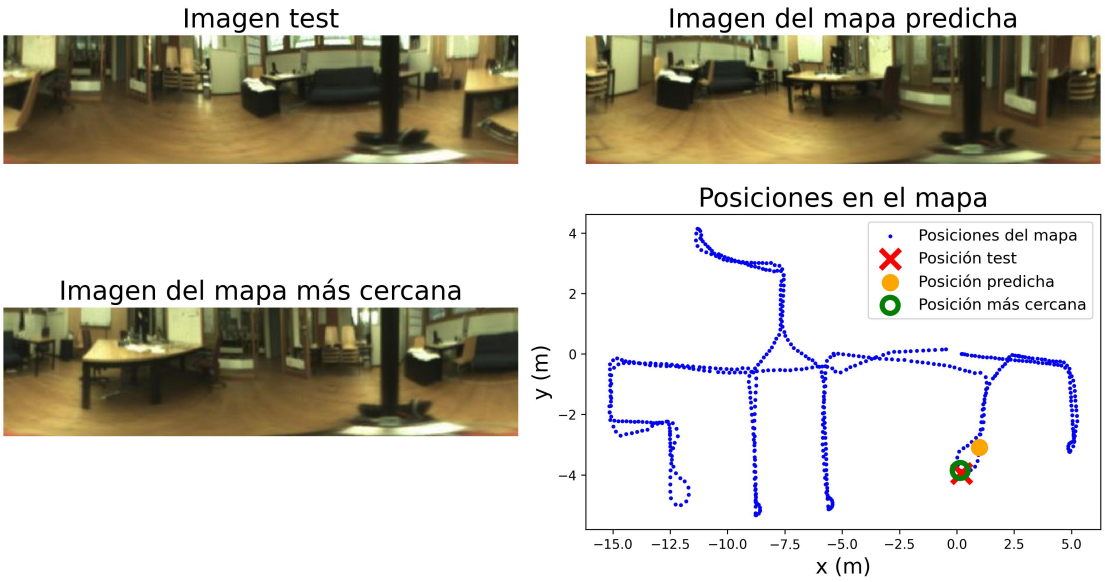


Figura 3.12: Ejemplo de predicción ligeramente errónea en condiciones nubladas, con el método Siamese VGG16 (Global) para FR-A.

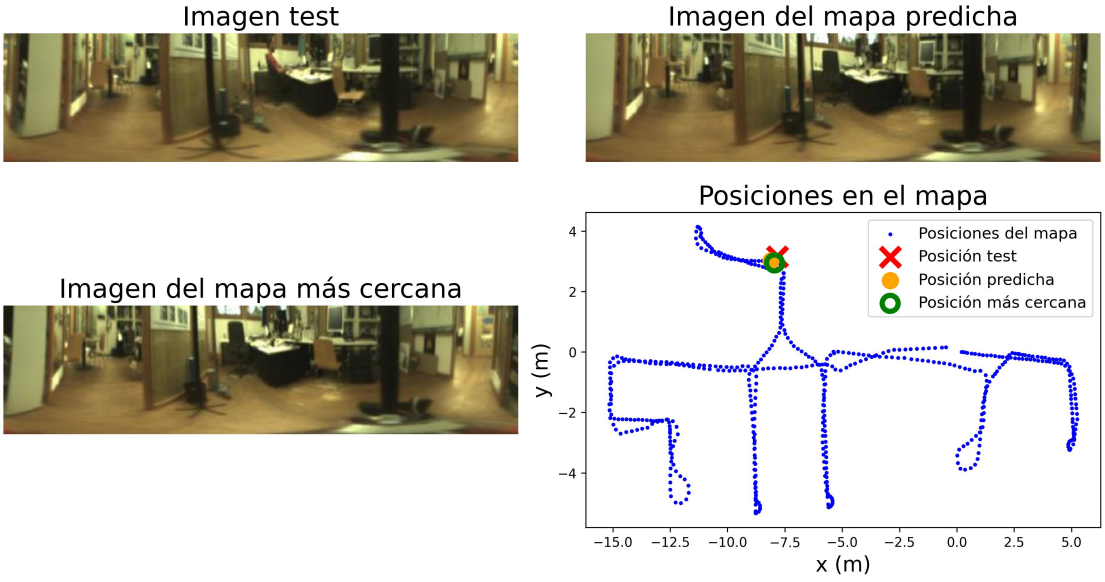


Figura 3.13: Ejemplo de predicción exitosa en condiciones nocturnas muy similares a las nubladas, con el método Siamese VGG16 (Global) para FR-A.

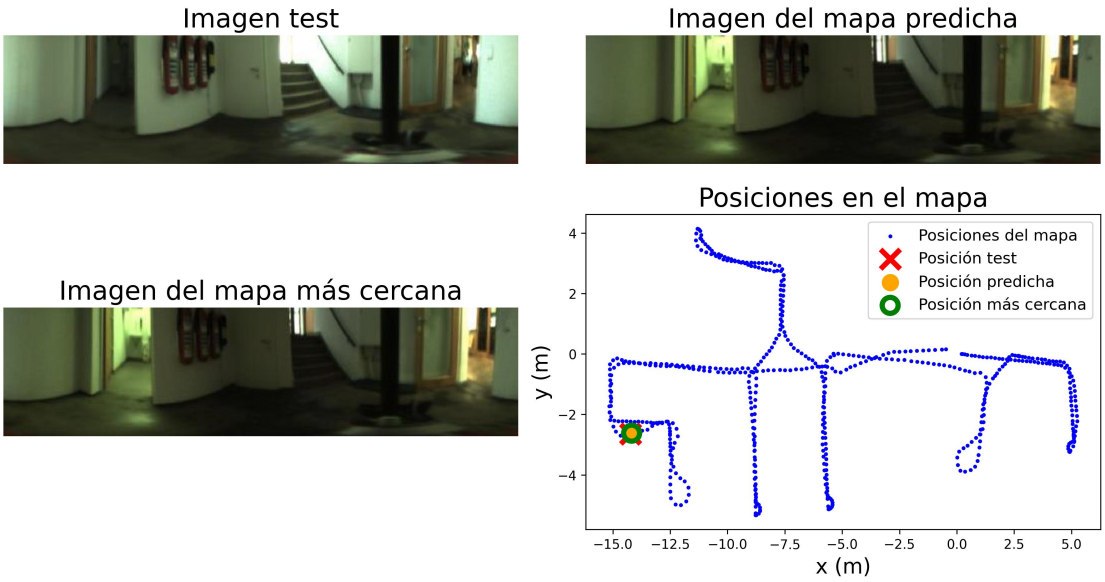


Figura 3.14: Ejemplo de predicción exitosa en condiciones soleadas pese al drástico cambio de iluminación, con el método Siamese VGG16 (Global) para FR-A.

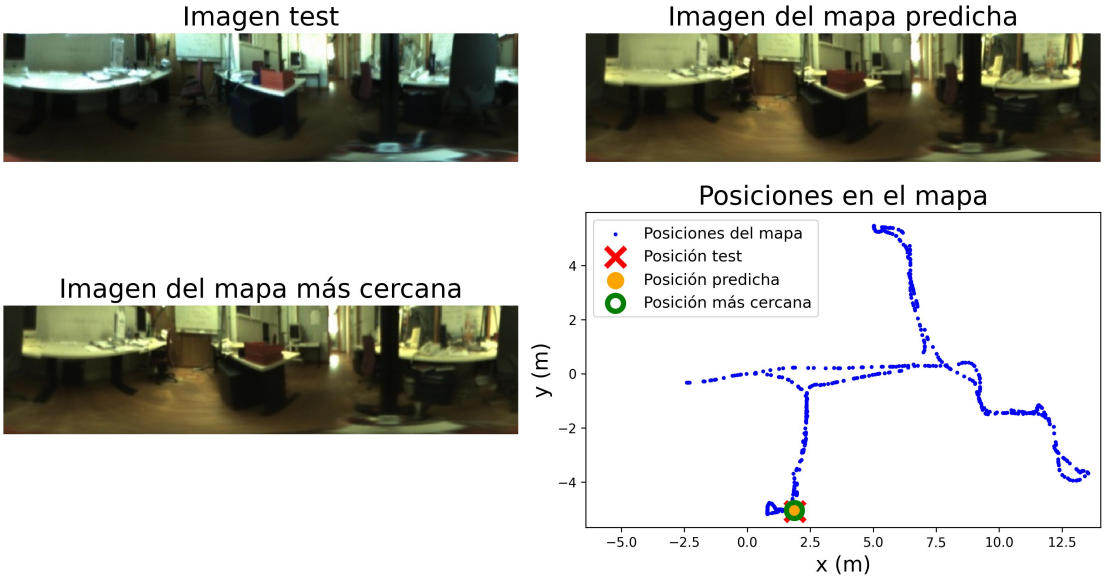


Figura 3.15: Ejemplo de predicción exitosa en condiciones soleadas pese al cambio de iluminación, con el método Siamese VGG16 (Global) para FR-B.

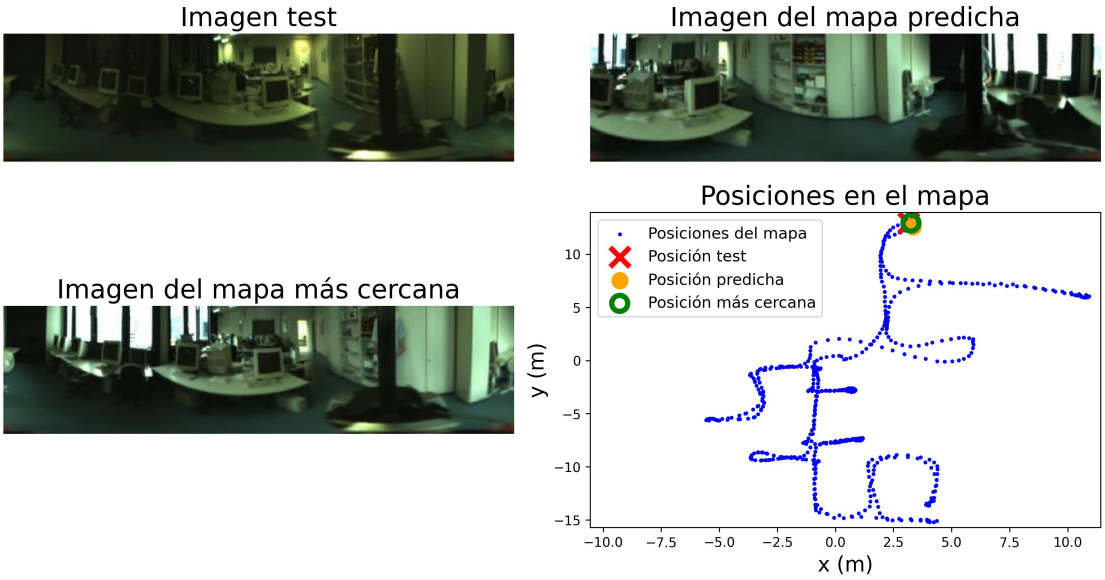


Figura 3.16: Ejemplo de predicción exitosa en condiciones nocturnas con poca visibilidad, con el método Siamese VGG16 (Global) para SA-A.

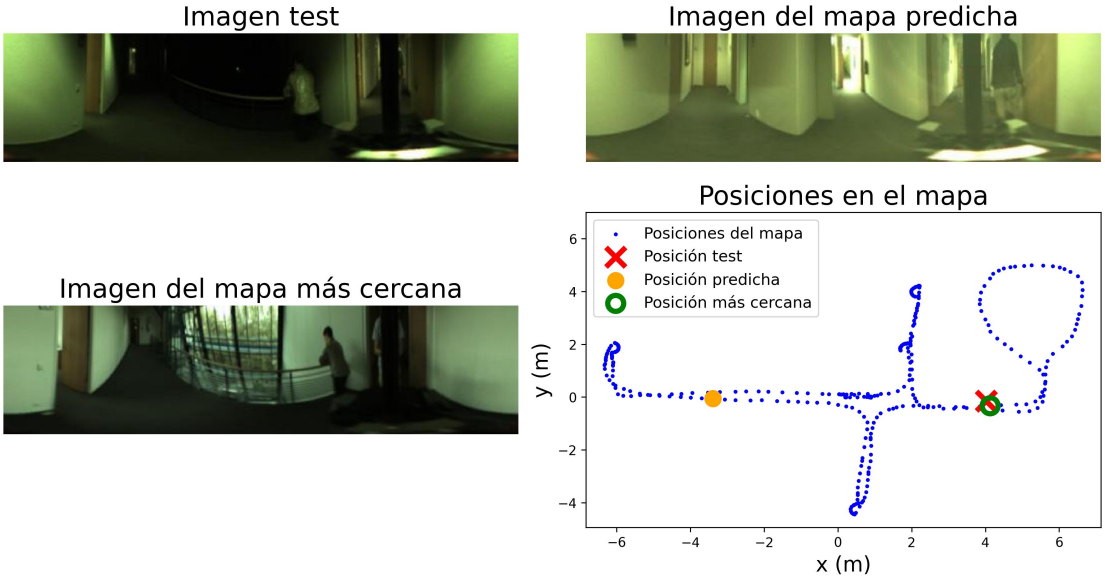


Figura 3.17: Ejemplo de predicción totalmente errónea en condiciones nocturnas debido a la baja visibilidad, con el método Siamese VGG16 (Global) para SA-B.

3.11 Conclusiones

En este capítulo se ha presentado un estudio detallado sobre el reconocimiento visual de lugares utilizando imágenes omnidireccionales. Se han explorado diferentes enfoques, incluyendo métodos jerárquicos y globales, diferentes arquitecturas de red, así como el uso de técnicas de aumento de datos y balance equilibrado del número de ejemplos utilizados durante el entrenamiento. Los resultados obtenidos demuestran que la elección del enfoque y las técnicas aplicadas tienen un impacto significativo en el rendimiento del sistema de localización.

En cuanto a las arquitecturas de red, el modelo VGG16 ha demostrado ser el más efectivo para la tarea de reconocimiento visual de lugares, abordada tanto de manera jerárquica mediante una arquitectura de clasificación (Single VGG16 Jerárquico) como de manera global mediante una arquitectura siamesa (Single VGG16 Global). En particular, el modelo VGG16 ha logrado superar a modelos más recientes y exigentes computacionalmente como ConvNeXt y EfficientNet, alcanzando un rendimiento notable en términos de *Recall at 1* (R@1) y *Recall at 1 %* (R@1 %). Este hallazgo sugiere que, a pesar de la evolución de las arquitecturas de red, VGG16 sigue siendo una opción sólida para tareas de reconocimiento visual, especialmente en entornos interiores con condiciones de iluminación variables.

Además, en cuanto al aumento de datos, se ha demostrado que el uso de técnicas específicas, como variaciones del contraste, puede mejorar significativamente el rendimiento del sistema en condiciones desafiantes. Sin embargo, el impacto que estos efectos producen en el rendimiento del sistema puede variar según la técnica utilizada. Por ejemplo, el aumento mediante focos de luz y sombras ha mostrado resultados excelentes en el método Siamese VGG16 (Global), mientras que este mismo aumento aplicado al modelo de Single VGG16 (Jerárquico) ha mostrado un rendimiento inferior al esperado. Por otro lado, el aumento mediante rotaciones debería ser beneficioso en todos los casos, sin embargo, para el modelo de Single VGG16 (Jerárquico) ha mostrado un rendimiento superior al base, mientras que para el modelo Siamese VGG16 (Global) ha mostrado un rendimiento inferior. Esto sugiere que el conjunto de datos está compuesto por trayectorias que no varían mucho en su orientación, lo que hace que el aumento mediante rotaciones no siempre resulte beneficioso.

Por otro lado, se ha observado que el método Single VGG16 (Jerárquico) supera Siamese VGG16 (Global) en entornos conocidos con condiciones de iluminación variables, con la limitación importante de que no es aplicable a entornos que no han sido previamente vistos. Por el contrario, Siamese VGG16 (Global) muestra una mayor flexibilidad y capacidad de generalización para entornos desconocidos y desafiantes. Esto sugiere que la elección entre un enfoque jerárquico o global debe considerar tanto el conocimiento previo disponible sobre el entorno como los requisitos específicos de la aplicación en términos de precisión y adaptabilidad.

En cuanto a las especificidades del método Siamese VGG16 (Global), se ha demostrado que el balance de ejemplos positivos y negativos durante el entrenamiento tiene un impacto significativo en el rendimiento del sistema. En este caso, se ha observado

que un balance del 10 % de ejemplos positivos y un 90 % de negativos produce los mejores resultados. Esto sugiere que aprender a distinguir entre lugares diferentes requiere de un mayor número de ejemplos, debido a la gran variabilidad que existe entre ellos. Además, la agregación de características obtenidas a partir de VGG16 se puede realizar mediante diferentes tipos de capas. En este sentido, se ha demostrado que una reducción de dimensionalidad del último mapa de características mediante una capa de convolución 1x1 seguida de una capa GeM *pooling* produce los mejores resultados, superando a la agregación mediante capas totalmente conectadas o mediante una única capa GeM.

En resumen, este capítulo ha proporcionado una visión integral sobre el reconocimiento visual de lugares utilizando imágenes omnidireccionales, destacando la importancia de la elección de arquitecturas de red, técnicas de aumento de datos y enfoques jerárquicos o globales. Estos hallazgos sientan las bases para futuras investigaciones en el campo del reconocimiento visual y la localización robótica, abriendo nuevas oportunidades para mejorar la precisión y robustez de los sistemas de navegación autónoma.

3.12 Publicaciones en las que se basa este capítulo

Los principales resultados que se han mostrado y descrito en este capítulo se han publicado en:

- *J.J. Cabrera, O. J. Céspedes, S. Cebollada, O. Reinoso, L. Payá.* ***An evaluation of CNN models and data augmentation techniques in hierarchical localization of mobile robots****. Publicado en la revista Evolving Systems (2024). Ed. Springer-Verlag. ISSN: 1868-6486. DOI: https://doi.org/10.1007/s12530-024-09604-6. Factor de impacto JCR 2024: 2.7, posición en el ranking JCR 2024 en la categoría "COMPUTER SCIENCE, ARTIFICIAL INTELLIGENCE": 111/204. Tercer cuartil* ***(Q3, JCR)***
  - Las contribuciones son las siguientes: este artículo presenta una evaluación de modelos de CNN y técnicas de aumento de datos para llevar a cabo la localización jerárquica de un robot móvil utilizando imágenes omnidireccionales. Se realiza un estudio sobre la influencia de diferentes modelos de CNN del estado del arte utilizados como *backbone* y se proponen diversos efectos visuales de aumento de datos para abordar la localización visual del robot. El método propuesto se basa en la adaptación y reentrenamiento de una CNN con un doble propósito: (1) realizar un paso de localización gruesa para predecir la habitación desde la que se capturó una imagen, y (2) abordar la localización fina recuperando la imagen más similar del mapa visual en la habitación predicha mediante una comparación entre pares de descriptores obtenidos de una capa intermedia de la CNN. Finalmente, se evalúa el impacto de diferentes efectos visuales de aumento de datos en el rendimiento de las CNN bajo condiciones reales de operación, incluyendo cambios en las condiciones de iluminación.
- *J.J. Cabrera, V. Román, A. Gil, O. Reinoso, L. Payá.* ***An experimental evaluation of Siamese Neural Networks for robot localization using omnidi-***

rectional imaging in indoor environments. *Publicado en la revista Artificial Intelligence Review (2024). Ed. Springer. ISSN: 1573-7462. DOI: https://doi.org/10.1007/s10462-024-10840-0. Factor de impacto JCR 2024: 13.9, posición en el ranking JCR 2024 en la categoría "COMPUTER SCIENCE, ARTIFICIAL INTELLIGENCE": 7/204. Primer cuartil* ***(Q1, JCR)***

- Las contribuciones son las siguientes: este artículo aborda el problema de la localización utilizando imágenes omnidireccionales captadas por un sistema de visión catadióptrico montado en un robot. Se explora el potencial de las Redes Neuronales Siamesas para modelar entornos interiores utilizando imágenes panorámicas como única fuente de información. Las Redes Neuronales Siamesas se utilizan para generar una función de similitud entre pares de imágenes panorámicas, lo que las hace especialmente adecuadas para tareas de recuperación de imágenes. Se evalúa su desempeño en la detección de si dos imágenes fueron capturadas en la misma habitación o en habitaciones diferentes, así como en el contexto de un problema de localización global. Los resultados superan a otras técnicas anteriores en el conjunto de datos COLD-Freiburg bajo diversas condiciones de iluminación.

- *O. J. Céspedes, S. Cebollada, J.J. Cabrera, O. Reinoso, L. Payá.* ***Analysis of Data Augmentation Techniques for Mobile Robots Localization by Means of Convolutional Neural Network***. *Publicado en Artificial Intelligence, Applications and Innovations (2023). Ed. Springer. ISBN: 978-3-031-34110-6. ISSN: 1868-4238.*

  - Las contribuciones son las siguientes: Este trabajo presenta una evaluación del uso de técnicas de aumento de datos para llevar a cabo el paso de localización gruesa dentro de un marco de localización jerárquica. Se analizan varios efectos visuales de forma individual para comprender su impacto en el rendimiento de la CNN. Los resultados permiten diseñar un aumento de datos útil para entrenar una CNN que resuelva el problema de localización en condiciones reales de operación, incluyendo cambios en las condiciones de iluminación.

4

Reconocimiento de lugares basado en LiDAR

4.1 Introducción

En muchas aplicaciones, los robots móviles deben realizar navegación autónoma en un entorno específico. A medida que se mueve, el robot debe ser capaz de reconocer o identificar diferentes áreas del entorno. Esta acción es equivalente a encontrar una correspondencia entre sus observaciones de sensores actuales y una parte de la base de datos, mapa o modelo generado previamente. Esta capacidad se denomina comúnmente reconocimiento de lugares. Para acelerar este proceso, frecuentemente, los autores se han concentrado en describir algunas partes del entorno mediante un descriptor. De esta manera, el robot debería ser capaz de reconocer una parte del entorno encontrando el descriptor en la base de datos que más se asemeja al descriptor asociado a sus observaciones actuales. El concepto de reconocimiento de lugares es de suma importancia en tareas como localización, mapeo y navegación.

Durante los últimos años, los sensores LiDAR han bajado de precio y peso, mientras que han aumentado en resolución. Por lo tanto, los sensores LiDAR permiten obtener un gran número de mediciones precisas del entorno que definen su forma y estructura. Al ser un sensor autoiluminado, es insensible a los cambios en la luz natural, por lo que es directamente aplicable a aplicaciones en exteriores. En consecuencia, han surgido nuevas aplicaciones potenciales de los sensores LiDAR en el área de la robótica móvil y, por lo tanto, es necesario centrarse en métodos que logren una descripción robusta de la escena. En la literatura, hasta ahora, podemos encontrar: a) analíticas basadas en descripciones *ad hoc*, que no necesitan entrenamiento previo [213, 214] y b) Descripciones basadas en el uso de Redes Neuronales Profundas [215], ya sea

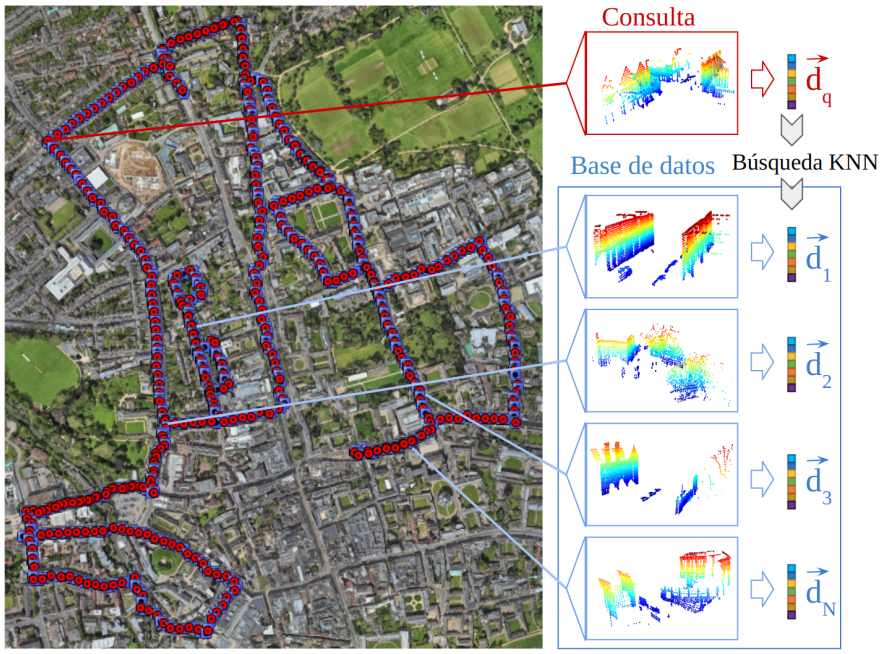


Figura 4.1: Reconocimiento de lugares basado en nubes de puntos. Cada nube de puntos de consulta (marcada con un recuadro rojo) se embebe en un descriptor global que se compara con los descriptores de las nubes de puntos de la base de datos (azul) mediante una búsqueda del vecino más cercano (*K-Nearest Neighbors*).

operando directamente sobre las coordenadas de los puntos [4] o sobre la proyección de los puntos a coordenadas de imagen [72].

4.1.1 Contribuciones de este capítulo

En este capítulo se presenta una técnica para la descripción robusta de escenas capturadas por un sensor LiDAR basada en el uso de una Red Neuronal Profunda. Se proponen varias mejoras y modificaciones partiendo de la base de varias arquitecturas recientes. Como resultado, la red propuesta es capaz de superar a todos los métodos existentes en el contexto del reconocimiento de lugares mediante LiDAR. En resumen, las principales contribuciones de este capítulo son:

- **MinkUNeXt:** una nueva Red Neuronal basada en convoluciones 3D dispersas para el reconocimiento de lugares a partir de nubes de puntos. Es el primer enfoque que utiliza una arquitectura U-Net para embeber nubes de puntos y el reconocimiento de lugares. La arquitectura ha sido desarrollada y optimizada específicamente para este problema, logrando mejoras sustanciales tanto en términos de diseño macro como micro.
- **Bloque MinkNeXt 3D:** se define un nuevo bloque residual compuesto íntegramente por convoluciones dispersas 3D, que supera el rendimiento de los bloques ResNet 3D. Este bloque incluye un cuello de botella invertido, activaciones GeLU y normalización por capas (*LayerNorm*), siguiendo la filosofía de diseño de ConvNeXt pero adaptada al dominio 3D. Esto permite procesar eficientemente datos irregulares y de alta dimensión como nubes de puntos.

- **Evaluación exhaustiva:** se realiza una evaluación exhaustiva de la arquitectura propuesta utilizando los conjuntos de datos Oxford RobotCar e In-house, mostrando que MinkUNeXt supera a otros métodos en términos de R@1 y R@1 %.
- **Estudio detallado:** se lleva a cabo un análisis detallado que justifica cada decisión de diseño tomada para la arquitectura MinkUNeXt, desde la configuración de la cardinalidad hasta el diseño del bloque residual. Este estudio valida las mejoras introducidas y su impacto en el rendimiento del modelo.

En conclusión, este capítulo presenta una arquitectura de red neuronal profunda para el reconocimiento de lugares a partir de nubes de puntos. La arquitectura propuesta, denominada MinkUNeXt, se basa en una U-Net modificada y mejorada, que utiliza un nuevo bloque residual llamado Bloque MinkNeXt 3D. Este bloque está compuesto por convoluciones dispersas 3D y sigue la filosofía de diseño de ConvNeXt, pero adaptada al dominio 3D. La topología propuesta ha demostrado ser capaz de superar significativamente el estado del arte actual en términos de *recall at 1* (R@1) y *recall at 1 %* (R@1 %), estableciendo un nuevo estándar en el reconocimiento de lugares basado en nubes de puntos.

4.2 Trabajos relacionados

Esta sección ofrece una visión general del estado del arte actual en el reconocimiento de lugares, explorando específicamente el uso de Redes Neuronales Profundas para el reconocimiento de lugares basado en nubes de puntos. En esta sección, los métodos se presentan en orden cronológico. Además, al final de este capítulo se proporciona una comparación de los principales resultados obtenidos por las arquitecturas más relevantes.

En este contexto, el primer enfoque para esta tarea se abordó en [4] con PointNetVLAD, un modelo de red basado en PointNet [75] para la extracción de características, seguido por una capa NetVLAD para la agregación de características. Las nubes de puntos que se utilizan como entrada en este tipo de arquitecturas no necesitan estar ordenadas, ya que emplean funciones simétricas como *Multilayer Perceptron* (MLP) o capas Totalmente Conectadas (*Fully Connected*). Posteriormente, un enfoque similar, llamado LPD-Net [91], mejoró el estado del arte al incorporar un bloque de extracción de características locales al inicio de la red y una agregación de vecinos basada en grafos.

Después de esto, surgió la arquitectura MinkLoc3D [84]. Se trata de una red piramidal (*Feature Piramid Network*, FPN) con convoluciones 3D dispersas para la extracción de características [11], seguida de un *Generalized Mean Pooling* (GeM) para la agregación de características en un sólo vector [216]. En ese momento, la arquitectura MinkLoc3D marcó un hito significativo, ya que superó ampliamente los métodos existentes en el estado del arte y demostró que el uso de capas convolucionales 3D era una buena elección para la extracción de características en nubes de puntos. A diferencia de las tipologías de red anteriores, al utilizar convoluciones 3D, se requiere que la nube de puntos de entrada esté ordenada, preservando las relaciones espaciales entre los

puntos. La misma situación ocurre con las imágenes, donde las convoluciones 2D han demostrado ser muy eficientes en la extracción de características gracias a las relaciones de vecindad entre píxeles. En este sentido, también han surgido algunas arquitecturas 2D que utilizan como entrada la nube de puntos proyectada en una imagen esférica (OverlapNet [72]). Otros trabajos, como [205], proponen la creación de una imagen *ad hoc* invariante a rotación: a partir de una representación en coordenadas polares de la nube de puntos, se calcula la distancia 2D entre puntos consecutivos que pertenecen al mismo ángulo de elevación (anillo) y, posteriormente, se obtiene un histograma por anillo, generando así una codificación 2D manual de la nube de puntos.

Además, algunas arquitecturas utilizan simultáneamente imágenes monoculares y nubes de puntos (MinkLoc++ [139], PIC-Net [138]). En este caso, ambas arquitecturas están formadas por dos ramas que procesan de manera independiente la imagen y la nube de puntos. Cada rama produce un vector de características y ambos vectores se agrupan finalmente en un solo vector mediante un proceso de agrupación (*pooling*). Como alternativa, cada punto puede asociarse con una característica correspondiente al valor RGB de la imagen [217]. Esto requiere una calibración precisa del sistema cámara-LiDAR. Por el contrario, algunos autores proponen utilizar la intensidad relativa devuelta por cada rayo LiDAR, como en MinkLoc-SI [218].

La arquitectura DAGC [219] fue la primera en introducir capas *self-attention* [220] para la extracción de características de nubes de puntos con el fin de realizar el reconocimiento de lugares. Posteriormente, otros autores continuaron con el uso de capas de atención, obteniendo resultados cercanos al estado del arte. En este sentido, se presentó NDT-Transformer [85], un modelo de red basado en 3 Codificadores Transformer que toma como entrada una nube de puntos modificada mediante una Transformada de Distribución Normal (NDT). Este enfoque preserva la forma geométrica de la nube de puntos mientras reduce el tamaño de ésta.

Simultáneamente, surgió PPT-Net [86], un Transformer con una distribución piramidal seguido de una capa NetVLAD. Basado en una idea similar, SOE-Net [221] extrae características locales utilizando una serie de MLPs y, posteriormente, aplica capas de atención en la agregación de esas características. Además, la red Retriever [222] también introduce capas de autoatención dentro de un autoencoder para realizar la agregación de características locales. Por otro lado, en la búsqueda de eficiencia y la implementación de estas arquitecturas en sistemas de localización en tiempo real, se presentó SVT-Net [223], un transformer 3D disperso basado en capas de convolución dispersas para la extracción de características.

Además, HiTPR [224] emplea *Farthest Point Sampling* [83] para reducir la dimensionalidad de la nube de puntos de entrada mientras preserva su información topológica original. Este trabajo introduce también un bloque Transformer para la extracción de características locales de corto alcance y otro bloque Transformer para la extracción de información global a largas distancias. Los enfoques basados en Transformer mencionados anteriormente presentaron resultados similares a los encontrados en el estado del arte. Sin embargo, la presentación de TransLoc3D [87] supuso un avance significativo.

Este capítulo presenta MinkUNeXt, una arquitectura basada en MinkUNet [11], modificada y mejorada para realizar reconocimiento de lugares a partir de nubes de puntos. Es una arquitectura de codificador-decodificador basada íntegramente en el bloque 3D MinkNeXt propuesto en la presente tesis, un bloque residual compuesto por convoluciones 3D dispersas que sigue la filosofía de ConvNeXt [10]. La extracción de características es realizada por el codificador-decodificador U-Net, y la agregación de dichas características en un único descriptor se lleva a cabo mediante *Generalized Mean Pooling* (GeM) [225]. La arquitectura propuesta demuestra que es posible superar el estado del arte actual utilizando únicamente convoluciones 3D dispersas, sin necesidad de arquitecturas más complejas como Transformers, capas de atención o convoluciones deformables. Así, este trabajo muestra que la arquitectura propuesta ofrece resultados superiores a los encontrados en la literatura, manteniendo eficiencia, escalabilidad y rendimiento.

4.3 MinkUNeXt: descripción global de nubes de puntos para reconocimiento de lugares

El reconocimiento de lugares a partir de nubes de puntos puede abordarse como una tarea de *embedding*. Para ello, es deseable contar con una arquitectura capaz de extraer las características más descriptivas de la escena y, además, agregarlas en un único vector que represente de manera general la información presente en la escena. En esta tesis se propone una solución pionera que emplea una arquitectura U-Net [226] en el contexto del reconocimiento de lugares. La mayoría de las arquitecturas similares a U-Net fueron diseñadas originalmente para la segmentación semántica, cuyo objetivo es asignar una categoría a cada píxel de una imagen de entrada o, en este caso, a cada punto de la nube de puntos de entrada. Sin embargo, la topología codificador-decodificador de una U-Net también es capaz de extraer y fusionar características relevantes de la escena, como se demostrará en la sección experimental.

4.3.1 Arquitectura Global

El modelo propuesto recibe como entrada una nube de puntos representada como un conjunto desordenado de coordenadas 3D $P = \{(x_i, y_i, z_i)\}$. Esta nube de puntos se cuantifica en un tensor disperso, que es una extensión de alta dimensión de una matriz dispersa donde los elementos no nulos se representan como un conjunto de índices C (coordenadas cuantificadas) y valores asociados (o características) F. Algunos trabajos [85, 91] proponen emplear como características ciertos atributos diseñados manualmente, como la componente vertical del vector normal, la varianza de altura, el cambio de curvatura o simplemente el valor de las coordenadas. Otros [84, 227] prefieren inicializar la característica de cada coordenada con "unos", es decir, la primera convolución (*stem*) sólo tomará como entrada características con valor de "uno" para los vóxeles no vacíos (esto es, $F = \{1\}$). Esta idea se basa en la premisa de que las coordenadas 3D son suficientes para describir la nube de puntos y que las características adicionales pueden ser aprendidas por la red durante el entrenamiento. Esta idea también se adopta en el presente trabajo, donde los datos de entrada $\hat{P} = \{(\hat{x}_i, \hat{y}_i, \hat{z}_i, 1)\}$ están compuestos por C, un conjunto de coordenadas 3D cuantificadas, y F, un vector de

"unos" cuya longitud es igual al número de puntos cuantificados.

La arquitectura global se representa en la Figura 4.2. El codificador de la red está compuesto por cinco convoluciones 3D Dispersas (coloreadas en amarillo). Entre ellas, la convolución inicial (*stem*), que preserva la dimensión de entrada de la nube de puntos, ya que su paso (*stride*) está fijado en 1 y su tamaño de *kernel* es 5. Mientras que cada una de las cuatro capas convolucionales siguientes reduce gradualmente la dimensión espacial, el campo receptivo aumenta, ya que las capas convolucionales sucesivas capturan patrones cada vez más amplios combinando la información de capas anteriores. Cada una de esas convoluciones reduce la dimensión de entrada por un factor de 2, ya que emplean un tamaño de *kernel* y *stride* de 2. Después del codificador, la dimensión de la nube de puntos de entrada se reduce por un factor de 32.

En una U-Net común, el decodificador está compuesto por cuatro Convoluciones 3D Dispersas Transpuestas, que aumentan la dimensión espacial por un factor de 2, reconstruyendo progresivamente la nube de puntos de entrada. Sin embargo, en esta arquitectura se propone reconstruir parcialmente la nube de puntos aplicando solo tres convoluciones transpuestas (coloreadas en naranja), ya que nuestro propósito es la descripción de la nube de puntos y no la segmentación semántica. La Subsección 4.4.5 justificará que las características extraídas con solo tres convoluciones transpuestas son más robustas para comprender el contexto general de la escena. Además, se aplican una Normalización por Lotes (*Batch Normalization*) y una función de activación ReLU (coloreadas en rojo) después de todas las convoluciones, lo que ayuda a estabilizar el proceso de entrenamiento. Adicionalmente, en esta arquitectura se propone emplear el Bloque Residual MinkNeXt (coloreado en azul) en lugar del Bloque ResNet convencional después de cada ReLU (excepto la correspondiente al *stem*). Este tipo de bloques residuales proporcionan un camino directo para el flujo de gradientes a través de la red, reduciendo el sobreajuste y mejorando las capacidades de generalización en datos no vistos. En esta arquitectura, también se utiliza para aumentar el número de mapas de características, como se detallará en la siguiente subsección 4.3.2.

La arquitectura U-Net se caracteriza por tener saltos entre el codificador y el decodificador. Por un lado, el codificador captura características a diferentes escalas espaciales, desde detalles finos (bajo nivel) hasta estructuras más globales (alto nivel) presentes en las nubes de puntos. Por otro lado, gracias a las conexiones de salto, el decodificador fusiona las características de bajo y alto nivel. Luego, se añade una Capa Totalmente Conectada, ya que las características de salida han demostrado ser robustas frente a cambios de punto de vista en el reconocimiento visual de lugares [228]. Además, esta Capa Totalmente Conectada también se emplea para extender los mapas de características hasta una dimensionalidad de 512. Posteriormente, los descriptores de puntos que conforman ese mapa de características se agregan en un único descriptor global mediante una *Generalized Mean Pooling* (GeM) [225].

4.3.2 Arquitectura del Bloque Residual

Como se mencionó anteriormente, en este trabajo se proponen tanto una arquitectura global como un bloque residual. En este sentido, se diseña un nuevo bloque

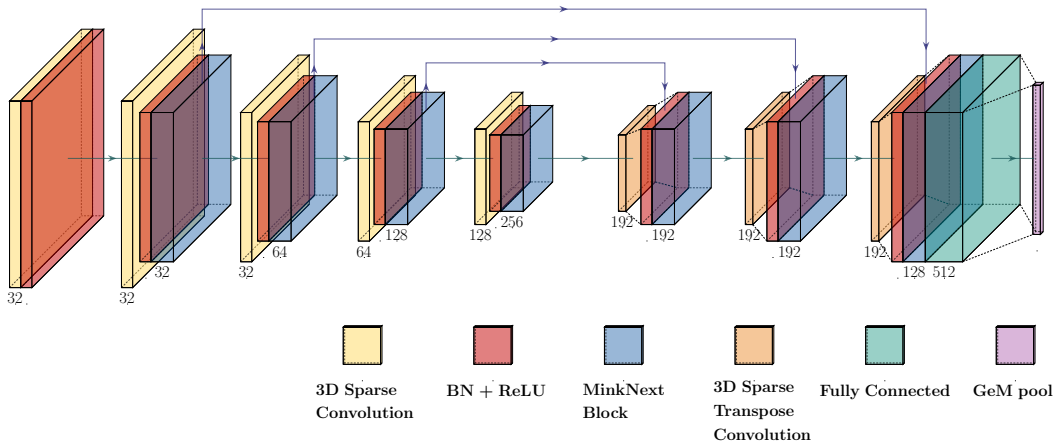


Figura 4.2: Este diagrama muestra la arquitectura del MinkUNeXt, la cual se basa en una red de segmentación semántica (U-Net) modificada y mejorada para llevar a cabo el reconocimiento de lugares a partir de nubes de puntos.

residual (Figura 4.3) compuesto íntegramente por convoluciones 3D dispersas, siguiendo la filosofía propuesta por ConvNeXt [10] y superando el rendimiento de los bloques ResNet. Hemos denominado a este bloque MinkNeXt, ya que aprovecha las ventajas de los bloques ResNet y está completamente implementado en Minkowski Engine [11].

En la arquitectura global (Figura 4.2), el bloque residual propuesto aparece en color azul después de cada función de activación ReLU (excepto en la correspondiente al *stem*). Dado que el bloque residual generalmente se emplea para aumentar el número de mapas de características (canales), la primera capa del bloque residual es una convolución 1x1x1 que amplía la dimensión de entrada hasta el tamaño de los canales de salida. Después de esto, se aplica un cuello de botella invertido (*inverted bottleneck*), expandiendo la dimensión cuatro veces y luego reduciéndola nuevamente hasta la dimensión de salida mediante dos convoluciones 3D dispersas. Este cuello de botella invertido fue propuesto originalmente por MobileNetV2 [229] y, en la actualidad, es un diseño importante en los bloques que conforman los Transformer. Además, también se aplica una convolución 1x1x1 en la conexión residual cuando las dimensiones de entrada y salida son diferentes.

La función de activación empleada en este bloque es la *Gaussian error Linear Unit(GeLU)* [230], que es más suave que la ReLU y se utiliza en los Transformadores más avanzados. Finalmente, la normalización se lleva a cabo mediante *LayerNorms* [231] en el flujo principal del bloque y mediante *BatchNorms* [232] en la conexión residual.

4.4 Experimentos

Esta sección describe los conjuntos de datos (Subsección 4.4.1), el etiquetado (Subsección 4.4.2) y el entrenamiento y evaluación de la arquitectura propuesta (Subsección 4.4.3). Posteriormente, se describen los detalles de implementación en la Subsección 4.4.4. A continuación, en la Subsección 4.4.5, presentamos un estudio sobre la influencia

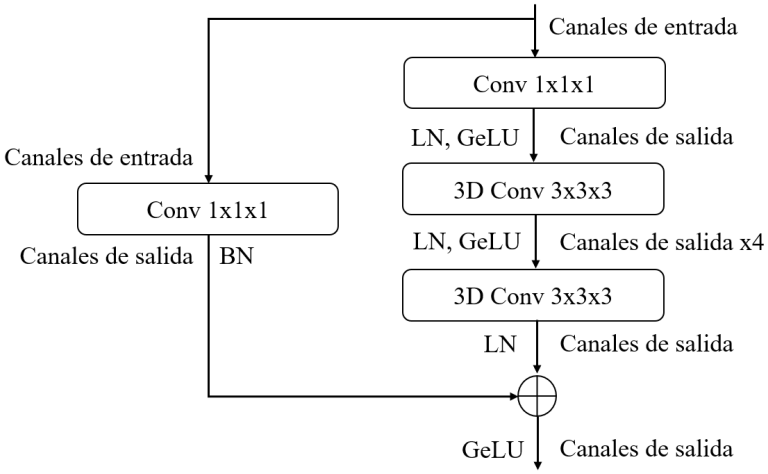


Figura 4.3: Este diagrama muestra el bloque MinkNeXt propuesto. Este bloque residual es una parte esencial de la red global, ya que aumenta el número de mapas de características a través de un cuello de botella invertido.

de los pasos de diseño llevados a cabo para obtener la arquitectura final. Finalmente, los resultados principales se comparan con otros enfoques en la literatura en la Subsección 4.4.6.

4.4.1 Conjuntos de datos

Con el fin de entrenar y evaluar el método propuesto en igualdad de condiciones, se han utilizado los conjuntos de datos y los protocolos de evaluación introducidos en [4]. Este es un marco comúnmente empleado y respetado por una gran cantidad de estudios, utilizado para comparar diferentes propuestas que abordan la tarea de reconocimiento de lugares mediante nubes de puntos. Este *benchmark* consta de 2 conjuntos de datos y 4 entornos diferentes:

- **Conjunto de Datos Oxford RobotCar [2]**. Este conjunto de datos fue generado utilizando sensores 2D SICK LMS-151 montados en un vehículo. El conjunto de datos cubre una trayectoria de 10 km a lo largo de la ciudad de Oxford. En total, se utilizan 44 secuencias de la misma trayectoria, las cuales están geográficamente divididas en conjunto de entrenamiento (70 %) y conjunto de test (30 %). Esto da como resultado 21711 submapas de entrenamiento y 3030 submapas de test.
- **Conjunto de Datos In-house [4]**. Este conjunto de datos consta de tres entornos diferentes: un Sector Universitario (U.S.), una Zona Residencial (R.A.) y un Distrito Comercial (B.D.). Estos conjuntos de datos se capturan utilizando un LiDAR Velodyne-64 montado en un vehículo motorizado que recorre cada una de las tres regiones. Las longitudes de las rutas son de 10 km, 8 km y 5 km, respectivamente. Está compuesto por 5 secuencias diferentes de cada una de las regiones U.S., R.A. y B.D., las cuales fueron capturadas en instantes diferentes. Además, cada secuencia de U.S. y R.A. está geográficamente dividida en entrenamiento y test, mientras que el entorno B.D. se utiliza únicamente para test.

En ambos conjuntos de datos, las lecturas de LiDAR se toman en intervalos regulares de 12.5 m y 25 m para el conjunto de entrenamiento y el conjunto de test, respectivamente. Además, ambos conjuntos de datos están formados por varios submapas. Cada submapa se construye capturando lecturas LiDAR consecutivas a lo largo de 20 m. A continuación, las nubes se registran y se procesan para crear un submapa consistente. Cada uno de estos submapas de entrenamiento y test se filtra eliminando el plano del suelo y también se muestrea regularmente mediante un filtro de cuadrícula para reducir su tamaño a 4096 puntos. Las coordenadas XYZ de los puntos que constituyen cada nube de puntos se trasladan y escalan para obtener una distribución de puntos con media cero en el rango [-1, 1] para cada coordenada. Cabe destacar que este procedimiento de normalización se aplica a todas las nubes de puntos, tanto de entrenamiento como de test. Este proceso de normalización era importante para tipologías de red anteriores, como PointNetVLAD [4], pero no es necesario para la arquitectura propuesta, ya que Minkowski Engine [11] no necesita normalizar las coordenadas de la nube de puntos. Sin embargo, se ha mantenido este procedimiento para garantizar la comparabilidad con otros métodos.

4.4.2 Etiquetado y similitud

Cada submapa del conjunto de datos está asociado a las coordenadas UTM de su respectivo centroide. Este constituye el identificador de cada submapa y se utiliza posteriormente durante el entrenamiento y la evaluación de la red. A continuación, definimos la similitud entre los submapas. Este concepto se denomina generalmente etiquetado en la literatura y es importante porque es necesario alimentar al modelo con submapas estructuralmente similares capturados en el mismo lugar y submapas estructuralmente diferentes provenientes de lugares diferentes. En este sentido, la mayoría de los protocolos de etiquetado propuestos se basan en la distancia euclídea de las coordenadas UTM desde las cuales se capturan las nubes de puntos (dos nubes de puntos se consideran estructuralmente similares si se capturan dentro de una distancia p y estructuralmente diferentes si se capturan desde una distancia mayor a n, donde $p < n$). Este procedimiento, por supuesto, es una aproximación que asume que los submapas capturados en la misma área tendrán una estructura similar, constituyendo así una manera sencilla pero efectiva de etiquetar los datos de entrenamiento. En este trabajo, se adopta este método con $p = 10m$ y $n = 50m$, como en la mayoría de los manuscritos referenciados. Otros autores también han propuesto otros métodos para el etiquetado de similitud en el contexto del reconocimiento de lugares. Por ejemplo, Chen *et al.* [72] proponen usar la superposición entre nubes de puntos como método alternativo para etiquetar nubes de puntos similares y diferentes. Para calcular la superposición entre dos nubes de puntos (es decir, submapas) debe realizarse un registro preciso, lo que dificulta la aplicación de esta técnica a grandes conjuntos de datos.

4.4.3 Entrenamiento y evaluación

Para el entrenamiento y evaluación del método propuesto, se han seguido los dos protocolos de evaluación establecidos en [4]:

- El primero, protocolo base, consiste en entrenar el modelo únicamente con los datos de entrenamiento de Oxford y evaluar con los datos de test de Oxford y

Tabla 4.1: Número de submapas de entrenamiento y test para los protocolos base y refinado.

	Protocolo base		**Protocolo refinado**	
	Entrenamiento	Test	Entrenamiento	Test
Oxford	21.7k	3.0k	21.7k	3.0k
In-house	-	4.5k	6.7k	1.7k

del conjunto de datos In-house (U.S., R.A. y B.D.).

- El segundo, protocolo refinado, consiste en entrenar con los datos de entrenamiento de Oxford y del conjunto de datos de entrenamiento In-house (U.S., R.A.) y evaluar con los datos de test de Oxford y el conjunto de datos de test In-house (U.S., R.A. y B.D.).

La Tabla 4.1 resume la cantidad de submapas de entrenamiento y test correspondientes a cada conjunto de datos y a cada uno de los protocolos definidos anteriormente. La evaluación de los descriptores para el reconocimiento de lugares basado en LiDAR se lleva a cabo mediante el *recall* de los K mejores candidatos. Siguiendo los métodos de evaluación más comunes (como en los trabajos citados en la Sección 4.2), se utilizan el *recall at 1* (*R@1*) y el *recall at 1 %* (*R@1 %*) para facilitar la comparación con otras técnicas.

El proceso de evaluación comienza con un "submapa actual", que es una nube de puntos extraída del conjunto de datos de test y que se compara con submapas de diferentes recorridos que cubren la misma región del mapa. Cada submapa de test es procesado por la red, que genera un vector descriptor que codifica su apariencia. Este descriptor se denomina "descriptor de test". A continuación, el descriptor de test se compara con todos los descriptores del mapa, y se selecciona la nube de puntos cuyo descriptor minimiza la distancia euclídea. Finalmente, se considera que el reconocimiento del lugar ha sido exitoso si la nube de puntos de test y la nube de puntos recuperada están a una distancia euclídea menor a 25 m.

4.4.4 Detalles de Implementación

En el presente trabajo entrenamos el modelo propuesto siguiendo el procedimiento establecido por [227]. En este sentido, se ha empleado la función de pérdida *Truncated Smooth-AP* (TSAP) la cual trata de maximizar el ranking de los *top-k* candidatos positivos.

$$\mathcal{L}_{TSAP} = \frac{1}{b}\sum_{q=1}^{b}(1 - AP_q) \tag{4.1}$$

donde b es el tamaño del lote y AP_q es la precisión promedio:

$$AP_q = \frac{1}{|P|}\sum_{i\in P}\frac{1+\sum_{j\in P, j\neq i} G(d(q,i)-d(q,j);\tau)}{1+\sum_{j\in \Omega, j\neq i} G(d(q,i)-d(q,j);\tau)} \tag{4.2}$$

Dado un submapa de referencia q, la precisión promedio AP_q se calcula a partir del conjunto de los k candidatos más cercanos P (positivos) y el conjunto de todos

Parámetro	Protocolo base	Protocolo refinado
Tamaño de Lote (b)	2048	2048
Número de épocas	400	500
Tasa de aprendizaje inicial (LR)	1×10^{-3}	1×10^{-3}
Épocas de reducción de LR	250, 350	350, 450
Decaimiento del Peso L2	1×10^{-4}	1×10^{-4}
Temperatura Sigmoide (τ)	0.01	0.01
Positivos por nube (k)	4	4
Escala de cuantificación (qs)	0.01	0.01

Tabla 4.2: Parámetros de Entrenamiento en los Protocolos base y Refinado

los positivos y negativos Ω. Además, la función G constituye una función Sigmoide $G(x;\tau) = \left(1+\exp\left(-\frac{x}{\tau}\right)\right)^{-1}$ con un parámetro τ que controla el afilamiento. El término $d(q,i)$ representa la distancia euclídea entre el descriptor de la nube de referencia q y la i-ésima nube de puntos del lote. El numerador representa el ranking de las nubes positivas i entre las top k nubes positivas (donde $k=4$), mientras que el denominador representa el ranking de una nube positiva i entre todos las demás nubes positivas y negativas.

Para el correcto funcionamiento de este tipo de función de pérdida, es necesario entrenar con un tamaño de lote elevado [227]. Específicamente se ha utilizado un tamaño de 2048 durante 400 y 500 épocas de entrenamiento para el protocolo de base y refinado, respectivamente. El optimizador utilizado para minimizar la función de pérdida es Adam con una *Initial Learning Rate* de 1×10^{-3}, y se divide por 10 en ciertas épocas dadas al planificador (*scheduler*), las cuales son las épocas 250 y 350 para el protocolo base y las épocas 350 y 450 para el protocolo refinado. La Tabla 4.2 resume todos los valores de los parámetros descritos anteriormente.

Adicionalmente, al trabajar con convoluciones dispersas, las nubes de puntos de entrada necesitan ser cuantificadas por un factor qs, que se establece en 0.01. Dado que estas nubes ya están normalizadas entre [-1, 1], se obtienen resoluciones espaciales de 200 vóxeles en cada eje de coordenadas. Para aumentar el número de instancias de entrenamiento y reducir el sobreajuste del modelo, se ha llevado a cabo un aumento de datos aplicando un *jitter* aleatorio de un valor entre [0, 0.001] individualmente a cada punto de la nube, una transformación aleatoria a la nube de puntos con un valor entre [0, 0.01] y una eliminación aleatoria del 10 % de los puntos.

Todos los experimentos se llevan a cabo en una GPU NVIDIA GeForce RTX 3090 con 24 GB. Nuestro código está disponible públicamente en la página web del proyecto https://juanjo-cabrera.github.io/projects-MinkUNeXt/.

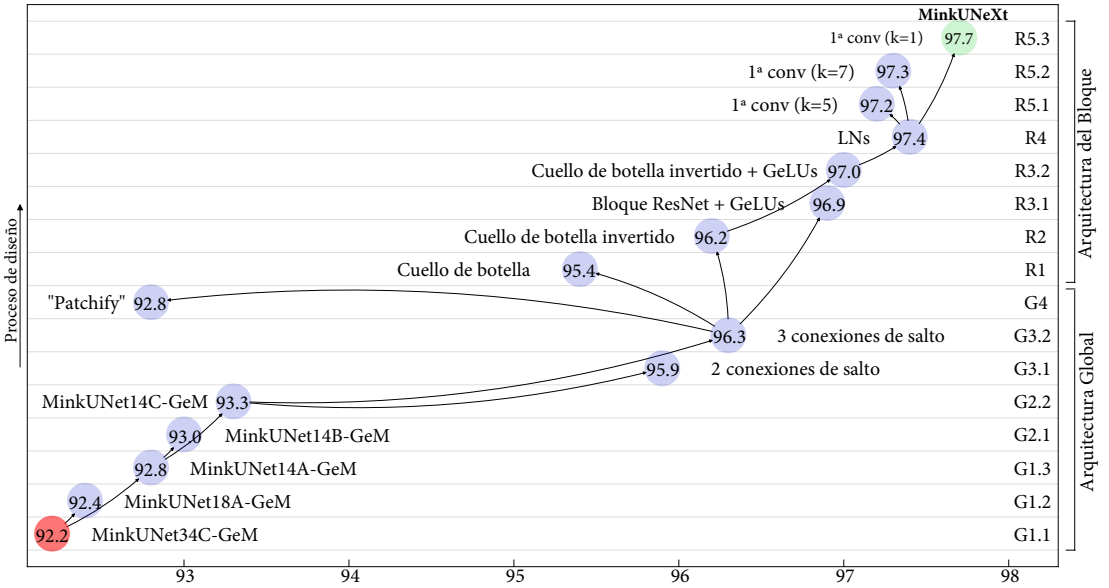


Figura 4.4: Este diagrama ilustra el progreso del diseño de la arquitectura propuesta desde MinkUNet hasta MinkUNeXt. Todas las modificaciones propuestas se resumen en la Tabla 4.3.

4.4.5 Diseño evolutivo: de MinkUNet a MinkUNeXt

El diseño parte de la arquitectura MinkUNet34C [11] como punto de partida. A continuación, se describen todas las decisiones de diseño. Cada paso de diseño se resume en dos subsecciones principales: (1) diseño global y (2) diseño del bloque residual, que se incluyen a continuación. Para cada paso, se presenta tanto el procedimiento como los resultados, comenzando desde MinkUNet34C hasta obtener la arquitectura MinkUNeXt. La evolución de la red y los resultados se presentan en la Figura 4.4. La Tabla 4.3 resume y describe los principales pasos del diseño.

4.4.5.1 Diseño Global

Como se mencionó anteriormente, el punto de partida es la arquitectura MinkUNet34C [11] y se modifica inicialmente añadiendo una capa de agregación de características (GeM). Este paso está marcado como G1.1 en la Figura 4.4. A continuación, se describe el resto de la hoja de ruta seguida hacia el diseño final. Cada uno de los pasos de diseño se clasifica en uno de los siguientes puntos: evaluación de la cardinalidad, evaluación del número de canales, cambio del número de saltos entre el codificador y el decodificador, y cambio a un *stem* con "Patchify".

G1. Evaluación de la cardinalidad. La cardinalidad se define como el número de bloques paralelos, lo que permite a la red aprender varias representaciones de entrada. En este sentido, se evalúan diferentes configuraciones de cardinalidad para cada bloque residual: (2, 3, 4, 6, 2, 2, 2, 2), (2, 2, 2, 2, 2, 2, 2, 2) y (1, 1, 1, 1, 1, 1, 1, 1), correspondientes a MinkUNet34, MinkUNet18 y MinkUNet14, respectivamente. Estos valores de cardinalidad representan el número de instancias de cada bloque residual, que aparecen en color azul en la Figura 4.2, pero

en este punto todavía son bloques ResNet. Además, estas configuraciones de cardinalidad se resumen respectivamente en los pasos G1.1, G1.2 y G1.3 de la Figura 4.4. Como se ilustra en el diagrama, reducir la cardinalidad al mínimo, sin bloques paralelos, muestra un mejor rendimiento y permite una mejora del 92.2 % al 92.8 % en términos de R@1. A partir de ahora, se usará 1 como cardinalidad de cada bloque residual.

G2. Evaluación del número de canales. El número de canales o filtros corresponde al número de mapas de características que la capa convolucional puede aprender. El número de filtros correspondiente a las capas convolucionales del codificador se fija en (32, 64, 128, 256), pero el número de canales del decodificador puede tomar los siguientes valores (128, 128, 96, 96), (128, 128, 128, 128) y (192, 192, 128, 128) correspondientes a MinkUNet14A, MinkUNet14B y MinkUNet14C. Este número de filtros del decodificador se resume respectivamente en los pasos G1.3, G2.1 y G2.2 de la Figura 4.4. El mejor resultado se obtiene con MinkUnet14C (G2.2) con un R@1 de 93.3 %. Por lo tanto, el número de filtros de las convoluciones transpuestas que se adoptarán en las variaciones posteriores de la arquitectura es (192, 192, 128, 128).

G3. Cambio del número de saltos entre el codificador y el decodificador. El modelo U-Net original se caracteriza por la presencia de 4 saltos entre el codificador y decodificador. En este sentido, en el presente trabajo se estudia el rendimiento de la red al reducir el número de saltos y la posterior eliminación de las convoluciones transpuestas tras el último salto. Además de las 4 conexiones de salto ya implementadas en las configuraciones anteriores, se han evaluado 2 y 3 saltos correspondientes a G3.1 y G3.2 en la Figura 4.4. Al reducir el número de saltos a 3 y eliminar las capas siguientes al último salto, el modelo muestra, con diferencia, la mayor mejora en R@1, aumentando desde un 93.3 % a un 96.3 %. Como resultado, solo se incluirán 3 conexiones de salto entre el codificador y el decodificador.

G4. Cambio del *stem* a "Patchify". El *stem* se refiere a la primera capa de la red, que realiza el procesamiento inicial de la nube. En este caso, el primer procesamiento se lleva a cabo mediante una convolución 3D dispersa con un tamaño de *kernel* de 5 y un *stride* de 1. El término "Patchify" se refiere al acto de dividir los datos de entrada en una secuencia de parches o palabras. Los Transformers Visuales [57] introdujeron este concepto, originalmente inspirado en los Transformers del lenguaje [220], donde cada frase está dividida en palabras. El Swin Transformer [89] utiliza como *stem* una convolución no superpuesta con un tamaño de *kernel* de 4 y un *stride* de 4. En este sentido, estos parámetros se adoptan para el *stem* en G4, pero el rendimiento de la red disminuye del 96.3 % al 92.8 %, por lo que "Patchify" se descarta.

4.4.5.2 Diseño del Bloque Residual

Esta sección describe cada paso de diseño desde el Bloque ResNet hasta el Bloque MinkNeXt propuesto. La hoja de ruta del diseño de este bloque residual se divide en

los siguientes puntos: creación de un cuello de botella en el bloque residual, creación de un cuello de botella invertido en el bloque residual, reemplazo de ReLUs por GeLUs, sustitución de *BatchNorms* (BNs) por *LayerNorms* (LNs) y evaluación de diferentes tamaños de *kernel*.

R1. Creación de un cuello de botella en el bloque residual. Un cuello de botella consiste en reducir la dimensionalidad de la capa oculta para luego expandirla a su tamaño original utilizando convoluciones 3D 1x1. Esta modificación condujo a peores resultados en el rendimiento de la arquitectura propuesta.

R2. Creación de un cuello de botella invertido en el bloque residual. Cada bloque Transformer se caracteriza por un cuello de botella invertido, que consiste en expandir la dimensionalidad del mapa de características de la capa oculta para luego reducirla a su tamaño original mediante convoluciones 3D 1x1. En este caso, se emplean convoluciones dispersas 3D con tamaño de *kernel* 3 y *stride* 1 para crear el cuello de botella invertido con una dimensión oculta cuatro veces más ancha que la dimensión de entrada. La Figura 4.4 muestra que este bloque de cuello de botella invertido produce mejores resultados en comparación con el bloque ResNet anterior cuando se analiza conjuntamente con la siguiente modificación (R3).

R3. Reemplazo de ReLUs por GeLUs. La ReLU [233] es la función de activación más común de los últimos años debido a su simplicidad y eficiencia. Sin embargo, los avanzados recientes en Transformers, como BERT de Google [234] o GPT-4 de OpenAI [235], emplean GeLUs [230], que es una variante más suave que las ReLUs. Siguiendo la misma filosofía, las ReLUs se reemplazan por GeLUs tanto en el bloque ResNet como en el bloque de cuello de botella invertido, pasos R3.1 y R3.2 de la Figura 4.4, respectivamente. En ambos casos, el rendimiento de la arquitectura mejora, pero se obtienen mejores resultados con el bloque de cuello de botella invertido, alcanzando un R@1 del 97.0 %. En consecuencia, se utilizará un cuello de botella invertido con GeLUs como bloque residual.

R4. Sustitución de *BatchNorms* (BNs) por *LayerNorms* (LNs). Las *BatchNorms* (BN) [232] juegan un papel fundamental en las redes convolucionales al mejorar la convergencia y mitigar el sobreajuste. Sin embargo, las BNs pueden introducir complejidades que pueden afectar negativamente el rendimiento del modelo. Recientemente, las *LayerNorms* (LNs) [231] se han implementado con éxito en Transformers. Por tanto, las BNs se reemplazan por LNs en el bloque residual propuesto, obteniendo una mejora del rendimiento del modelo hasta el 97.4 %. Como resultado, se emplearán las LNs en lugar de las BNs en el bloque residual.

R5. Evaluación de diferentes tamaños de *kernel*. Los Transformers Visuales se caracterizan por emplear grandes tamaños de *kernel* con una dimensión mínima de 7. Sin embargo, como se muestra en la Figura 4.4 (R5), el uso de tamaños de *kernel* más pequeños es beneficioso en la presente tarea de reconocimiento de lugares, tanto en las capas de entrada, como en las ocultas y finales del bloque

ID	Modificaciones de diseño
G1.1	Cardinalidad: (2, 3, 4, 6, 2, 2, 2, 2) → (2, 2, 2, 2, 2, 2, 2, 2)
G1.2	Cardinalidad: (2, 2, 2, 2, 2, 2, 2, 2) → (1, 1, 1, 1, 1, 1, 1, 1)
G2.1	Canales del decodificador: (128, 128, 96, 96) → (128, 128, 128, 128)
G2.2	Canales del decodificador: (128, 128, 96, 96) → (192, 192, 128, 128)
G3.1	4 conexiones de salto → 2 conexiones de salto
G3.2	4 conexiones de salto → 3 conexiones de salto
G4	*Stem* (k=5, s=1 → k=4, s=4)
R1	Bloque ResNet → Cuello de botella
R2	Bloque ResNet → Cuello de botella invertido
R3.1	Bloque ResNet con ReLUs → Bloque ResNet con GeLUs
R3.2	Cuello de botella invertido con ReLUs → Cuello de botella invertido con GeLUs
R4	Cuello de botella invertido con BNs → Cuello de botella invertido con LNs
R5.1	1ª convolución del cuello de botella invertido (k=3 → k=5)
R5.2	1ª convolución del cuello de botella invertido (k=3 → k=7)
R5.3	1ª convolución del cuello de botella invertido (k=3 → k=1)

Tabla 4.3: Esta tabla resume todas las modificaciones propuestas en el proceso de diseño de la arquitectura, desde MinkUNet hasta MinkUNeXt.

residual. En este sentido, se encuentra la mejor configuración de parámetros con un tamaño de *kernel* de 1 en la primera convolución y tamaños de *kernel* de 3 en las convoluciones ocultas y finales. Esto conduce a la arquitectura final del modelo y del bloque residual, que hemos denominado MinkUNeXt y bloque MinkNeXt, respectivamente.

4.4.6 Comparación con el estado del arte

Como se definió en la Subsección 4.4.3, se han seguido los dos protocolos de entrenamiento y evaluación previamente establecidos en [4] para el reconocimiento de lugares a partir de los conjuntos de datos del Oxford RobotCar e In-house. El protocolo base consiste en entrenar el modelo únicamente con los datos de entrenamiento de Oxford y evaluar con los datos de test de Oxford e In-house (U.S., R.A. y B.D.). Por el contrario, el protocolo refinado consiste en entrenar con los datos de entrenamiento de Oxford e In-house (U.S., R.A.) y evaluar con los datos de test de Oxford e In-house (U.S., R.A. y B.D.). Estos protocolos son ampliamente utilizados en la literatura, por lo que la comparación se realiza en los mismos términos y condiciones. Además, los resultados comparativos mostrados aquí se han obtenido directamente de los trabajos que se referencian.

Las Tablas 4.4 y 4.5 presentan una visión general de las diferentes técnicas propuestas en el estado del arte en comparación con MinkUNeXt bajo los mismos protocolos de entrenamiento y evaluación (base y refinado), en términos de *recall at 1 (R@1)* y *recall at 1 % (R@1 %)*. Cada columna presenta los resultados obtenidos en cada uno de los entornos, mientras que las dos últimas columnas presentan los resultados promedio. Además, la Tabla 4.6 compara la eficiencia de las arquitecturas más relevantes

en términos de número de parámetros y tiempo de inferencia.

4.4.6.1 Resultados con el Protocolo Base

La Tabla 4.4 presenta los resultados de varios métodos en términos de *recall at 1 (R@1)* y *recall at 1 % (R@1 %)*. Se puede observar que PointNetVLAD estableció el punto de partida para el reconocimiento de lugares a partir de nubes de puntos en el Oxford Robotcar y el conjunto de datos In-house. Posteriormente, PCAN supera ligeramente a PointNetVLAD en la mayoría de los entornos. Además, BPT destaca con resultados realmente competitivos, especialmente en Oxford y U.S. Sin embargo, RPR-Net sobrepasa a BPT en U.S, R.A y B.D., mostrando mejores capacidades de generalización. Algunos trabajos, como DAGC y Retriever, no proporcionan resultados de R@1 para todos los conjuntos de datos. Sin embargo, presentan resultados de R@1 % que muestran un rendimiento mejor que PCAN, pero peor que BPT. Además, LPD-Net, HiTPR, EPC-Net y E^{2}PN-GeM muestran resultados similares entre sí en múltiples escenarios. SOE-Net, sólo proporciona resultados de R@1 % pero son realmente prometedores, ya que están cerca de MinkLoc3D, la primera arquitectura que logra superar el 90 % en R@1 para el conjunto de datos de Oxford. Además, HiBi-Net, PPT-Net y SVT-Net muestran un rendimiento ligeramente superior, específicamente para el conjunto de datos In-house. TransLoc3D da un paso adelante con el mejor resultado hasta la fecha en Oxford y un rendimiento sólido en los otros escenarios. Además, la versión mejorada de MinkLoc3D (MinkLoc3Dv2) supera al resto de las arquitecturas. Además, KPPR también muestra un rendimiento notable, pero sólo presenta resultados de *recall at 1 %* en el caso de U.S., R.A., B.D.

Finalmente, la arquitectura propuesta, MinkUNeXt, demuestra un rendimiento superior en términos de R@1 y R@1 % en Oxford, superando a todos los métodos existentes con un 97.5 % en R@1 y un 99.3 % en R@1 %. Sin embargo, el rendimiento disminuye ligeramente cuando el modelo se evalúa en U.S., R.A. y B.D. Cabe destacar que el conjunto de datos de Oxford y los tres conjuntos de datos in-house se obtuvieron utilizando sensores diferentes. El conjunto de datos de Oxford se captura con varios SICK LMS-151 2D y el conjunto de datos In-house con un Velodyne de 64 canales. Además, los submapas dentro del conjunto de datos de Oxford contienen escenas que son completamente urbanas, caracterizadas por entornos urbanos y estructurados. En contraste, los escenarios presentes en el conjunto de datos in-house son considerablemente más abiertos, con una disposición más dispersa. Esta diferencia en la naturaleza de las escenas capturadas puede influir significativamente en los resultados y el rendimiento del modelo en cada conjunto de datos.

4.4.6.2 Resultados con el Protocolo Refinado

En cuanto al rendimiento de los modelos al entrenar con el protocolo refinado (Tabla 4.5), PointNetVLAD también introdujo el punto de referencia inicial, logrando un buen rendimiento en U.S. R.A. y B.D. a pesar de la simplicidad de su arquitectura. PCAN y DAGC presentaron resultados similares a PointNetVLAD para el conjunto de datos In-house, pero especialmente mejores en Oxford. En contraste, LPD-Net y SOE-

Net muestran un rendimiento sustancialmente mejor en todas las métricas y conjuntos de datos. MinkLoc3D también logra superar el 90 % en R@1 en Oxford y generalmente funciona bien en todas las métricas y conjuntos. PPT-Net no proporciona valores para el *recall at 1* (R@1), pero muestra un rendimiento prometedor en el *recall at 1 %* (R@1 %). Además, SVT-Net destaca especialmente en U.S., R.A. y B.D. Por otro lado, TransLoc3D logra buenos resultados en todas las métricas, siendo uno de los mejores métodos en general. Al igual que en el anterior protocolo, MinkLoc3Dv2 obtuvo los mejores resultados en el estado del arte hasta el momento, mostrando mejoras sobre MinkLoc3D.

Finalmente, el modelo MinkUNeXt propuesto muestra mejoras considerables en el *recall at 1* (R@1) y el *recall at 1 %* (R@1 %) en todos los escenarios, obteniendo los mejores resultados del estado del arte hasta la fecha. La métrica de *recall at 1* en el conjunto de datos de Oxford es del 97.7 % y supera al segundo clasificado, MinkLoc3Dv2, en un 0.8 %. En los escenarios B.D. y R.A., supera a MinkLoc3Dv2 en 0.1 % y 1.1 %, respectivamente. Sin embargo, se obtienen resultados ligeramente inferiores (0.3 %) con esta métrica en el conjunto de datos de U.S. En cuanto a los resultados en términos de R@1 % para el protocolo refinado, el margen de mejora es muy estrecho. Sin embargo, los resultados en Oxford se mejoran en un 0.2 % para alcanzar el 99.3 %, en R.A. en 0.5 % para alcanzar el 99.9 % y en B.D. en 0.1 % para alcanzar el 97.7 %. Además, aunque el modelo anteriormente presentó resultados ligeramente inferiores en U.S. en términos de R@1, el rendimiento de la red en la métrica R@1 % es igual al mejor resultado del estado del arte con un valor de 99.9 %. La media R@1 y R@1 % en los 4 conjuntos de datos mejora en un 0.4 % y un 0.2 %, respectivamente. Para concluir, entrenar MinkUNeXt con el protocolo refinado supera las dificultades de generalización presentadas al entrenar con el protocolo base, ya que el modelo se adapta a diferentes sensores y entornos.

4.4.6.3 Resultados en términos de eficiencia

La Tabla 4.6 presenta una comparación entre los principales métodos de reconocimiento de lugares basados en nubes de puntos, considerando el número de parámetros del modelo y el tiempo de inferencia por muestra. PointNetVLAD y LPD-Net, aunque tienen un número relativamente altos de parámetros (alrededor de 20M), presentan tiempos de inferencia bajos (3.92 ms y 5.48 ms, respectivamente) debido a la simplicidad de sus arquitecturas basadas en MLPs. MinkLoc3D y MinkLoc3Dv2 son modelos ligeros con 1.1M y 2.7M de parámetros, respectivamente. Sin embargo, la segunda versión tiene un tiempo de inferencia más alto (9.63 ms) debido a la incorporación de bloques adicionales en su arquitectura. TransLoc3D, con 10.97M de parámetros, utiliza mecanismos de atención que aumentan su tiempo de inferencia a 7.27 ms. En contraste, MinkUNeXt es el modelo con mayor número de parámetros (43.5M), pero mantiene un tiempo de inferencia competitivo (10.75 ms), dando lugar a la mejor relación entre capacidad y eficiencia, lo que permite su uso en aplicaciones en tiempo real. Este resultado se debe a la eficiencia de las convoluciones dispersas 3D utilizadas en su arquitectura junto con el uso de Convoluciones 1x1x1 para aumentar la dimensionalidad de las características en los bloques residuales.

	Oxford		U.S.		R.A.		B.D.		Media	
Método	R@1	R@1 %	R@1	R@1 %	R@1	R@1 %	R@1	R@1 %	R@1	AR@1 %
PointNetVLAD [4]	62.8	80.3	63.2	72.6	56.1	60.3	57.2	65.3	59.8	69.6
PCAN [236]	69.1	83.8	62.4	79.1	56.9	71.2	58.1	66.8	61.6	75.2
DAGC [219]	-	87.5	-	83.5	-	75.7	-	71.2	-	79.5
BPT [237]	85.7	93.3	80.5	89.3	77.4	86.6	74.1	78.5	79.4	86.9
Retriever [222]	-	91.9	-	91.9	-	87.4	-	85.5	-	89.2
RPR-Net [238]	81.0	92.2	83.2	94.5	83.3	91.3	80.4	86.4	82.0	91.1
LPD-Net [91]	86.3	94.9	87.0	96.0	83.1	90.5	82.5	89.1	84.7	92.6
HiTPR [224]	87.8	94.6	86.0	94.0	81.3	89.1	81.8	88.3	84.2	91.5
EPC-Net [239]	86.2	94.7	-	96.5	-	88.6	-	84.9	-	91.2
E^2PN-GeM [240]	84.8	93.2	88.1	95.3	83.7	90.5	83.3	87.7	85.0	91.7
SOE-Net [221]	-	96.4	-	93.2	-	91.5	-	88.5	-	92.4
MinkLoc3D [84]	93.0	97.9	86.7	95.0	80.4	91.2	81.5	88.5	85.4	93.2
HiBi-Net [241]	87.5	95.1	87.8	-	85.8	-	83.0	-	86.0	-
NDT-Transformer [85]	93.8	97.7	-	-	-	-	-	-	-	-
PPT-Net [86]	93.5	98.1	90.1	97.5	84.1	93.3	84.6	90.0	88.1	94.7
SVT-Net [223]	93.7	97.8	90.1	96.5	84.3	92.7	85.5	90.7	88.4	94.4
TransLoc3D [87]	95.0	98.5	-	94.9	-	91.5	-	88.4	-	93.3
MinkLoc3Dv2 [227]	96.3	98.9	**90.9**	96.7	**86.5**	93.8	**86.3**	91.2	**90.0**	95.1
KPPR [242]	91.5	97.1	-	**98.0**	-	**95.1**	-	**92.1**	-	**95.6**
MinkUNeXt (nuestro)	**97.5**	**99.3**	88.9	96.5	85.0	91.3	85.2	90.1	89.1	94.3

Tabla 4.4: Resultados de evaluación en términos de *recall at 1* (R@1) y *recall at 1 %* (R@1 %) de los diferentes métodos de reconocimiento de lugares entrenados usando el protocolo base.

4.5 Resultados cualitativos de la tarea de reconocimiento de lugares

En esta sección se presentan ejemplos visuales de los resultados obtenidos por MinkUNeXt en los conjuntos de datos de Oxford RobotCar (Figuras 4.5, 4.6 y 4.7) e In-house (Figuras 4.8, 4.9 y 4.10). Estos ejemplos se han seleccionado de las secuencias de test de cada conjunto de datos, y se muestran para ilustrar la capacidad del modelo para reconocer lugares en diferentes entornos y condiciones. En cada figura, se muestra un ejemplo de las diferentes secuencias del conjunto de datos, donde se observa la nube de puntos de test capturada por el sensor LiDAR y la predicción de la nube más cercana en el espacio del descriptor de la base de datos. Además, se comprueba si coincide con la nube más cercana en el espacio métrico de la posición. Las posiciones del mapa se representan con puntos azules, la posición actual (correspondiente a la nube de test) con una cruz roja, la posición predicha con un círculo amarillo y la posición real (la mejor predicción posible) con un anillo verde. Además, se representan con rectángulos rojos las zonas donde se lleva a cabo la evaluación del modelo. En el caso del entorno *Bussiness District* (B.D.), se considera completamente para la evaluación, es por ello que en este no aparece ningún rectángulo rojo.

Los resultados cualitativos obtenidos en el conjunto de datos de Oxford RobotCar (Figuras 4.5, 4.6 y 4.7) muestran ejemplos de las secuencias 2014-11-14-16-34-33, 2015-02-17-14-42-12 y 2015-11-13-10-28-08, respectivamente. Como se puede apreciar, en este conjunto de datos transcurre hasta un año entre la primera y la última secuencia, lo que provoca cambios significativos en el entorno. En este sentido, el modelo MinkUNeXt es capaz de adaptarse a estos cambios y realizar predicciones correctas. Por ejemplo, en la Figura 4.5 se observan cambios en la vegetación, y aun

	Oxford		U.S.		R.A.		B.D.		Media	
Método	R@1	R@1 %	R@1	R@1 %	R@1	R@1 %	R@1	R@1 %	R@1	R@1 %
PointNetVLAD [4]	63.3	80.1	86.1	94.5	82.7	93.1	80.1	86.5	78.0	88.6
PCAN [236]	70.7	86.4	83.7	94.1	82.5	92.5	80.3	87.0	79.3	90.0
DAGC [219]	71.5	87.8	86.3	94.3	82.8	93.4	81.3	88.5	80.5	91.0
LPD-Net [91]	86.6	94.9	94.4	98.9	90.8	96.4	90.8	94.4	90.7	96.2
SOE-Net [221]	89.3	96.4	91.8	97.7	90.2	95.9	89.0	92.6	90.1	95.7
MinkLoc3D [84]	94.8	98.5	97.2	99.7	96.7	99.3	94.0	96.7	95.7	98.6
PPT-Net [86]	-	98.4	-	99.7	-	99.5	-	95.3	-	98.2
SVT-Net [223]	94.7	98.4	97.0	**99.9**	95.2	99.5	94.4	97.2	95.3	98.8
TransLoc3D [87]	95.0	98.5	97.5	99.8	97.3	99.7	94.8	97.4	96.2	98.9
MinkLoc3Dv2 [227]	96.9	99.1	**99.0**	99.7	98.3	99.4	97.6	**99.1**	97.9	99.3
MinkUNeXt (nuestro)	**97.7**	**99.3**	98.7	**99.9**	**99.4**	**99.9**	**97.7**	99.0	**98.3**	**99.5**

Tabla 4.5: Resultados de evaluación en términos de *recall at 1* (R@1) y *recall at 1 %* (R@1 %) de los diferentes métodos de reconocimiento de lugares entrenados usando el protocolo refinado.

Método	Parámetros (M)	Tiempo de inferencia (ms)
PointNetVLAD [4]	19.78	3.92
LPD-Net [91]	19.81	5.48
MinkLoc3D [84]	1.1	3.29
TransLoc3D [87]	10.97	7.27
MinkLoc3Dv2 [227]	2.7	9.63
MinkUNeXt (nuestro)	43.5	10.75

Tabla 4.6: Comparación del número de parámetros y el tiempo de inferencia de los diferentes métodos de reconocimiento de lugares.

así, el modelo logra identificar correctamente el lugar. En la Figura 4.6 el modelo no logra identificar correctamente el lugar, debido a que se trata de nubes de puntos muy similares capturadas en una calle con edificios a ambos lados y con una estructura muy parecida. Por otro lado, en la Figura 4.7 MinkUNeXt realiza una predicción correcta a pesar de que el vehículo circula en sentido contrario respecto a la base de datos.

Los resultados cualitativos obtenidos en el conjunto de datos In-house (Figuras 4.8, 4.9 y 4.10) muestran ejemplos de las secuencias 2, 4 y 2 de U.S., R.A. y B.D., respectivamente. En este caso, el modelo MinkUNeXt también es capaz de adaptarse a los cambios en la escala y la densidad de la nube de puntos. Por ejemplo, en la Figura 4.8 el modelo no logra realizar una predicción correcta debido a que la nube de puntos capturada es bastante similar a la nube de puntos obtenida de la base de datos, dando lugar a confusión. Sin embargo, en la Figura 4.9 MinkUNeXt logra realizar una predicción correcta a pesar de que hay elementos dinámicos en la escena, es decir, vehículos que aparecen en la nube de puntos de test, pero no en las nubes que conforman la base de datos. Por último, en la Figura 4.10 el modelo también realiza una predicción correcta a pesar del cambio de escala.

Por último, para ver más ejemplos se puede visitar la página del proyecto https://juanjo-cabrera.github.io/projects-MinkUNeXt/.

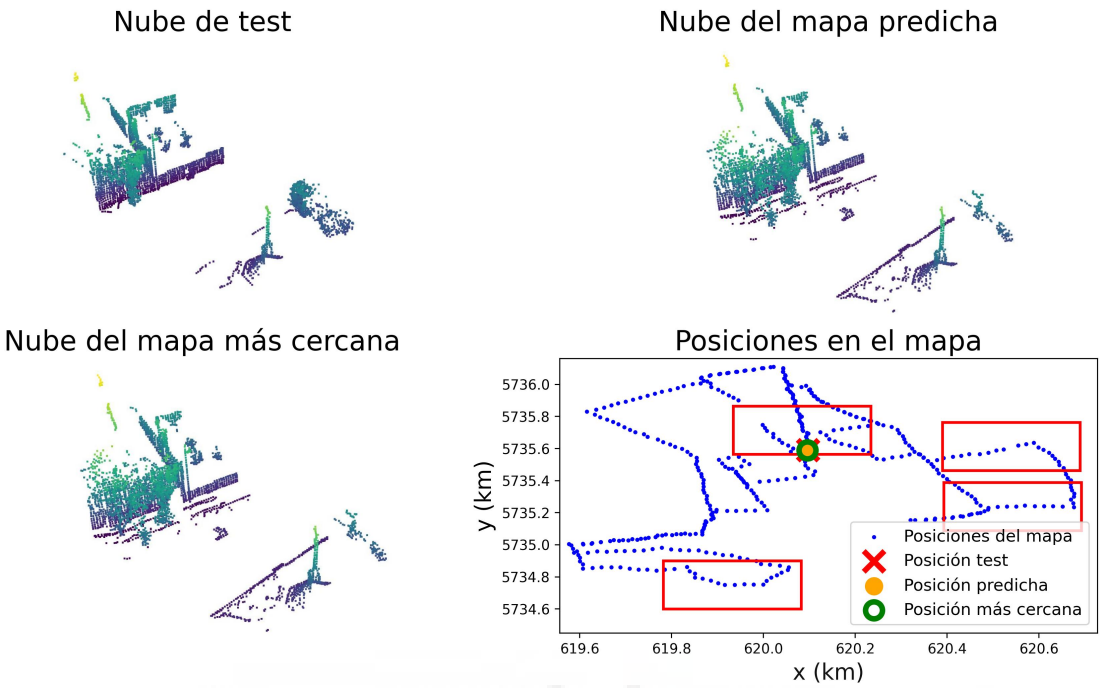


Figura 4.5: Ejemplo obtenido de la secuencia 2014-11-14-16-34-33 del conjunto Oxford Robotcar con predicción exitosa pese a cambios en la vegetación.

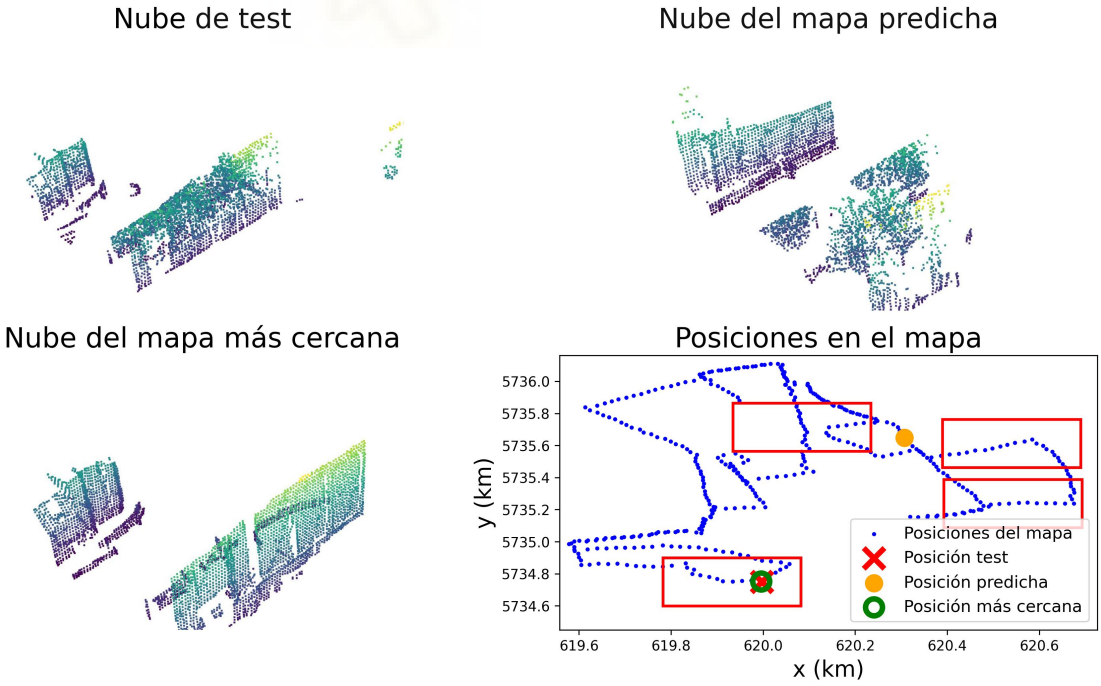


Figura 4.6: Ejemplo obtenido de la secuencia 2015-02-17-14-42-12 del conjunto Oxford Robotcar con predicción totalmente errónea.

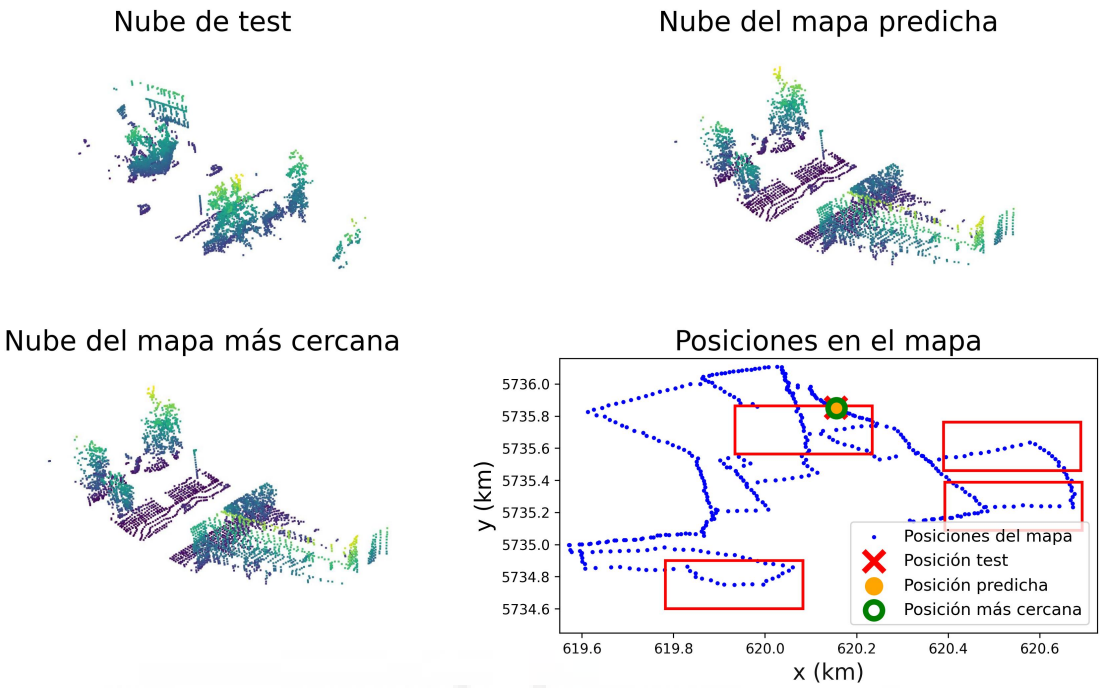


Figura 4.7: Ejemplo obtenido de la secuencia 2015-11-13-10-28-08 del conjunto Oxford Robotcar con predicción exitosa pese al cambio de orientación.

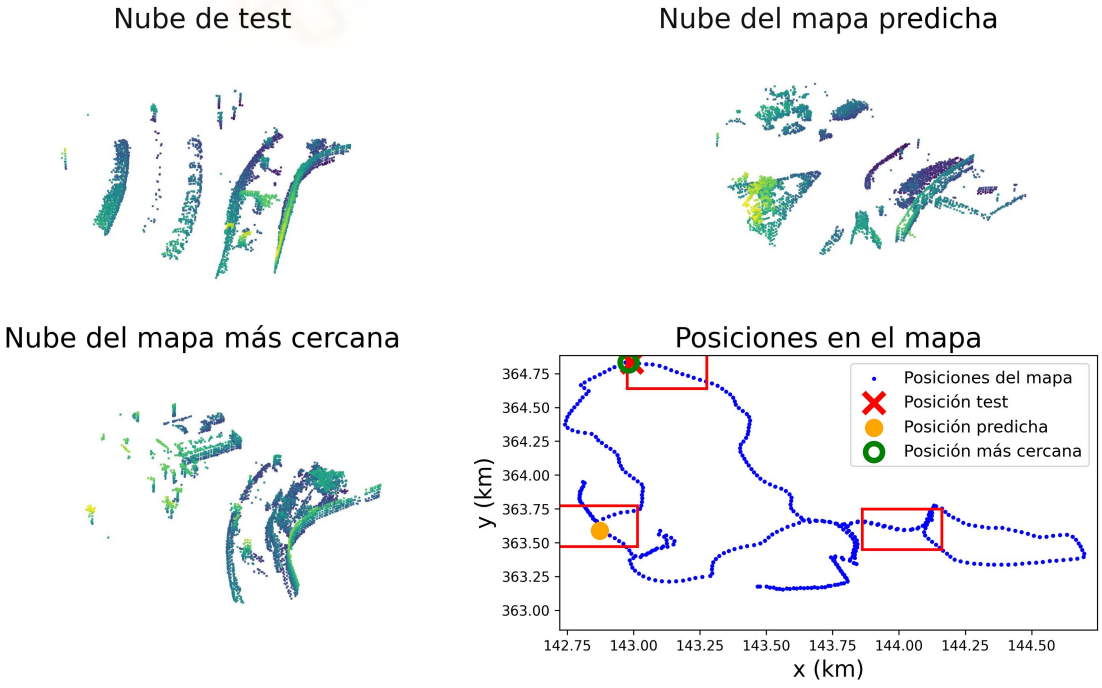


Figura 4.8: Ejemplo obtenido de la secuencia 2 del conjunto In-house (U.S.) con predicción totalmente errónea.

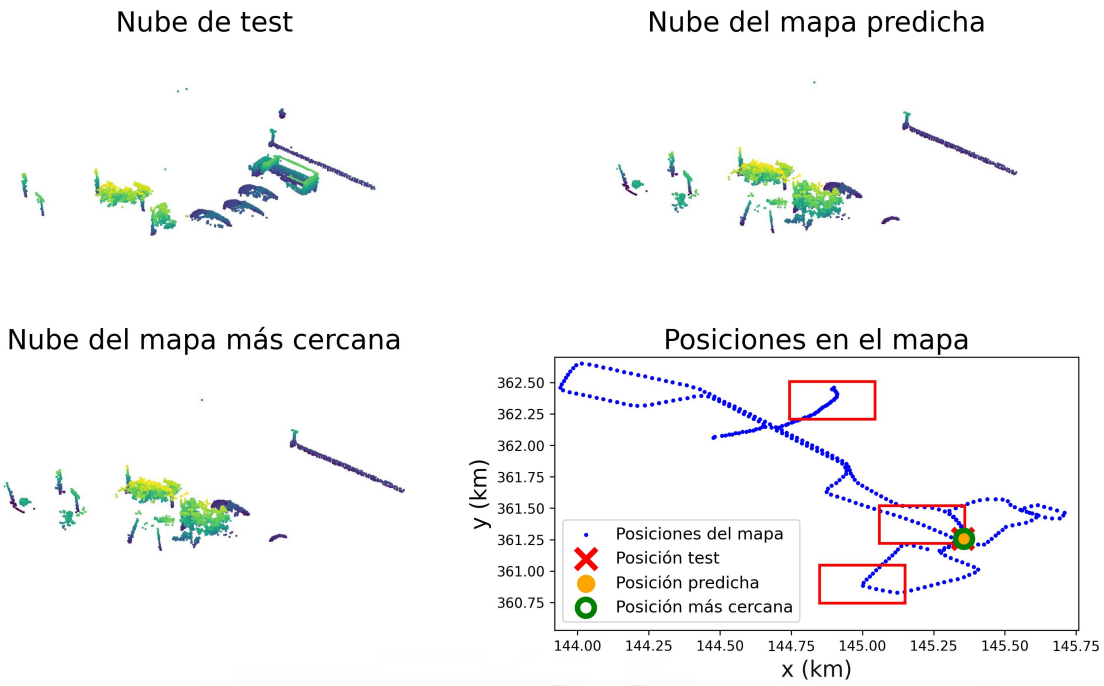


Figura 4.9: Ejemplo obtenido de la secuencia 4 del conjunto In-house (R.A.) con predicción exitosa pese a los elementos dinámicos.

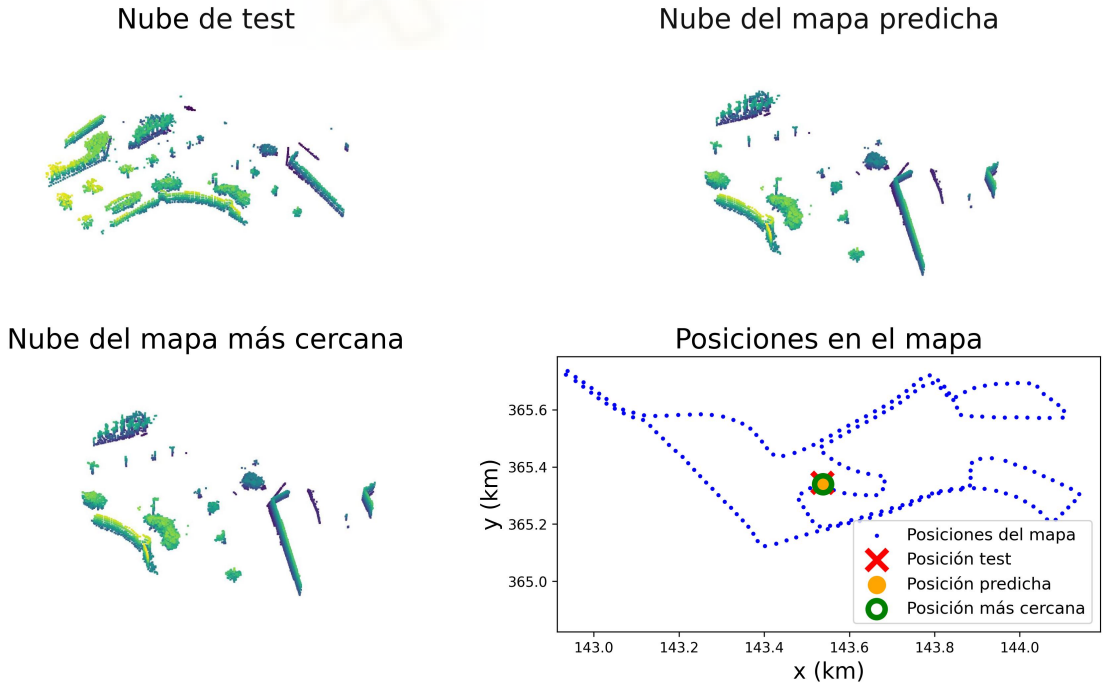


Figura 4.10: Ejemplo obtenido de la secuencia 2 del conjunto In-house (B.D.) con predicción exitosa pese al cambio de escala.

4.6 Conclusiones

En este capítulo se ha presentado MinkUNeXt, una arquitectura basada en MinkUNet [11] ampliamente modificada y mejorada para el reconocimiento de lugares basado en nubes de puntos. Es una arquitectura codificador-decodificador basada en el Bloque MinkNeXt 3D propuesto en la presente tesis: un bloque residual compuesto por convoluciones 3D dispersas que sigue la filosofía de ConvNeXt [10]. La extracción de características se realiza mediante un codificador-decodificador U-Net. La agregación de estas características en un solo descriptor se lleva a cabo mediante un *Generalized Mean Pooling* (GeM) [225]. La arquitectura diseñada demuestra que es posible superar el estado del arte actual confiando únicamente en convoluciones 3D dispersas sin utilizar propuestas más complejas y sofisticadas como Transformers, Capas de Atención o Convoluciones Deformables.

La red propuesta muestra que el uso de una arquitectura U-Net para el reconocimiento de lugares basado en nubes de puntos es beneficioso, ya que es capaz de capturar información detallada y semántica del entorno. La fusión de características a múltiples escalas espaciales mejora la robustez del modelo para el reconocimiento de lugares, permitiéndole adaptarse a variaciones en la geometría y densidad de la nube de puntos, así como a diferentes escenarios.

También cabe destacar que el método propuesto logra superar a un estado del arte ya saturado. En particular, la red logra un R@1 del 97.5 % y un R@1 % del 99.3 % cuando se entrena con el protocolo refinado. Por lo tanto, hay poco margen de mejora y se necesitan escenarios más desafiantes para poder observar mayores diferencias en los resultados obtenidos con distintas técnicas que justifiquen el empleo de métodos más avanzados.

En trabajos futuros se considerará la inclusión de información visual para el reconocimiento de lugares. En este sentido, consideramos que esto resultaría en una representación más rica del entorno en comparación con el uso exclusivo de LiDAR. Sin embargo, la información visual puede verse afectada por cambios en las condiciones de iluminación, estación y clima, lo que plantea un gran desafío.

5

Reconocimiento de lugares basado en pseudo-LiDAR

5.1 Introducción

La robótica móvil ha experimentado avances significativos recientemente, impulsados por la necesidad de desarrollar sistemas autónomos capaces de navegar y operar en entornos variados y complejos. La navegación autónoma de robots móviles implica la integración de sensores y algoritmos para percibir, comprender e interactuar con el entorno. Los avances en tecnología de sensores han sido cruciales en el desarrollo de estos sistemas, permitiendo mayor precisión y fiabilidad.

Uno de los desafíos persistentes en robótica móvil es el reconocimiento de lugares, que implica identificar áreas específicas en el entorno con fines de navegación y localización. Los sistemas actuales de reconocimiento de lugares utilizan una variedad de sensores, como LiDARs [72] y cámaras [243], cada uno con sus propias ventajas y limitaciones. Los sistemas LiDAR son sensores auto-iluminados, por lo tanto intrínsecamente invariantes a los cambios de iluminación en la escena. Los sistemas LiDAR poseen la capacidad de obtener una nube de puntos 3D detallada y precisa de la escena a alta frecuencia. Sin embargo, pueden tener dificultades para obtener información precisa en entornos con superficies reflectantes o transparentes. En comparación con los sistemas LiDAR, las cámaras omnidireccionales son mucho más económicas y, al mismo tiempo, proporcionan gran cantidad de información del entorno que incluye: forma, textura y color. Además, las cámaras omnidireccionales son capaces de capturar una vista 360º del entorno que rodea al robot [244]. Por contra, la apariencia de la imagen capturada por una cámara puede verse significativamente alterada cuando se enfrenta a condiciones cambiantes de luz natural y/o artificial.

Dadas las ventajas e inconvenientes de los sensores disponibles, algunos investigadores están explorando enfoques multi-sensor, como la combinación de cámaras estándar con LiDAR [245] o con cámaras infrarrojas [142]. Tales enfoques pretenden combinar las fortalezas de cada tecnología, mejorando la robustez bajo diversas condiciones de iluminación y potenciando la efectividad general de los sistemas autónomos en aplicaciones reales. Sin embargo, la combinación de varios sensores puede aumentar significativamente el coste de la plataforma robótica, la complejidad computacional y de la tecnología necesaria para la calibración, sincronización y procesamiento de datos en tiempo real. Además del planteamiento de algoritmos más sofisticados para la fusión de datos.

En el contexto del reconocimiento de lugares, el uso de una única cámara omnidireccional como única fuente de información es aconsejable: los sensores del robot son más económicos evitando el uso de sistemas LiDAR. Por el contrario, lograr una descripción invariante a perturbaciones en la escena basada únicamente en datos visuales es desafiante. Para abordar este problema, este capítulo propone el uso de nubes de puntos pseudo-LiDAR [246], que se generan a partir de mapas de profundidad obtenidos por medios de modelos de estimación de la profundidad. El enfoque propuesto aprovecha las ventajas de la información de profundidad para mejorar la robustez frente a cambios en la apariencia visual, manteniendo al mismo tiempo una configuración sensorial económica y ligera.

5.1.1 Contribuciones de este capítulo

Este capítulo presenta un enfoque novedoso para el reconocimiento de lugares utilizando nubes de puntos pseudo-LiDAR generadas a partir de imágenes omnidireccionales. En particular, proponemos un método para abordar el reconocimiento de lugares basado únicamente en vistas panorámicas, que se transforman en mapas de profundidad mediante Distill Any Depth [13], un modelo de estimación de profundidad entrenado utilizando un método de destilación multi-maestro. Posteriormente, se generan nubes de puntos pseudo-LiDAR a partir de las imágenes de profundidad panorámicas. Al embeber estas nubes de puntos con MinkUNeXt, es posible mejorar la capacidad del sistema para reconocer y mapear el entorno con mayor precisión, independientemente de cambios sustanciales en la apariencia. Las principales contribuciones de este capítulo pueden resumirse de la siguiente manera:

- Abordar el reconocimiento visual de lugares en escenarios de interior bajo diferentes condiciones de iluminación mediante nubes de puntos pseudo-LiDAR, que se calculan a partir de vistas panorámicas por medio de Distill Any Depth.
- Mejorar la robustez de los modelos de reconocimiento de lugares a cambios en la apariencia mediante la aplicación de una nueva técnica de aumento de datos llamada *Distilled Depth Variations*.
- Mejorar las nubes de puntos 3D con características visuales, principalmente basadas en el gradiente de la intensidad, que es muy robusto ante cambios de iluminación, diferentes configuraciones de cámara y plataformas de adquisición de datos, y por tanto mejoran la generalización del método propuesto.

- Evaluación exhaustiva del método propuesto en la base de datos COLD [1], que incluye una amplia variedad de condiciones de iluminación y diferentes entornos con diversas plataformas de adquisición de datos. Se presentan resultados comparativos con otros métodos del estado del arte.

5.2 Trabajos relacionados

Esta sección revisa el estado del arte del reconocimiento de lugares con técnicas de aprendizaje profundo. En particular, se analizan varios enfoques que han empleado redes neuronales profundas con sensores LiDAR, cámaras o una combinación de ambos. Además, esta sección incluye algunos modelos recientes que estiman la profundidad basándose únicamente en información visual.

El reconocimiento de lugares ha sido un tema de investigación durante varias décadas, pero no fue hasta la última década cuando algunos autores comenzaron a explorar el uso de técnicas de aprendizaje profundo [247, 248]. Tomando imágenes de partida, muchos investigadores adaptaron CNNs, previamente entrenadas para resolver tareas de clasificación de objetos, para el problema de reconocimiento de lugares, incluyendo VGG16 [5] o ResNet [6], para sus tareas específicas. Sin embargo, NetVLAD [53] fue la primera CNN entrenada específicamente para el reconocimiento visual de lugares (VPR). Posteriormente, los transformers visuales revolucionaron la forma de extraer características de imágenes, dando lugar a ViT [57], Swin-L [89, 249] o DINO [14, 250], entre otros. Actualmente, los enfoques de VPR se centran en diseñar técnicas de entrenamiento eficientes y escalables, por ejemplo, CosPlace [16], EigenPlaces [54] o AnyLoc [58], o desarrollar métodos de agregación de características para obtener descriptores robustos, como MixVPR [56] o SALAD [59]. Además, otros trabajos han propuesto soluciones que explotan las propiedades de la visión omnidireccional para VPR [251, 252]. Los resultados de los métodos mencionados se ven afectados adversamente cuando las escenas se capturan con diferentes condiciones de iluminación o las imágenes sufren de *visual aliasing*.

Por otra parte, los sensores LiDAR se han convertido en una opción popular para abordar el reconocimiento de lugares [253]. En este ámbito, PointNetVLAD [4] fue la primera red neuronal capaz de obtener descriptores globales a partir de nubes de puntos. A partir de ese momento, otros enfoques desarrollaron arquitecturas más complejas, basadas en redes convolucionales 3D, para abordar el mismo problema, como MinkLoc3Dv2 [227] o MinkUNeXt [254], o basadas en transformers [88, 255]. Otros enfoques han explorado la combinación de datos visuales con LiDAR [139]. En particular, [256] y [142] utilizan sensores LiDAR junto con cámaras omnidireccionales. Sin embargo, los LiDAR siguen siendo dispositivos costosos que podrían evitarse para reducir el coste total de la plataforma móvil.

Los avances recientes en modelos fundacionales han permitido un progreso significativo en varias tareas de visión por computador. Estos modelos entrenados a gran escala con conjuntos de datos diversos y extensos, han demostrado notables capacidades para transferir conocimiento a través de múltiples tareas específicas [257], como

la estimación de profundidad, que consiste en predecir la distancia a la cámara para cada píxel de una imagen [258]. Los modelos de estimación de profundidad pueden clasificarse en dos grupos: modelos generativos [259, 260], que son capaces de modelar los detalles con mayor precisión, y modelos discriminativos [261, 262], que son más robustos a cambios en la escena. Depth Anything [263] pertenece al segundo grupo y logró los mejores resultados en el estado del arte. Además, los mismos autores han presentado recientemente Depth Anything V2 (DAv2) [12], que ha superado claramente a su versión anterior. Inspirados en este trabajo, Hu *et al*. [264] y posteriormente Chen *et al*. [265] han propuesto estimadores de profundidad para vídeo, que garantizan la consistencia en la predicción de profundidad a través de fotogramas consecutivos. También, Guo *et al*. [266] han desarrollado y entrenado un modelo con diferentes tipos de imágenes (estándar, ojo de pez y equirectangulares), con el objetivo de aumentar la calidad de los mapas de profundidad independientemente del tipo de cámara.

La destilación de modelos ha contribuido significativamente a los recientes avances en la estimación de profundidad. Este proceso consiste en entrenar modelos más pequeños empleando el modelo más grande como maestro, es decir, utilizando las predicciones de este modelo como etiquetas para entrenar los modelos estudiantes. En este sentido, DAv2 presenta un modelo Gigante y tres versiones destiladas (Base, Pequeño, Grande). Además, He *et al*. [13] han desarrollado recientemente Distill Any Depth, que emplea DAv2 como arquitectura principal que ha sido entrenado con múltiples maestros, GenPercept [267] y DAv2-Large, lo que conduce a un mejor rendimiento que DAv2.

Debido al auge de los modelos de estimación de profundidad en los últimos años, otros autores han buscado integrar mapas de profundidad predichos en sus algoritmos para abordar diferentes tareas específicas, incluyendo detección de objetos [268], realidad aumentada [269] o diagnóstico médico [270]. Dentro del reconocimiento de lugares, Hettiarachchi *et al*. [271] han desarrollado una red profunda que combina imágenes estándar y mapas de profundidad estimados obtenidos con el modelo ZoeDepth [262]. No obstante, el uso de mapas de profundidad generados a partir de imágenes para obtener una representación pseudo-LiDAR aún permanece inexplorado en este campo.

En consecuencia, en este capítulo se presenta un novedoso método de reconocimiento de lugares, que tiene como objetivo aumentar la robustez frente a variaciones de iluminación y otros fenómenos visuales. La solución propuesta emplea imágenes panorámicas como única fuente de información, a partir de las cuales se generan mapas de profundidad utilizando Distill Any Depth [13]. Posteriormente, estos mapas de profundidad se transforman en nubes de puntos 3D pseudo-LiDAR, que se procesan con la arquitectura MinkUNeXt [254] para obtener un descriptor global.

5.3 Reconocimiento de lugares mediante Pseudo-LiDAR a partir de vistas omnidireccionales

Para abordar el reconocimiento visual de lugares, este capítulo emplea imágenes omnidireccionales capturadas en entornos interiores utilizando un sistema catadióptrico

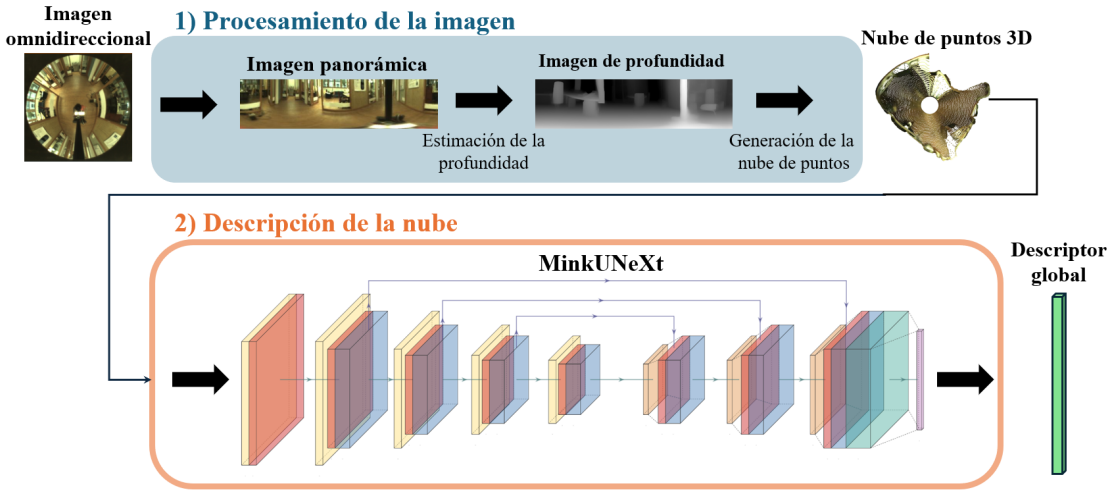


Figura 5.1: Esquema general del método propuesto en este capítulo, que consta de dos pasos: (1) la imagen omnidireccional se transforma en una nube de puntos 3D mediante el mapa estimado de la profundidad, obtenido con Distill Any Depth [13] y (2) la nube de puntos se embebe en un descriptor global con la arquitectura MinkUNeXt.

montado en un robot móvil. Este sistema combina un espejo hiperbólico y una cámara monocular para capturar imágenes con un campo de visión de 360º, permitiendo una obtención completa de datos visuales en una sola imagen. Estas imágenes se convierten posteriormente a formato panorámico con una resolución de 128x512x3 píxeles (RGB), proporcionando un amplio campo de visión horizontal adecuado para tareas de navegación en interiores. Estas imágenes panorámicas se transforman luego en imágenes de profundidad utilizando Distill Any Depth [13]. El método completo se detalla en la Figura 5.1. El resto de la propuesta se resume en las siguientes subsecciones.

5.3.1 Estimación de profundidad

Distill Any Depth [13] se basa en el modelo Depth Anything V2 [12], una arquitectura de estimación de profundidad de última generación diseñada para generar mapas de profundidad precisos a partir de imágenes monoculares. Este modelo utiliza como arquitectura base a DINOv2 [14], un transformer visual reconocido por su capacidad para capturar relaciones entre elementos distantes en la imagen e información semántica. Estas características permiten estimaciones de profundidad altamente detalladas y precisas, incluso en entornos interiores desafiantes. El modelo Distill Any Depth se entrena utilizando un método de destilación multi-maestro, que integra las fortalezas de varios maestros, en este caso GenPercept [267] y DAv2 [12], mientras combina características de profundidad locales y globales para mejorar las predicciones del modelo estudiante.

El modelo de estimación de profundidad procesa las imágenes panorámicas para inferir un mapa de profundidad a nivel de píxel, convirtiendo la imagen panorámica en una representación de profundidad. Cada píxel en el mapa de profundidad codifica la distancia relativa desde la cámara a la escena observada, proporcionando una comprensión espacial detallada del entorno. Estos mapas de profundidad sirven como base

(a) (b)

(c)

Figura 5.2: Ejemplos de (a) una imagen omnidireccional de la base de datos COLD convertida a formato panorámico, (b) un mapa de profundidad obtenido con Distill Any Depth a partir de la imagen panorámica y (c) un mapa de profundidad después del proceso de *inpainting* de LaMa.

para generar las nubes de puntos, que posteriormente se utilizan para el reconocimiento de lugares. La Figura 5.2 ilustra un ejemplo de un mapa de profundidad generado utilizando Distill Any Depth a partir de una imagen panorámica.

5.3.2 Post-procesamiento de profundidad

Dado que el sistema catadióptrico omnidireccional consiste en una cámara monocular emparejada con un espejo hiperbólico, la estructura que soporta el espejo aparece consistentemente como un artefacto en las imágenes de profundidad, produciendo una oclusión. Para aumentar la robustez de las imágenes de profundidad ante rotaciones, se emplean técnicas de *inpainting* como LaMa [17].

LaMa es un modelo de *inpainting* de última generación que utiliza convoluciones de Fourier para reconstruir áreas en imágenes, rellenando eficazmente los huecos con datos realistas y contextualmente apropiados. En el enfoque actual, LaMa se utiliza para reconstruir los valores de profundidad en el área ocluida por la estructura que soporta el espejo hiperbólico, mejorando así la integridad y precisión general de los datos de profundidad. La aplicación de LaMa al mapa de profundidad minimiza el impacto de los artefactos visuales y asegura que las nubes de puntos generadas a partir de las imágenes sean visualmente coherentes y robustas.

Este paso de post-procesamiento es crítico para mantener la fiabilidad del sistema de reconocimiento de lugares, especialmente en entornos de interior dinámicos donde la iluminación y la posición de los objetos pueden cambiar frecuentemente. Los mapas de profundidad rectificados con *inpainting* proporcionan una base más precisa para generar nubes de puntos 3D, que posteriormente se utilizarán para la tarea de reconocimiento de lugares. La Figura 5.2 (b) y (c) muestra un ejemplo de un mapa de profundidad antes y después de realizar la operación de *inpainting* con LaMa, respectivamente.

5.3.3 Estimación de nubes de puntos

La información de profundidad d obtenida del modelo Distill Any Depth se representa inicialmente como un valor adimensional en el rango de 0 a 255. Como resultado, cada píxel está asociado a una distancia de profundidad estimada. Estos valores se transforman luego en mediciones de profundidad (d_m) en metros utilizando la siguiente ecuación:

$$d_m = d_{min} + d \cdot d_s \tag{5.1}$$

donde (d_{min}) y (d_s) son la distancia mínima y el factor de escala de profundidad escogidos de manera arbitraria, respectivamente. Para obtener la nube de puntos, primero, cada píxel en la imagen panorámica se mapea a coordenadas cilíndricas, donde la distancia radial de cada píxel es d_m y el ángulo azimutal θ y la altura z se calculan con la ecuación 5.2:

$$\theta = \frac{u}{w} \cdot 2\pi, \quad z = \left(v - \frac{h}{2}\right) \cdot h_s, \tag{5.2}$$

donde u es la posición horizontal del píxel y w es el ancho de la imagen, v es la posición vertical del píxel, h es la altura de la imagen y h_s es el factor de escala vertical, elegido también de manera arbitraria.

A continuación, las coordenadas cilíndricas se convierten a coordenadas cartesianas $(x,\ y,\ z)$ utilizando la siguiente ecuación:

$$y = d_m \cdot \sin(\theta), \quad x = d_m \cdot \cos(\theta), \quad z = z \tag{5.3}$$

La nube de puntos obtenida de cada imagen omnidireccional representa una estructura espacial tridimensional del entorno, proporcionando información crítica de distancia que mejora la capacidad del robot para reconocer lugares con precisión.

La Tabla 5.2 contiene los valores de los parámetros utilizados para generar las nubes de puntos. Además, la Figura 5.3 incluye tres imágenes omnidireccionales y las nubes de puntos estimadas para cada una de ellas. Las tres imágenes son capturadas bajo diferentes condiciones de iluminación desde, aproximadamente, la misma posición y orientación, y, como resultado, las nubes de puntos estimadas deberían ser iguales. Sin embargo, las diferentes condiciones de iluminación alteran la estimación de profundidad. En el enfoque actual, buscamos mejorar la resiliencia del modelo de reconocimiento de lugares frente a estos efectos entrenándolo con una técnica de aumento de datos específicamente diseñada para este problema, la cual se detalla en la Sección 5.3.5.

5.3.4 Extracción de características y descripción de nubes de puntos

El reconocimiento de lugares a partir de nubes de puntos puede abordarse como una tarea de descripción, donde el objetivo es extraer las características más descriptivas de una escena y agregarlas en un único vector descriptor que mejor represente la información de la escena.

Para este propósito, MinkUNeXt [254] se entrena tanto para la extracción como para la agregación de características. La extracción de características se realiza

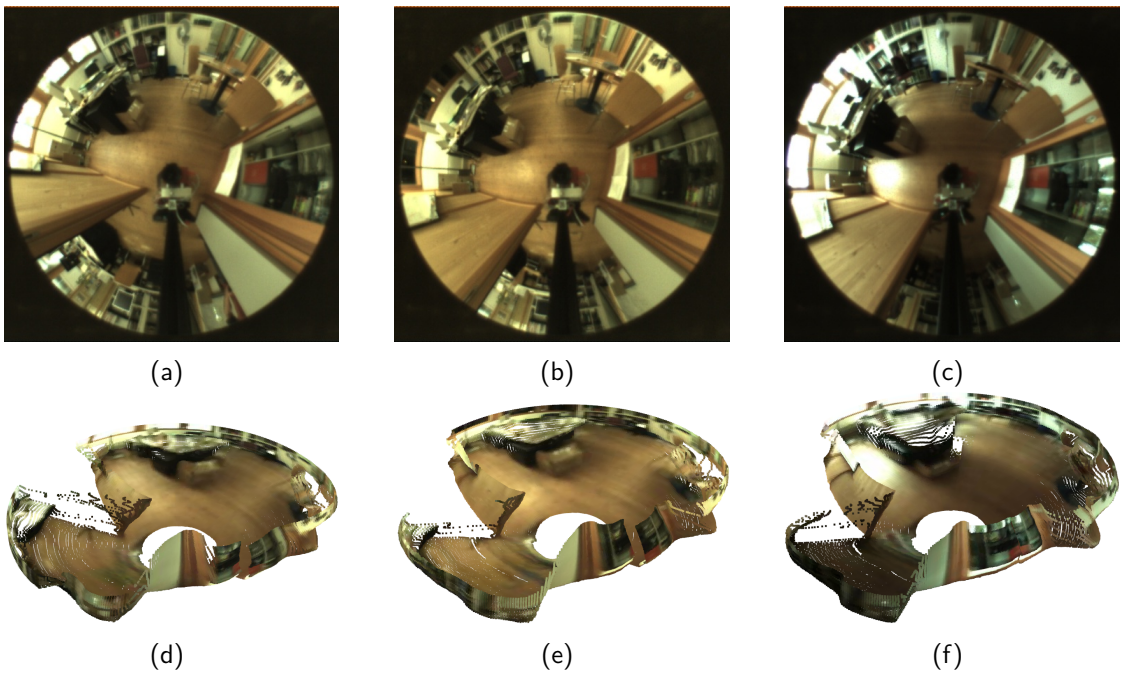


Figura 5.3: Imágenes omnidireccionales (a, b, c) capturadas en condiciones nubladas, nocturnas y soleadas, respectivamente, y sus nubes de puntos estimadas (d, e, f), obtenidas a partir de las imágenes de profundidad.

mediante una arquitectura codificador-decodificador en forma de U, mientras que la agregación de estas características en un único descriptor es gestionada por una capa *Generalized Mean Pooling* (GeM) [225]. MinkUNeXt, diseñada específicamente para el reconocimiento de lugares en entornos de exterior, emplea convoluciones 3D dispersas [11], lo que lo hace particularmente adecuado para procesar nubes de puntos donde la información de entrada es dispersa.

La entrada al modelo MinkUNeXt es una nube de puntos representada como un conjunto desordenado de coordenadas 3D $P = \{(x_i, y_i, z_i)\}$, tal y como se detalló en la Sección 4.3.1. Esta nube de puntos se cuantifica en un tensor disperso, que extiende el concepto de una matriz dispersa a dimensiones superiores, con elementos no nulos representados por un conjunto de coordenadas C y sus características asociadas F. Estas características pueden derivarse de la imagen de entrada (RGB, escala de grises, tono, etc.), de las coordenadas de la nube de puntos (coordenadas xyz, normales, etc.), o inicializarse a 'unos' para permitir que el modelo aprenda las características más adecuadas dada la nube de puntos de entrada.

El enfoque propuesto consiste en alimentar el modelo MinkUNeXt con los gradientes de la imagen como características de la nube. El gradiente se calcula utilizando un operador Sobel aplicado a la imagen de intensidad con un *kernel* 3x3. La imagen gradiente resultante resalta áreas de alta frecuencia espacial, a menudo asociadas con bordes y texturas en la imagen.

El gradiente se descompone luego en magnitud y dirección, donde la magnitud representa la intensidad del gradiente en cada píxel, y la dirección indica la orientación

del gradiente. La dirección se proyecta al círculo unidad utilizando las funciones seno y coseno, permitiendo una representación compacta y diferenciable de la orientación del gradiente.

La magnitud y dirección del gradiente se concatenan posteriormente para formar un vector de características para cada punto de la nube. Este vector de características captura tanto la intensidad como la orientación del gradiente de cada píxel, proporcionando datos valiosos sobre la estructura local y la textura de la imagen. Al incorporar esta información en las características de la nube de puntos, se mejora la capacidad del modelo para reconocer lugares basándose tanto en características geométricas como visuales. Otras características visuales también pueden asociarse a cada punto de la nube, como el tono, la saturación, la intensidad, etc. Se ha realizado un estudio exhaustivo y los resultados se detallan en la Sección 5.4.4.3.

5.3.5 Aumento de datos

En este trabajo de investigación, se presenta una novedosa técnica de aumento de datos llamada *Distilled Depth Variations*, diseñada específicamente para mejorar la robustez de los modelos 3D dispersos al procesar nubes de puntos pseudo-LiDAR. Dado que la evidencia experimental muestra que los estimadores de profundidad son sensibles a factores ambientales como las condiciones de iluminación, lo que conduce a predicciones inconsistentes. Para mitigar este efecto, nuestra estrategia principal de aumento simula variaciones realistas de estimación de profundidad, mejorando así la capacidad de generalización del modelo de reconocimiento de lugares en diferentes condiciones de iluminación. Para una evaluación integral, este enfoque de aumento de datos se compara con técnicas convencionales comúnmente utilizadas en el procesamiento de nubes de puntos, que incluyen:

- **Eliminación de Puntos**: esta técnica elimina aleatoriamente un subconjunto de puntos (por ejemplo, eliminando un 20 % de los puntos de la nube) para simular oclusiones parciales o fallos del sensor. Incentiva el aprendizaje de características contextuales y estructurales en lugar de sobreajustarse a distribuciones específicas de puntos.
- **Rotación**: la nube se rota alrededor del eje vertical con un ángulo aleatorio (en el rango [-180, +180] º), simulando cambios de punto de vista que son comunes en la navegación autónoma a medida que el robot se mueve por el entorno y cambia su orientación.
- **Eliminar Bloque**: una región cuboide de tamaño variable (por ejemplo, 25 % del volumen de la nube) se elimina de la nube de puntos. A diferencia de la eliminación de puntos, esta técnica simula grandes oclusiones, entrenando al modelo para manejar regiones significativas de datos faltantes.
- **Traslación de Bloque Radial**: una región cuboide de tamaño variable (por ejemplo, 20 % del volumen de la nube) se traslada a lo largo del eje radial relativo al centro de la nube de puntos. Esta técnica está diseñada específicamente para abordar variaciones en la predicción de profundidad causadas por diferentes condiciones de iluminación.

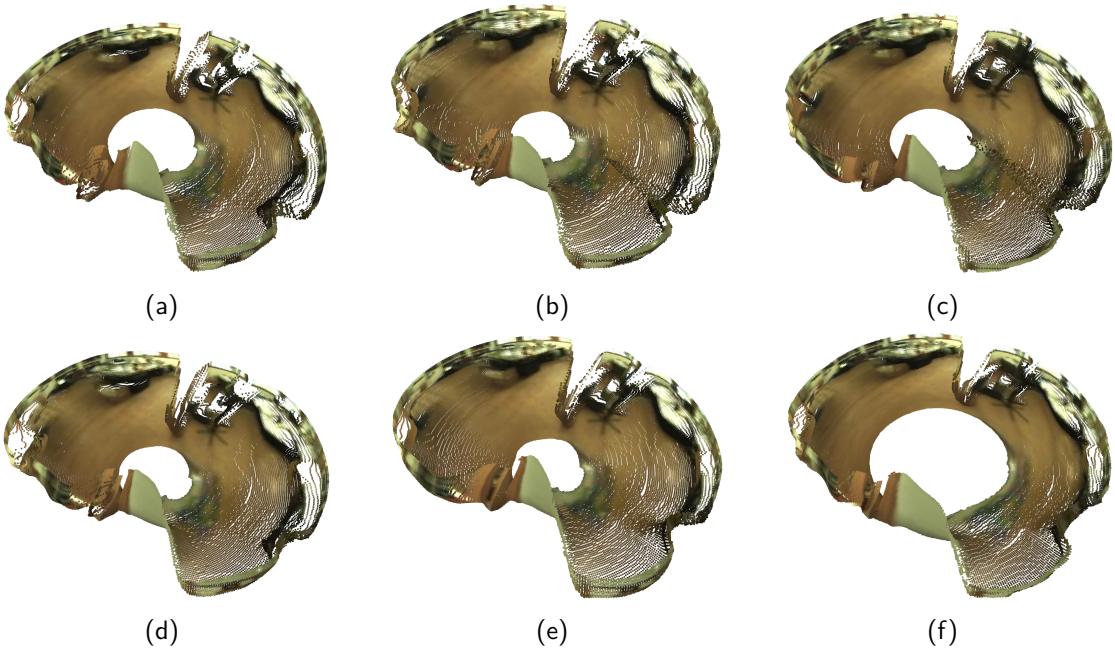


Figura 5.4: Ejemplo del efecto de Variaciones de Profundidad Destilada aplicado a una imagen nublada. (a) Distill Any Depth Grande, (b) Distill Any Depth Base, (c) Distill Any Depth Pequeño, (d) Depth Anything Grande, (e) Depth Anything Base y (f) Depth Anything Pequeño.

- **Escalado Planar**: la nube de puntos se escala uniformemente por un factor, que varía entre 0.8 y 1.2, a lo largo de los ejes x e y para emular cambios en la estimación de profundidad o variaciones en el tamaño de los objetos. Esta técnica también aborda variaciones en la predicción de profundidad bajo condiciones cambiantes de iluminación.
- **Distorsiones Elásticas**: esta técnica de aumento aplica deformaciones elásticas aleatorias a la nube de puntos, simulando transformaciones no rígidas que pueden ocurrir en escenarios del mundo real. El método genera una cuadrícula de ruido gaussiano suavizado, que se interpola y aplica a las coordenadas espaciales de la nube de puntos.
- ***Distilled Depth Variations***: esta es una novedosa técnica propuesta en este capítulo, que estima selectivamente la profundidad utilizando diferentes versiones destiladas de DAv2 (pequeño, base, grande) y Distill Any Depth (pequeño, base, grande). A diferencia de técnicas como Traslación de Bloque Radial o Escalado Planar, este método introduce distorsiones de profundidad basadas en las predicciones de modelos menos robustos (por ejemplo, las variantes pequeña y base). Al simular las inexactitudes de estimadores de profundidad menos robustos, se mejora la resiliencia del modelo de reconocimiento de lugares a errores de estimación de profundidad inherentes en los procesos de generación de pseudo-LiDAR. En la Figura 5.4 se muestran ejemplos de nubes de puntos generadas a partir de imágenes nubladas utilizando diferentes versiones del modelo Distill Any Depth y DAv2.

5.4 Experimentos

5.4.1 Conjunto de datos

Al igual que en el problema de reconocimiento visual de lugares, las imágenes escogidas en este capítulo pertenecen a la base de datos COLD [1], que puede descargarse desde https://www.cas.kth.se/COLD/. Esta base de datos está compuesta por varios entornos interiores: Freiburg A y B (FR-A, FR-B) y Saarbrücken A y B (SA-A, SA-B). En cada entorno, un robot móvil sigue diferentes trayectorias y captura imágenes omnidireccionales con un sistema catadióptrico. Estas imágenes han sido capturadas bajo tres condiciones de iluminación diferentes: nublado, noche y soleado. Además, las imágenes contienen personas en movimiento y cambios en la posición de los objetos de la escena. En general, este conjunto de datos presenta una variedad de condiciones desafiantes que validan eficazmente nuestro método.

La Tabla 5.1 muestra el número de imágenes que componen los conjuntos de entrenamiento, base de datos, evaluación y pruebas adicionales. El modelo MinkUNeXt se ha entrenado con 4338 imágenes capturadas en escenarios nublados del conjunto Freiburg A (FR-A *seq2_cloudy1* y *seq2_cloudy3*). Los otros escenarios y condiciones de iluminación se utilizan para evaluar la robustez y el rendimiento de nuestro sistema: FR-A (*seq2_cloudy2*, *seq2_night2*, *seq2_sunny2*), FR-B (*seq3_cloudy2*, *seq3_sunny2*), SA-A (*seq2_cloudy2*, *seq2_night1*) y SA-B (*seq4_cloudy2*, *seq4_night2*, *seq4_sunny1*). La base de datos consiste en imágenes nubladas de cada entorno y se obtiene muestreando fotogramas consecutivos para asegurar una distancia media de 20 cm entre puntos de captura. Las secuencias específicas elegidas para las bases de datos son FR-A (*seq2_cloudy3*), FR-B (*seq3_cloudy1*), SA-A (*seq2_cloudy3*) y SA-B (*seq4_cloudy1*).

	Entren.	**Mapa**	**Test**			**Extra test**		
	Nublado	Nublado	Nublado	Noche	Soleado	Nublado	Noche	Soleado
FR-A	4338	556	2595	2707	2114	-	-	-
FR-B	-	560	-	-	-	2008	-	1797
SA-A	-	586	-	-	-	2774	2267	-
SA-B	-	321	-	-	-	836	870	872

Tabla 5.1: Número de imágenes de entrenamiento y evaluación de los diferentes escenarios para las tres condiciones de iluminación.

5.4.2 Etiquetado y similitud

Al igual que en capítulos anteriores (Sección 4.4.2), cada imagen en el conjunto de datos está anotada con sus datos de pose, que sirven como *ground truth* de la trayectoria del robot. A continuación, se define el concepto de similitud entre nubes de puntos. Este concepto es crucial para el reconocimiento de lugares, ya que el modelo necesita ser entrenado con nubes de puntos estructuralmente similares capturadas desde la misma ubicación, así como con nubes de puntos estructuralmente diferentes capturadas desde distintas ubicaciones. La mayoría de los protocolos de similitud se

Parámetro	Valor
Distancia mínima de profundidad (d_{min})	1.0 m
Factor de escala de profundidad (d_s)	0.002 m/píxel
Factor de escala vertical (h_s)	0.015 m/píxel
Distancia positiva (p)	0,4 m
Distancia negativa (n)	0,4 m
Tamaño del lote (b)	512
Número de épocas	200 (50)
Tasa de aprendizaje inicial	1×10^{-3}
Épocas de reducción de LR	150, 180 (20, 30)
Decaimiento de peso L2	1×10^{-4}
Positivos por consulta (k)	16
Escala de cuantificación (qs)	0.01
Umbral de distancia (d)	0.5 m

Tabla 5.2: Parámetros de generación de nubes de puntos y entrenamiento

basan en la distancia euclídea entre las coordenadas desde las que se capturaron las nubes de puntos. Dos nubes de puntos se consideran similares si se capturan dentro de una distancia p y diferentes si se capturan desde una distancia mayor que n (donde $p \leq n$). Este enfoque asume que las muestras capturadas desde la misma vecindad compartirán una apariencia similar, lo que lo convierte en un método sencillo pero efectivo para el entrenamiento.

5.4.3 Detalles de implementación

En este capítulo, el modelo MinkUNeXt se vuelve a entrenar utilizando la función de pérdida *Truncated Smooth-AP* definida en la Sección 4.4.4. Para un rendimiento efectivo, esta función de pérdida necesita un tamaño de lote grande. En los experimentos de este capítulo, se utilizó un tamaño de lote de 512, y se empleó el optimizador AdamW para minimizar el error. Además, el análisis comparativo de la Sección 5.4.4 se realiza entrenando el modelo MinkUNeXt durante 50 épocas con un decaimiento de la tasa de aprendizaje (*Learning Rate*) en las épocas 20 y 30. Por el contrario, para la comparación con el estado del arte presentada en la Sección 5.4.5, el modelo se entrena durante 200 épocas disminución del *Learning Rate* en las épocas 150 y 180, para garantizar un rendimiento óptimo. Los parámetros y valores utilizados para generar las nubes de puntos y entrenar el modelo se resumen en la Tabla 5.2.

Todos los experimentos se llevan a cabo en una GPU NVIDIA GeForce RTX 3090 con 24 GB. Nuestro código está disponible públicamente en la página web del proyecto `https://juanjo-cabrera.github.io/projects-pL-MinkUNeXt/`.

5.4.4 Análisis comparativo

Esta sección estudia diferentes estimadores de profundidad de última generación, el efecto de la técnica de aumento de datos propuesta para pseudo-LiDAR y la selección

Estimador de profundidad	Nublado		Noche		Soleado		Promedio	
	R@1	R@1 %	R@1	R@1 %	R@1	R@1 %	R@1	R@1 %
Marigold (LCM) [260]	80.29	46.60	80.85	46.94	67.22	32.40	76.12	41.98
DepthPro [272]	80.25	94.72	77.70	94.29	60.22	86.33	72.72	91.78
DAv2 (Pequeño) [12]	89.64	97.83	92.21	99.48	79.94	94.18	87.26	97.16
DAv2 (Base) [12]	89.52	96.74	89.61	98.40	78.95	94.47	86.03	96.54
DAv2 (Grande) [12]	**91.42**	97.36	94.32	99.70	86.28	96.50	90.67	97.85
Distill Any Depth (Pequeño) [13]	90.26	**98.18**	90.46	98.79	76.92	93.57	85.88	96.85
Distill Any Depth (Base) [13]	91.19	97.83	93.32	**99.81**	84.34	96.17	89.62	97.94
Distill Any Depth (Grande) [13]	91.00	97.87	**94.92**	99.74	**88.41**	**97.40**	**91.44**	**98.34**

Tabla 5.3: Análisis comparativo de diferentes modelos de estimación de profundidad en el entorno Freiburg Parte A (FR-A). Se indica en negrita el mejor resultado de cada columna y se subraya el segundo mejor.

de las características visuales que acompañan a las nubes de puntos que alimentan al modelo MinkUNeXt.

5.4.4.1 Estimadores de profundidad

En este experimento, se emplean estimadores de profundidad de última generación y sus modelos destilados para generar mapas de profundidad a partir de imágenes panorámicas. Estos mapas de profundidad se convierten posteriormente en nubes de puntos pseudo-LiDAR, que se utilizan para entrenar el modelo MinkUNeXt para la tarea de reconocimiento de lugares. El conjunto de entrenamiento está compuesto por 4338 nubes de puntos generadas a partir de imágenes capturadas bajo condiciones de iluminación nubladas en el entorno Freiburg Parte A (FR-A). La Tabla 5.3 contiene los valores R@1 y R@1 % obtenidos con cada estimador de profundidad bajo tres condiciones de iluminación diferentes (nublado, noche y soleado) en el mismo entorno.

En cuanto a la comparación de los estimadores de profundidad de última generación (Tabla 5.3), DAv2-L [12] superó al resto de modelos en condiciones nubladas, que es la iluminación empleada tanto como información de entrenamiento como de base de datos (Sección 5.4.1). Sin embargo, la versión MT-Grande de Distilled Any Depth [13] logró los mejores resultados en condiciones nocturnas y soleadas, y también el mejor rendimiento promedio en este entorno. En comparación, otros modelos como Marigold (LCM) [260], DepthPro [272] o las variantes más pequeñas de DAv2 (Pequeño y Base) y Distilled Any Depth (MT-Pequeño y MT-Base) mostraron un *recall* competitivo pero inferior, particularmente en escenarios de iluminación desafiantes.

Basándonos en estos resultados, se selecciona Distilled Any Depth MT-Grande como el modelo para estimar datos pseudo-LiDAR, debido a su superior precisión y robustez en diversas condiciones de iluminación. Esta elección asegura una estimación de profundidad estable, la cual es crítica para el éxito de la tarea de reconocimiento de lugares.

Efecto de aumento	Nublado		Noche		Soleado		Promedio	
	R@1	R@1 %	R@1	R@1 %	R@1	R@1 %	R@1	R@1 %
Base	91.00	97.87	94.92	99.74	88.41	97.40	91.44	98.34
Eliminación de puntos	91.31	97.75	94.92	99.59	87.38	97.02	91.20	98.12
Rotación	**91.89**	98.21	94.84	**99.85**	86.28	96.78	91.00	98.28
Eliminar bloque	90.45	**98.72**	94.55	99.78	86.71	97.50	90.57	**98.67**
Traslación de bloque radial	91.00	98.33	**95.44**	99.78	87.94	97.45	91.46	98.52
Escalado planar	91.23	97.40	95.18	99.70	87.56	97.78	91.32	98.29
Distorsiones elásticas	90.96	97.59	95.32	99.74	86.00	97.30	90.76	98.21
Distilled Depth Variations	91.62	97.63	94.84	99.78	**89.45**	**98.06**	**91.97**	98.49

Tabla 5.4: Evaluación del aumento de datos en Freiburg A. Se indica en negrita el mejor resultado de cada columna y se subraya el segundo mejor.

5.4.4.2 Aumento de datos

En cuanto al aumento de datos, el método propuesto *Distilled Depth Variations*, que emplea mapas de profundidad derivados de estimadores de profundidad destilados, se compara con efectos convencionales de aumento de datos de nubes de puntos (Sección 5.3.5). La Tabla 5.4 presenta las métricas R@1 y R@1 % para cada efecto de aumento en diferentes condiciones de iluminación.

En primer lugar, el modelo base (sin aumento de datos) exhibió valores considerablemente altos de R@1 % y R@1 en todas las condiciones, con una ligera degradación en la precisión bajo condiciones soleadas. Esto resalta la falta de robustez de los modelos estimadores de profundidad cuando varían las condiciones de iluminación, particularmente en entornos soleados.

Entre los efectos de aumento individuales, que se aplicaron con una probabilidad del 40 %, los efectos de aumento clásicos no condujeron a mejores resultados al considerar el promedio de las tres condiciones de iluminación. Algunos de ellos, como el efecto de rotación, produjeron un mejor rendimiento de MinkUNeXt bajo condiciones nubladas. Por contra, el valor de R@1 ha disminuido sustancialmente en escenarios nocturnos y soleados, causado por un sobreajuste del modelo a la condición de entrenamiento. Mientras tanto, el enfoque propuesto mejoró claramente el R@1 en condiciones nubladas (+0.62 %) y soleadas (+1.04 %), con respecto al experimento base. A pesar de que el *recall* no mejoró en condiciones nocturnas (-0.08 %), el valor base de R@1 ya era alto, por lo que esta técnica de aumento de datos condujo a una solución más equilibrada para todas las condiciones de iluminación.

5.4.4.3 Características de entrada

Esta sección analiza las características visuales asociadas a las coordenadas 3D de cada punto. La Tabla 5.5 presenta la evaluación de diferentes características en las diversas condiciones de iluminación. Las características empleadas en este experimento están relacionadas con información de color (RGB, escala de grises y tono) o intensidad (gradiente).

Los resultados muestran el impacto de la selección de características en la robus-

Características visuales	Nublado		Noche		Soleado		Promedio	
	R@1	R@1 %	R@1	R@1 %	R@1	R@1 %	R@1	R@1 %
Info. no visual	91.62	97.63	94.84	99.78	89.45	**98.06**	91.97	98.49
RGB	92.20	98.29	94.29	**99.89**	86.14	95.60	90.88	97.92
Escala de grises	92.05	97.83	95.84	99.85	89.74	97.49	92.54	98.39
Tono	91.93	97.79	94.88	99.55	88.69	97.78	91.83	98.37
Gradiente (Mag.)	91.73	97.40	95.18	99.55	89.12	**98.06**	92.01	98.34
Gradiente (Arg.)	93.13	**99.11**	95.36	99.70	88.46	97.73	92.32	**98.85**
Gradiente (Mag., Arg.)	92.82	98.72	**95.99**	99.74	89.36	98.01	92.72	98.82
[+]Gradiente (Mag., Arg.)	**94.49**	**99.11**	95.92	99.67	**90.73**	97.78	**93.71**	**98.85**

[+] Resultados obtenidos para 200 épocas.

Tabla 5.5: Evaluación de diferentes características de entrada en Freiburg A. Se indica en negrita el mejor resultado de cada columna y se subraya el segundo mejor.

tez y precisión del modelo de reconocimiento de lugares, particularmente en escenarios de iluminación desafiantes. En términos generales, cuando se añaden características visuales a la nube de puntos, el modelo logró un fuerte rendimiento bajo las mismas condiciones de iluminación que en el entrenamiento (nublado). Además, el uso de algunas características particulares, como la escala de grises o el gradiente, condujo a mejores resultados en escenarios nocturnos. Sin embargo, el rendimiento se degrada generalmente en condiciones soleadas, especialmente para RGB, destacando las limitaciones de las características de color para manejar imágenes brillantes y sobreexpuestas.

Como regla general, el uso de las características de gradiente (tanto magnitud como argumento) condujo al mejor equilibrio entre las tres condiciones de iluminaciones. En consecuencia, este experimento se extendió a 200 épocas, mientras que el resto de los experimentos consistieron en 50 épocas. En este caso, se observó una mejora significativa en cada condición de iluminación, en particular bajo condiciones nubladas (+2.87 %) y soleadas (+1.28 %), con respecto al modelo entrenado sin características visuales.

5.4.5 Comparación con el estado del arte

Para evaluar la calidad del método propuesto en este capítulo, se compara tanto con enfoques propuestos de capítulos anteriores (Sección 3.9) como con métodos actuales del estado del arte en VPR. Las Tablas 5.6 y 5.7 presentan las métricas R@1 y R@1 %, respectivamente, para cada técnica en todos los entornos y condiciones de iluminación. El mejor resultado en cada columna se resalta en negrita, mientras que el segundo mejor resultado está subrayado.

Los resultados muestran que nuestro enfoque pL-MinkUNeXt supera consistentemente a los métodos competidores en términos de precisión global, alcanzando un R@1 del 87.31 % y un R@1 % del 97.52 %. Esta superioridad es particularmente notable en condiciones desafiantes, como escenarios con iluminación soleada (Freiburg-A), donde logra un R@1 del 90.73 %, superando al segundo mejor método (MixVPR) por un margen significativo de 7.43 %. Este rendimiento robusto en condiciones variables

R@1	Freiburg-A			Freiburg-B		Saarbrücken-A		Saarbrücken-B			
	Nublado	Noche	Soleado	Nublado	Soleado	Nublado	Noche	Nublado	Noche	Soleado	Global
Triplet VGG16 [252]	83.70	83.20	61.00	55.60	48.20	33.70	17.60	38.50	32.60	51.40	50.55
AnyLoc [58]	89.40	92.46	78.67	84.16	90.93	73.79	66.03	88.28	78.97	83.94	82.66
SALAD [59]	90.64	90.80	80.13	84.46	90.71	75.59	66.83	85.65	82.64	84.17	83.16
CosPlace [16]	91.64	92.21	79.99	85.76	92.60	76.56	62.68	91.39	80.57	86.58	84.00
Eigenplaces [54]	92.22	93.02	81.41	85.31	**93.16**	77.92	64.31	91.75	80.23	86.12	84.55
MixVPR [56]	90.98	93.50	83.30	84.01	92.82	76.56	70.53	91.15	81.61	87.16	85.16
Single VGG16	90.88	91.21	55.16	84.46	82.86	74.27	51.87	84.93	75.98	81.88	77.35
Siamese VGG16	89.29	94.10	71.24	**86.40**	82.75	77.79	49.25	81.10	73.79	84.63	79.03
pL-MinkUNeXt	**94.49**	**95.92**	**90.73**	85.71	91.32	**79.12**	**71.25**	**93.18**	**83.10**	**88.30**	**87.31**

Tabla 5.6: Comparación con otros métodos de VPR en diferentes entornos en términos de R@1. Se indica en negrita el mejor resultado de cada columna y se subraya el segundo mejor.

R@1 %	Freiburg-A			Freiburg-B		Saarbrücken-A		Saarbrücken-B			
	Nublado	Noche	Soleado	Nublado	Soleado	Nublado	Noche	Nublado	Noche	Soleado	Global
Triplet VGG16 [252]	96.18	98.41	81.60	83.27	73.73	66.36	42.52	55.50	55.63	73.85	72.71
AnyLoc [58]	98.69	99.22	97.92	91.19	**98.05**	95.16	92.24	99.52	91.95	98.28	96.22
SALAD [59]	98.69	99.45	97.87	90.69	96.49	95.16	92.46	98.44	93.91	97.59	96.08
CosPlace [16]	98.77	99.37	96.40	91.48	96.38	95.29	85.84	98.92	90.69	98.85	95.19
EigenPlaces [54]	98.15	99.45	95.79	91.48	95.88	95.34	88.18	99.28	90.00	98.85	95.24
MixVPR [56]	97.53	99.48	**98.01**	89.79	96.38	94.33	90.43	99.40	90.57	**99.31**	95.52
Single VGG16	**99.50**	99.74	82.69	91.48	94.21	98.34	82.27	95.22	88.62	97.36	92.94
Siamese VGG16	97.52	**99.78**	95.03	91.33	94.05	98.53	78.80	92.70	89.08	97.82	93.46
pL-MinkUNeXt	99.11	99.67	97.78	**92.48**	97.89	**98.94**	**95.31**	**100.0**	**94.71**	**99.31**	**97.52**

Tabla 5.7: Comparación con otros métodos de VPR en diferentes entornos en términos de R@1 %. Se indica en negrita el mejor resultado de cada columna y se subraya el segundo mejor.

de iluminación subraya la capacidad de nuestro modelo para extraer características discriminantes e invariantes a cambios ambientales.

En entornos nocturnos, pL-MinkUNeXt también demuestra una clara ventaja, con un R@1 del 95.92 % en Freiburg-A y 71.25 % en Saarbrücken-A, superando a métodos muy sofisticados del estado del arte. Esta robustez puede atribuirse al uso de características visuales invariantes a los cambios de iluminación y a un entrenamiento basado en el efecto *Distilled Depth Variations*.

A su vez, se muestran los enfoques presentados en la Sección 3.9, Single VGG16 y Siamese VGG16, que muestran un rendimiento inferior (77.35 % y 79.03 % de R@1 global, respectivamente) en comparación con métodos más sofisticados. Sin embargo, estos enfoques aún logran resultados competitivos en condiciones de iluminación nubladas y nocturnas en Freiburg-A, especialmente en términos de R@1 % (99.50 % y 99.78 %, respectivamente). Esto sugiere que, aunque estos métodos son menos complejos, pueden ser efectivos en escenarios específicos donde la variabilidad de la iluminación es limitada.

Entre los métodos competidores, MixVPR muestra el segundo mejor rendimiento global (85.16 % R@1), seguido de cerca por Eigenplaces (84.55 %) y CosPlace (84.00 %). Sin embargo, incluso estos métodos avanzados presentan limitaciones en ciertos escenarios.

Estos resultados demuestran que la arquitectura pL-MinkUNeXt representa un avance significativo en el estado del arte de VPR, combinando efectivamente representaciones estructurales y visuales para lograr un reconocimiento de lugares robusto ante las diversas condiciones ambientales que caracterizan los entornos del mundo real.

5.5 Resultados cualitativos de la tarea de reconocimiento de lugares

En esta sección se presentan ejemplos visuales de los resultados obtenidos por pL-MinkUNeXt. Estos ejemplos ilustran la capacidad del método bajo diferentes condiciones ambientales (Figuras 5.5, 5.6 y 5.7) y entornos nunca vistos antes (Figuras 5.8, 5.9 y 5.10). En cada figura, se observa la nube de puntos pseudo-LiDAR y la predicción de la nube más cercana en el espacio del descriptor de la base de datos y se comprueba si coincide con la nube más cercana en el espacio métrico de la posición. Las posiciones del mapa se representan con puntos azules, la posición actual con una cruz roja, la posición predicha con un círculo amarillo y la posición real con un anillo verde.

Los ejemplos de las Figuras 5.5, 5.6 y 5.7 muestran el rendimiento del método en el entorno Freiburg Parte A (FR-A) bajo diferentes condiciones de iluminación. En la Figura 5.5, se muestra un ejemplo en el que las condiciones de iluminación son nubladas tanto en el test como en la base de datos. En este caso, el método logra una predicción correcta donde la nube predicha coincide en orientación con la nube actual, aunque la más cercana de la base de datos está rotada 180º. En el ejemplo de la Figura 5.6, en condiciones nocturnas, el método también logra una predicción correcta, aunque la nube predicha está ligeramente desplazada. En el ejemplo de la Figura 5.7, en condiciones soleadas, se muestra un error puntual que presenta el conjunto de datos, en el que ciertas imágenes que conforman el test tienen asignada una posición incorrecta durante los primeros instances de la secuencia. Específicamente, se trata de unas 50 imágenes que están desplazadas en el mapa, lo que provoca un deterioro en los resultados aunque las predicciones sean correctas.

En las Figuras 5.8, 5.9 y 5.10, se presentan ejemplos de otros entornos, como Freiburg Parte B (FR-B), Saarbrücken Parte A (SA-A) y Saarbrücken Parte B (SA-B), respectivamente. En estos ejemplos, se observa que el método es capaz de generalizar a diferentes entornos y condiciones de iluminación. En el ejemplo de la Figura 5.8, en condiciones soleadas, el método logra una predicción correcta a pesar de la variación de iluminación. En el ejemplo de la Figura 5.9, en condiciones nubladas, el método también logra una predicción correcta. En este caso, aunque la base de datos y la nube actual han sido capturadas en condiciones nubladas, el cambio de iluminación es muy drástico. Por último, en el ejemplo de la Figura 5.10 capturado bajo condiciones nocturnas, se observa un error puntual en la predicción, pero el método aún logra una buena aproximación a la posición real. Para ver más ejemplos, visite la página del proyecto https://juanjo-cabrera.github.io/projects-pL-MinkUNeXt/.

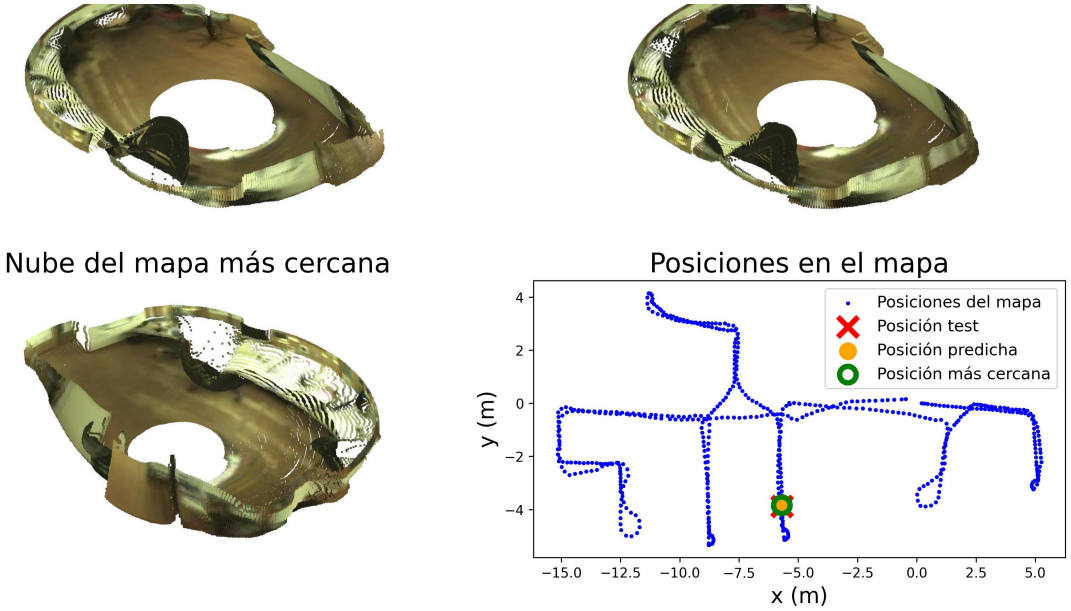


Figura 5.5: Ejemplo de predicción exitosa en condiciones nubladas con pL-MinkUNeXt en el entorno FR-A.

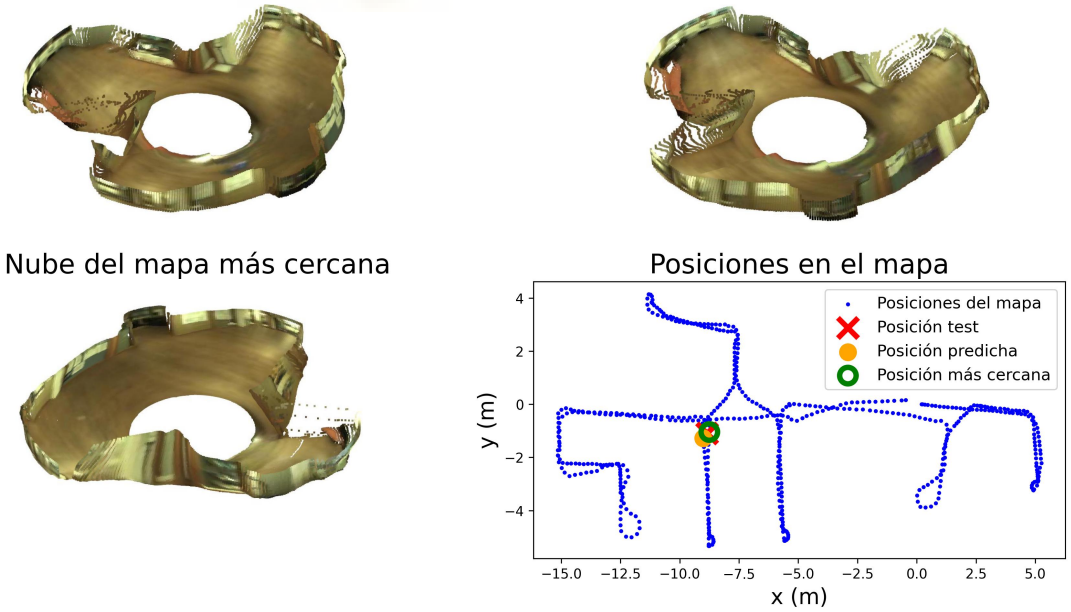


Figura 5.6: Ejemplo de predicción exitosa en condiciones nocturnas con iluminación artificial y sin perturbaciones lumínicas del exterior con pL-MinkUNeXt en el entorno FR-A.

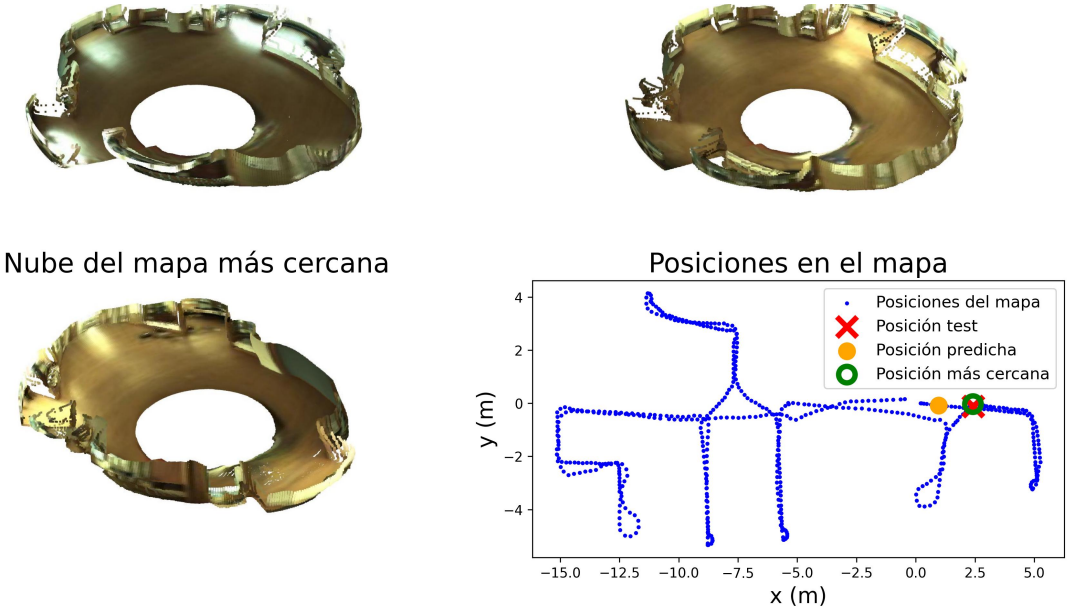


Figura 5.7: Ejemplo a priori erróneo en condiciones soleadas, pero realmente existe un error en las coordenadas de las imágenes del mapa, con pL-MinkUNeXt en el entorno FR-A.

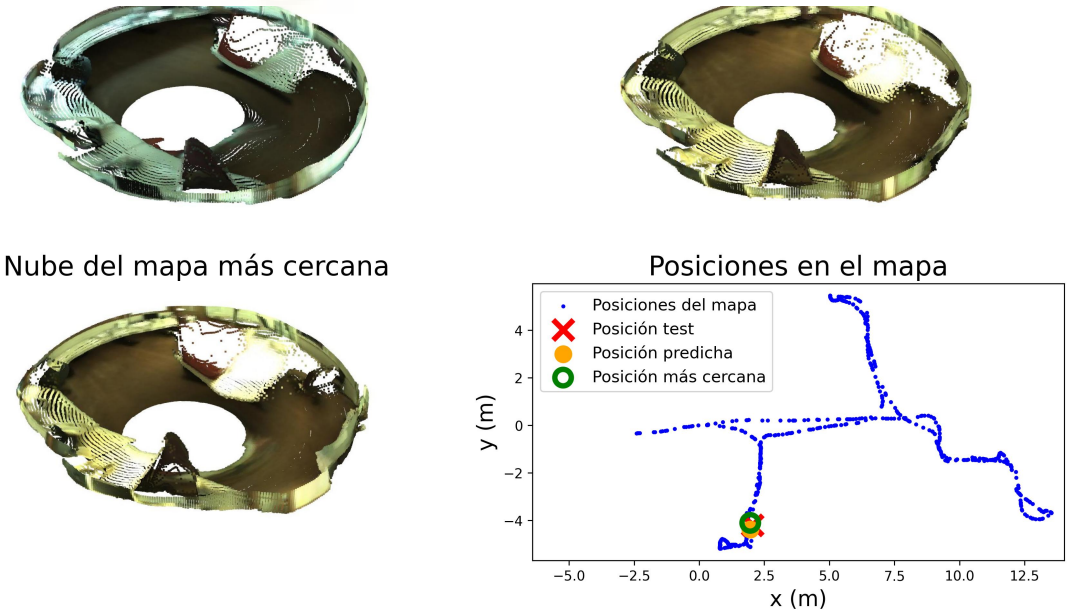


Figura 5.8: Ejemplo de predicción correcta en condiciones soleadas pese al cambio de iluminación, con el método pL-MinkUNeXt en el entorno FR-B.

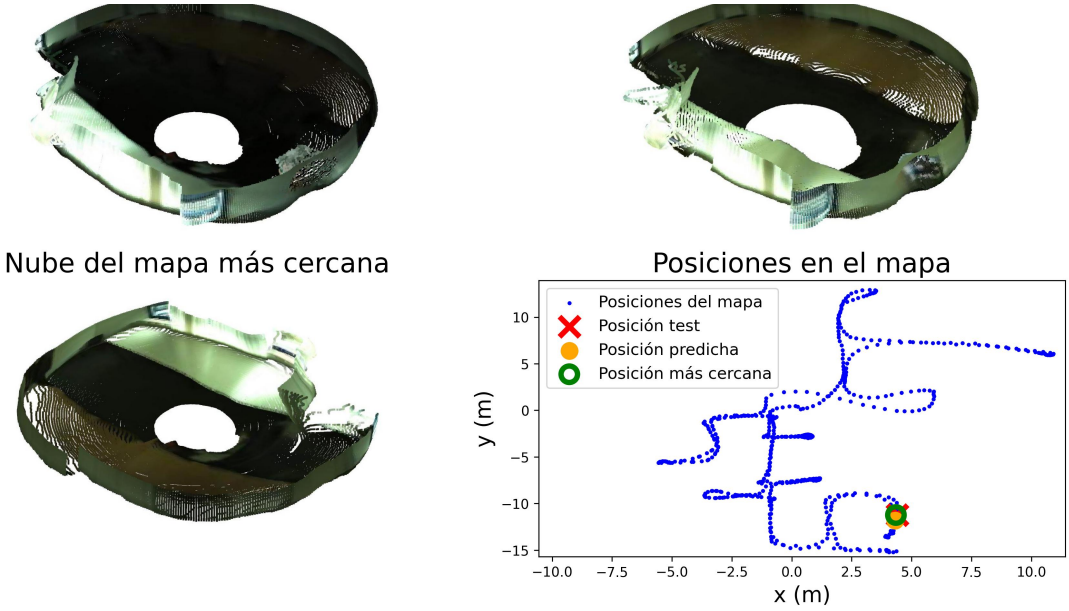


Figura 5.9: Ejemplo de predicción correcta en condiciones nubladas pese al fuerte cambio lumínico, con el método pL-MinkUNeXt en el entorno SA-A.

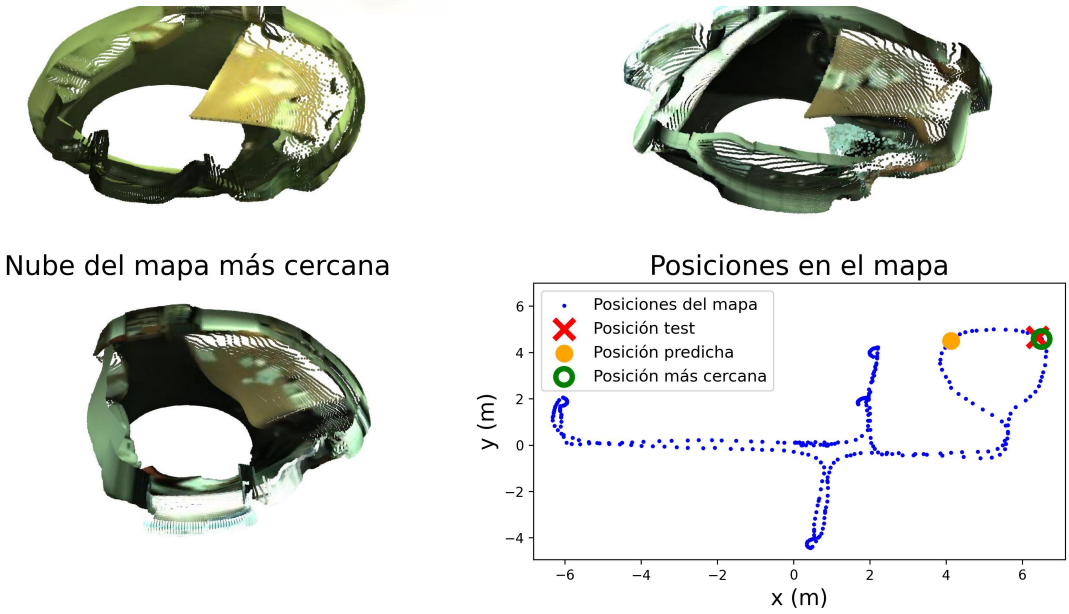


Figura 5.10: Ejemplo de predicción ligeramente errónea en condiciones nocturnas, con el método pL-MinkUNeXt en el entorno SA-B.

5.6 Conclusiones

En este capítulo se ha presentado un método para abordar el reconocimiento de lugares utilizando nubes de puntos pseudo-LiDAR a partir de imágenes panorámicas. Para crear estas nubes de puntos, se genera un mapa de profundidad de cada imagen panorámica utilizando Distill Any Depth, un estimador de profundidad de última generación que aprovecha los beneficios de la destilación. La información pseudo-LiDAR resultante se introduce posteriormente en MinkUNeXt para obtener descriptores de apariencia global. A pesar de que los sensores de visión se ven muy afectados por los cambios naturales y artificiales de la iluminación, nuestro enfoque demuestra una gran capacidad para aumentar la invariancia de los descriptores aprendidos por MinkUNeXt en entornos y condiciones de iluminación desafiantes, presentándose como una solución efectiva para el reconocimiento de lugares.

Además, se propone una novedosa técnica de aumento de datos, que consiste en entrenar la red con diferentes estimadores de profundidad, es decir, Distill Any Depth y DAv2, junto con sus versiones destiladas (Pequeño, Base, Grande). Adicionalmente, las nubes de puntos se enriquecen con características visuales basadas en el gradiente de intensidad. En resumen, el enfoque propuesto demuestra un excelente rendimiento en el reconocimiento de lugares debido a la combinación de nubes de puntos y estrategias eficientes de aumento de datos, basadas en el uso de estimadores de profundidad destilados.

Dado que la aplicabilidad del método actual se limita a cámaras omnidireccionales, el trabajo futuro se centrará en integrar datos visuales con técnicas de estimación de profundidad en el dominio de imagen, explorando estrategias de fusión temprana y tardía. Mediante la combinación de estas modalidades, el objetivo es extender este enfoque a otros tipos de cámaras, mejorando su versatilidad para diversas plataformas robóticas.

6

Reconocimiento cruzado de lugares entre diferentes modalidades de sensor: LiDAR y cámaras *fisheye*

6.1 Introducción

En capítulos anteriores se ha llevado a cabo el reconocimiento de lugares mediante un mismo tipo de sensor, ya sea LiDAR o cámara, de manera que se utiliza el mismo sensor tanto para capturar la base de datos como para adquirir las observaciones durante la navegación. Este enfoque es efectivo y ha demostrado ser robusto en diversas condiciones ambientales y estructurales del entorno.

Sin embargo, en ocasiones nos podemos encontrar en una situación diferente: el sensor con el que se ha capturado la base de datos difiere del sensor instalado sobre la plataforma móvil. Así pues, por ejemplo, el nuevo sensor con el que cuenta el robot podría tener mayor resolución, precisión o distancia máxima de detección. Esta situación es, en general, fácilmente subsanable, pues los datos se pueden filtrar para acomodarse a la base de datos original. En este capítulo, por otra parte, nos centramos en el problema del reconocimiento de lugares utilizando un sensor de una modalidad completamente diferente al que se utilizó para capturar la base de datos de un entorno específico. En este caso, nos centramos en desarrollar una técnica que permite utilizar la base de datos original empleando un sensor completamente diferente. De esta manera, se evitan los costes de capturar y almacenar una nueva base de datos con el nuevo sensor, lo cual puede ser costoso. Además, los sistemas multi-robot pueden disponer de diferentes configuraciones sensoriales para operar en un mismo entorno, por lo que es necesario desarrollar métodos que permitan realizar el reconocimiento de lugares entre diferentes modalidades de sensor, como cámaras y LiDARs.

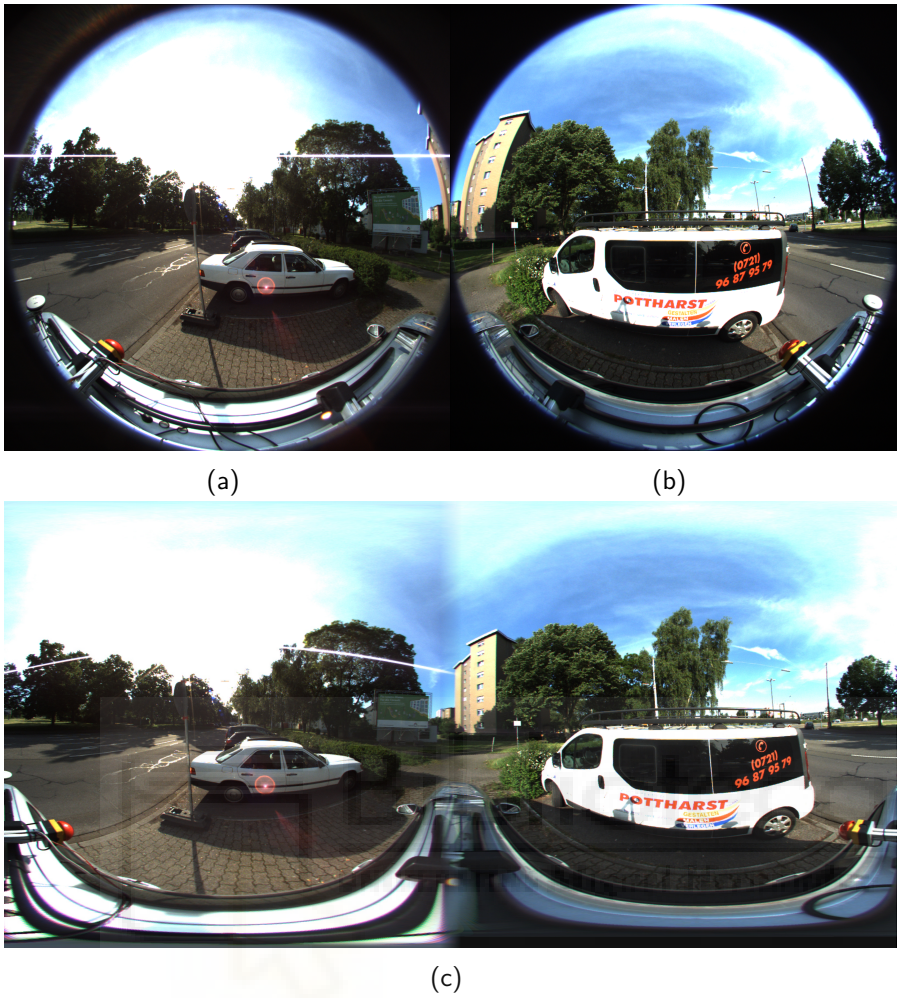


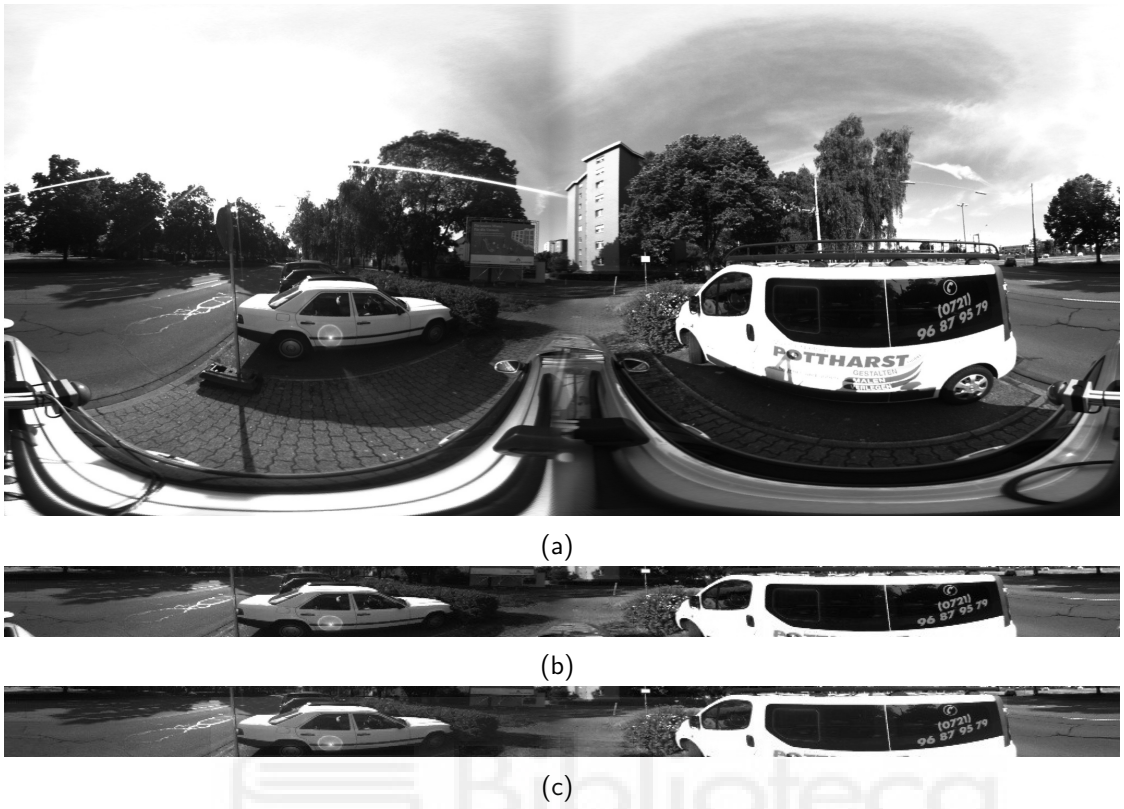
Figura 6.1: Transformación de imagen *fisheye* izquierda (a) y derecha (b) a representación equirectangular (c).

El enfoque presentado en este capítulo tiene un gran potencial práctico, ya que elimina la necesidad de mantener el mismo tipo de sensor para la captura de la base de datos y las lecturas empleadas para llevar a cabo el reconocimiento de lugares. De esta manera, se pueden incorporar a la plataforma robótica sensores más avanzados como LiDARs o nuevos robots al sistema con diferente configuración sensorial, sin necesidad de capturar nuevos datos. También se abre la posibilidad de crear una base de datos usando un sensor LiDAR y, a continuación, utilizar cámaras RGB, menos costosas computacionalmente, durante la operación de los robots. Para lograr esto, se han de transformar las lecturas de los diferentes tipos de sensor al mismo espacio del descriptor. Este capítulo aborda este desafío, proponiendo un método para el reconocimiento de lugares entre modalidades de sensor LiDAR y cámaras *fisheye*. En concreto, se propone transformar tanto las imágenes *fisheye* como las lecturas LiDAR al espacio de la profundidad y semántico, lo que permite utilizar una única arquitectura de red para realizar el reconocimiento de lugares de manera robusta y eficiente.

6.1.1 Contribuciones de este capítulo

En este capítulo, se utiliza la base de datos KITTI-360 [3] para diseñar y evaluar la técnica propuesta. Las imágenes capturadas por cámaras *fisheye* se transforman a una imagen 360º equirectangular que a su vez se convierte al dominio de la intensidad (escala de grises), al dominio de la profundidad por medio de Depth Anything V2 [12] y se segmenta semánticamente por medio de SegFormer [15]. Por otro lado, las lecturas LiDAR se convierten a su vez en imágenes de intensidad e imágenes de rango mediante la proyección esférica y en imágenes de LiDAR segmentadas por medio de MinkUNet [11]. Este método permite realizar el reconocimiento de lugares por medio de un único modelo de red neuronal, el cual admite como entrada datos de intensidad, profundad y segmentación semántica de LiDAR y cámara. En concreto, se ha seleccionado la arquitectura CosPlace [16] para llevar a cabo el reconocimiento de lugares debido a su simplicidad y eficacia. Las principales contribuciones de este capítulo son las siguientes:

- **CrossPlace:** un novedoso método para reconocimiento de lugares entre modalidades heterogéneas de sensor (LiDAR y cámaras *fisheye*) que transforma los dos datos de entrada diferentes al espacio común de la intensidad, profundidad y semántica, permitiendo utilizar una única arquitectura de red para ambas modalidades de sensor.
- **Proceso de transformación de modalidades:** una metodología de procesamiento para convertir tanto imágenes *fisheye* como nubes de puntos LiDAR al espacio de la intensidad, profundidad e información semántica. Esta transformación incluye la generación de imágenes equirectangulares a partir de dos cámaras *fisheye* y su conversión a mapas de profundidad y semánticos mediante modelos avanzados de procesamiento de imágenes y nubes de puntos.
- **Optimización de representaciones de intensidad, profundidad y semántica:** un conjunto de técnicas específicamente diseñadas para mejorar la calidad de las representaciones de intensidad, profundidad y semántica, incluyendo la densificación de los datos LiDAR en la proyección esférica, la eliminación del vehículo que transporta los sensores presente en los datos capturados y la potenciación de las imágenes de profundidad predichas para mejorar la discriminación de objetos distantes.
- **Arquitectura unificada:** adaptación de la arquitectura CosPlace para el procesamiento eficiente de imágenes de intensidad, profundidad y semántica procedentes de diferentes modalidades de sensor, demostrando que un único modelo puede realizar reconocimiento *cross-modal* de lugares de manera efectiva.
- **Entrenamiento *cross-modal* unificado:** una estrategia de aprendizaje que selecciona ejemplos positivos y negativos independientemente del tipo de sensor, permitiendo que el modelo aprenda una representación común en el espacio del descriptor sin necesidad de procedimientos de destilación entre modelos independientes.
- **Eliminación del paradigma *teacher-student*:** a diferencia de otros enfoques previos que requieren entrenar primero un modelo de red para una modalidad y luego destilar el conocimiento a otro, CrossPlace entrena directamente con todas las modalidades de sensor de manera conjunta, simplificando el proceso y



(a)

(b)

(c)

Figura 6.2: Imagen equirectangular en escala de grises(a); recortada al FOV del LiDAR (b); e *inpainting* para eliminar oclusiones el vehículo que transporta los sensores (c).

mejorando la coherencia entre representaciones de diferentes sensores.

- **Fusión de información de intensidad, profundidad y semántica:** integración de información de intensidad, profundidad y semántica en un espacio común, tanto de manera temprana como tardía, lo que permite al modelo aprender representaciones más ricas y discriminantes para el reconocimiento de lugares.
- **Evaluación exhaustiva:** análisis detallado del rendimiento de CrossPlace en la base de datos KITTI-360, demostrando la viabilidad del reconocimiento de lugares con sensores diferentes en entornos urbanos y autovías. Se demuestra que este enfoque supera a otros métodos en términos de R@1, estableciendo un nuevo estándar para el reconocimiento *cross-modal* de lugares entre LiDAR y cámaras *fisheye*.

6.2 Trabajos relacionados

El reconocimiento de lugares entre diferentes modalidades de sensor es un área emergente en la robótica móvil. Tradicionalmente, los sistemas de reconocimiento de lugares han utilizado el mismo tipo de sensor tanto para capturar la base de datos como para las lecturas empleadas para llevar a cabo el reconocimiento del lugar. Sin embargo, este enfoque presenta limitaciones prácticas, ya que los sensores pueden cambiar con el tiempo y es necesario adaptar los sistemas de reconocimiento a nuevas modalidades de sensor sin necesidad de capturar una nueva base de datos. En este contexto, surge

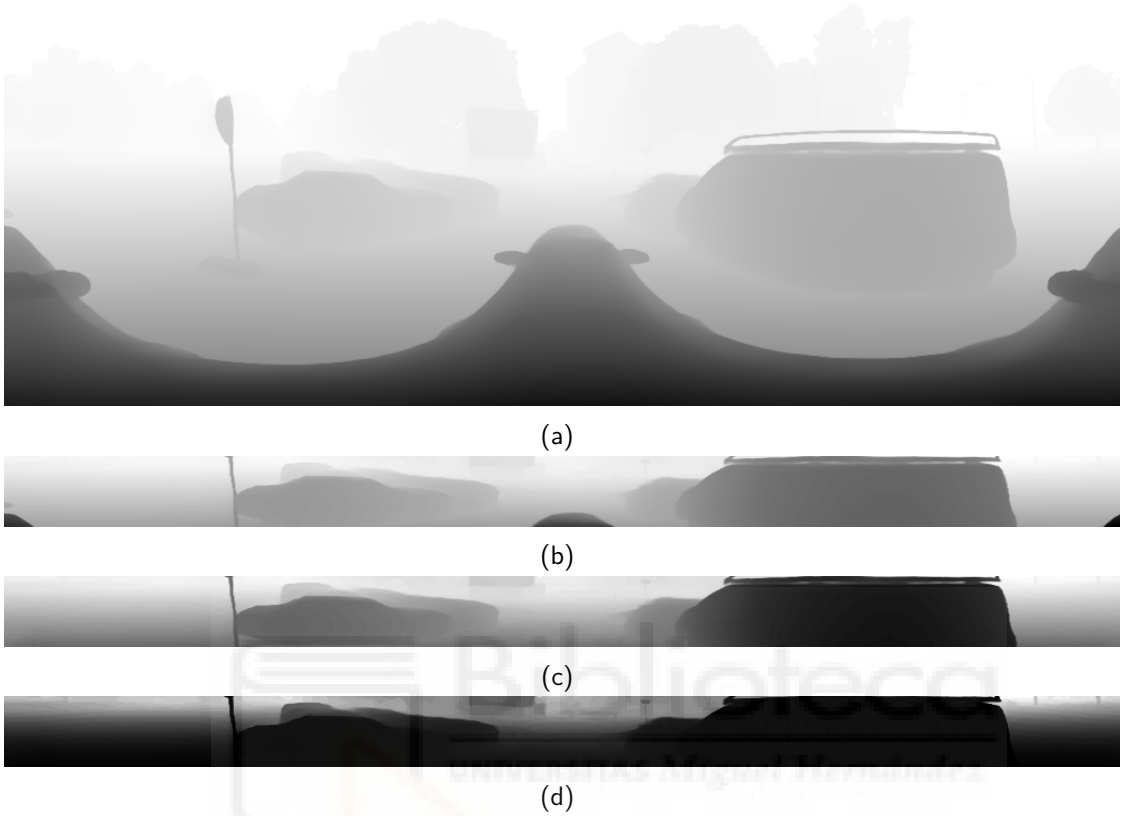
la necesidad de desarrollar métodos que permitan realizar el reconocimiento de lugares utilizando diferentes modalidades de sensor, como cámaras y LiDARs.

En años recientes han surgido diversos enfoques para abordar el reconocimiento de lugares entre diferentes modalidades de sensor. Cattaneo *et al*. [156] se centran en llevar al mismo espacio del descriptor imágenes estándar y nubes de puntos por medio de VGG16 [5] y PointNet [75], respectivamente. Por otro lado, Yin *et al*. [157] proponen i3dLoc, un método que busca robustez ante condiciones ambientales inconsistentes transformando imágenes equirectangulares en proyecciones de rango por medio de arquitecturas GAN (*Generative Adversarial Networks*) y utilizando una técnica de aprendizaje contrastivo para alinear las representaciones de imagen y nube de puntos. El método $(LC)^2$ [273] plantea transformar tanto las imágenes estándar como las nubes de puntos de LiDAR en imágenes de disparidad y rango, respectivamente. De este modo, se reduce la brecha entre las representaciones de imagen y nube de puntos. Posteriormente, emplean dos arquitecturas de red separadas para procesar las imágenes de disparidad y rango.

Zhao *et al*. [274] desarrollaron un sistema que utiliza mecanismos de atención para correlacionar imágenes equirectangulares y nubes de puntos por medio de ResNet-18 [6] y PointNet [75], respectivamente. I2P-Rec [275] proponen la proyección *Bird's Eye View* (BEV) tanto de la nube de puntos del LiDAR como para la nube estimada a partir de un par estéreo, con ello entrenan ResNet-34 para la extracción de características con NetVLAD[53] para la agregación de las mismas. Por otro lado, VXP [159] alinea correspondencias entre vóxeles (unidades 3D) y píxeles (unidades 2D) de manera autosupervisada, unificándolos en un espacio de características común. El método emplea DINO ViTs-8 [257] para la extracción de características de la imagen estándar y VoxelNet [160] para el LiDAR. Este método ha proporcionado los mejores resultados del estado del arte en el reconocimiento de lugares *cross-modal* entre imágenes estándar y nubes de puntos LiDAR en conjuntos de datos como Oxford RobotCar [2] y KITTI [276].

LIP-Loc [161] aplica el enfoque de CLIP [277], que lleva al mismo espacio del descriptor texto, audio e imágenes, pero en este caso para el reconocimiento de lugares entre imágenes estándar y nubes de puntos LiDAR, utilizando una técnica de aprendizaje contrastivo. Este método se centra en la alineación de características entre imágenes estándar y nubes de puntos proyectadas a imágenes de rango, logrando una representación común que permite el reconocimiento *cross-modal*. Por su parte, SaliencyI2PLoc [278] presenta un transformer dual basado en un *Vision Transformer* (ViT) [57] y PointNet [75] para el reconocimiento de lugares a partir de imágenes equirectangulares en bases de datos conformadas por nubes de puntos LiDAR.

Además, enfoques como DistilVPR [155] emplean técnicas de destilación de conocimiento *cross-modal* para transferir conocimiento desde extractores de características 3D a extractores de características de imagen, logrando una representación más robusta y consistente entre modalidades. Otros trabajos recientes como VOLoc [279] abordan el problema de la compresión de mapas LiDAR para consultas visuales eficientes. Además,



(a)

(b)

(c)

(d)

Figura 6.3: Imagen equirectangular de profundidad (a); recortada al FOV del LiDAR (b); *inpainting* para eliminar oclusiones el vehículo que transporta los sensores (c); y potenciación de la imagen de profundidad para mejorar la discriminación de objetos distantes (d).

UniLoc [162] se destaca como una solución universal para el reconocimiento de lugares a escala urbana, capaz de utilizar cualquier modalidad: texto, imagen o nube de puntos. Sin embargo, este método emplea información visual para colorear la nube de puntos LiDAR, lo que resulta incoherente con la tarea planteada en este capítulo, ya que se busca realizar el reconocimiento de lugares sin combinar la información de diferentes modalidades de sensor. Además, esa fusión de la información visual y el LIDAR requiere generalmente la calibración de ambos sistemas, lo que no siempre es sencillo [280].

A pesar de los avances en el reconocimiento de lugares entre diferentes modalidades de sensor, muchos de los enfoques existentes requieren entrenar modelos por separado para cada modalidad y luego destilar el conocimiento entre ellos. Esto puede ser ineficiente y complicado, especialmente cuando se trata de múltiples modalidades. La técnica propuesta en este capítulo, CrossPlace, busca superar estas limitaciones al transformar las lecturas de diferentes tipos de sensor al espacio común de la intensidad, profundidad e información semántica, permitiendo un entrenamiento unificado y una representación coherente entre modalidades.

6.3 Reconocimiento de lugares entre diferentes modalidades de sensor (cámaras *fisheye* y LiDAR) basado en un espacio común de la información

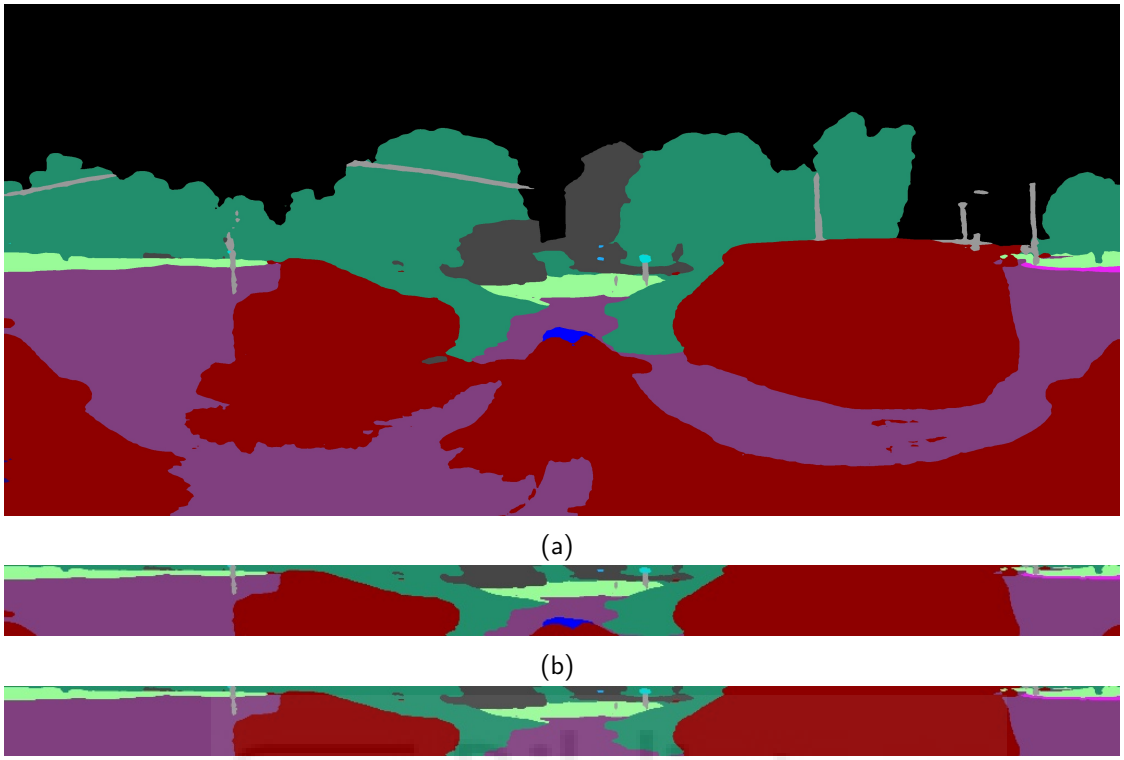
En esta sección se presenta el enfoque propuesto para el reconocimiento de lugares entre diferentes modalidades de sensor, específicamente entre cámaras *fisheye* y LiDARs. El método se basa en transformar las lecturas de ambos tipos de sensor al espacio común de la intensidad, la profundidad y de la segmentación semántica, lo que permite utilizar una única arquitectura de red tanto para LiDAR como para cámaras. A continuación se describen las características clave de cada uno de estos espacios:

1. Intensidad: el LiDAR puede proporcionar valores de intensidad para cada punto, mientras que las cámaras capturan la intensidad luminosa en imágenes en escala de grises. De este modo, se puede comparar directamente la intensidad de las lecturas LiDAR con las imágenes en escala de grises obtenidas a partir de las cámaras *fisheye*, aunque existen ciertas diferencias entre las modalidades, ya que la intensidad del LiDAR depende directamente de la cantidad de energía reflejada por la superficie en la que incidió el rayo láser (emitido por el sensor a una longitud de onda de 865 nm y 903 nm, en el caso de un sensor del fabricante Ouster y Velodyne, respectivamente). Esta intensidad LiDAR depende, por tanto, de las propiedades de la superficie y del ángulo de incidencia, mientras que la intensidad de las cámaras puede verse afectada por las condiciones de iluminación.

2. Profundidad: la profundidad representa la distancia desde el sensor hasta cada punto del entorno. Esta profundidad se puede obtener de forma directa a partir de las lecturas del LiDAR, que proporcionan una nube de puntos tridimensional con información de distancia. Por otro lado, las cámaras *fisheye* no proporcionan directamente información de profundidad, pero se pueden utilizar modelos de estimación de profundidad para generar mapas de profundidad a partir de las imágenes capturadas por las cámaras [12]. Estos mapas de profundidad representan la distancia relativa entre los objetos en la escena y el sensor, lo que permite una comparación directa con las lecturas LiDAR.

3. Semántica: la segmentación semántica asigna una etiqueta de clase a cada píxel de la imagen o punto de la nube. En ambos casos, se requiere un modelo de segmentación (2D en el caso de la imagen y 3D en el caso de la nube de puntos) para generar mapas semánticos que representen la distribución de diferentes clases en la escena, como edificios, vehículos, peatones, etc. Estos mapas semánticos son útiles para identificar regiones funcionales del entorno y pueden mejorar la discriminación entre diferentes lugares.

En esta sección, se describe el proceso de transformación de las imágenes *fisheye* y las nubes de puntos LiDAR al espacio común de la intensidad, profundidad e información semántica. Además, se detalla el preprocesamiento de cada tipo de dato para garantizar una representación coherente y útil para el reconocimiento de lugares. A continuación, en la Sección 6.3.1 se presenta la arquitectura unificada que permite realizar el reconocimiento de lugares utilizando una única red, independientemente del tipo de sensor utilizado para capturar los datos.



(a)

(b)

(c)

Figura 6.4: Imagen equirectangular segmentada semánticamente (a); recortada al FOV del LiDAR (b); e *inpainting* para eliminar oclusiones el vehículo que transporta los sensores (c).

6.3.1 Transformación de imágenes *fisheye* al espacio de la intensidad, profundidad y semántica

En el sistema de visión del conjunto de datos usado, las cámaras *fisheye* están rotadas 180º entre sí para capturar imágenes opuestas del entorno. Estas imágenes se combinan a su vez para generar una imagen 360º equirectangular. En este trabajo, empleamos la transformación geométrica basada en un polinomio propuesta por Flores *et al*. [281] para generar una única vista de 360º a partir de la información capturada por ambas lentes *fisheye*. Dado el par de imágenes *fisheye*, el primer paso consiste en convertir cada una de ellas a formato equirectangular. Para representar la imagen *fisheye* en un formato esférico, se lleva a cabo un mapeo inverso, es decir, dado un píxel en la imagen equirectangular, se proyecta sobre una esfera unidad y después sobre la imagen *fisheye*. De esta forma, cada píxel de la imagen final (imagen equirectangular) tiene asociado un valor RGB. Para llevar a cabo la segunda proyección, se ha empleado el modelo de cámara de Mei *et al*. [282] que proporciona la propia base de datos.

Una vez disponemos del par de imágenes equirectangulares, es importante calcular y aplicar una transformación geométrica para expresarlas en el mismo sistema de referencia. Tras realizar un estudio, Flores *et al*. [281] proponen que esta transformación geométrica entre ambas imágenes equirectangulares venga dada por un polinomio. Esta es la técnica que se emplea en el presente capítulo. Para poder estimar esta transformación, se requiere de correspondencias de puntos extraídos de ambas imágenes. Estas

correspondencias se obtienen mediante el uso de un detector de características, como ORB [18], que identifica puntos clave y sus descriptores en ambas imágenes *fisheye*. Posteriormente, se obtienen los puntos correspondientes para estimar la función polinómica que mejor describe la transformación entre las dos imágenes *fisheye*. En la Figura 6.1 (a), (b) y (c) se muestran las dos imágenes *fisheye* y la imagen equirectangular resultante del proceso de transformación. Cabe destacar que el modelo de transformación polinómica se calcula una sola vez y se aplica a todas las imágenes *fisheye* del conjunto de datos, lo que permite una conversión eficiente y consistente de las imágenes *fisheye* a imágenes equirectangulares. La unificación de las dos imágenes *fisheye* en una única imagen equirectangular no solo evita la redundancia de información, sino que también facilita la comparación directa con las representaciones generadas a partir del LiDAR. Al transformar ambas vistas en una sola imagen 360º, se obtiene una visión continua y sin solapamientos del entorno, lo que elimina duplicidades y zonas de información repetida presentes si se procesaran ambas imágenes por separado. Además, el hecho de trabajar con una sola imagen permite el uso de una arquitectura de red unificada para ambas modalidades de sensor, lo que simplifica el proceso de reconocimiento de lugares y mejora la eficiencia del sistema.

La transformación de la imagen equirectangular al espacio de la intensidad se realiza mediante una simple conversión a escala de grises (Figura 6.2 (a)), ya que las imágenes *fisheye* capturan información en color. Esta conversión es necesaria para equiparar las imágenes equirectangulares con las imágenes de intensidad generadas a partir de las lecturas LiDAR, que también se representan en escala de grises.

Seguidamente, se plantea la obtención de una imagen de profundidad a partir de la imagen 360º por medio del modelo Depth Anything V2 Large [12], el cual fue entrenado con imágenes estándar en una amplia variedad de escenarios y condiciones de iluminación, lo que lo hace adecuado para el reconocimiento de lugares en entornos urbanos complejos. Sin embargo, en este trabajo se testa su capacidad ante imágenes equirectangulares 360º que poseen distorsión. Por tanto, se emplea este modelo de estimación de profundidad para convertir la imagen equirectangular a un mapa de profundidad (Figura 6.3 (a)). Este mapa de profundidad proporciona información sobre la distancia relativa entre los puntos de la escena a las cámaras, lo cual aporta información relevante para el reconocimiento de lugares.

Para la segmentación semántica, se utiliza el modelo SegFormer [15] para generar una imagen de segmentación semántica a partir de la imagen equirectangular. Al igual que el modelo de profundidad, SegFormer fue entrenado con imágenes estándar para segmentar diferentes objetos en la escena, como edificios, vehículos y peatones. En este trabajo, se aplica sobre imágenes equirectangulares, lo que permite segmentar la imagen resultante (Figura 6.4 (a)).

Finalmente, las imágenes de intensidad, profundidad y semánticas resultantes se recortan al campo de visión (*Field Of View*, FOV) del sensor LiDAR, lo que permite una comparación directa con las lecturas de nubes de puntos convertidas a rango (Figura 6.2 (b), Figura 6.3 (b) y Figura 6.4 (b)). Además, se utiliza el modelo de *inpain-*

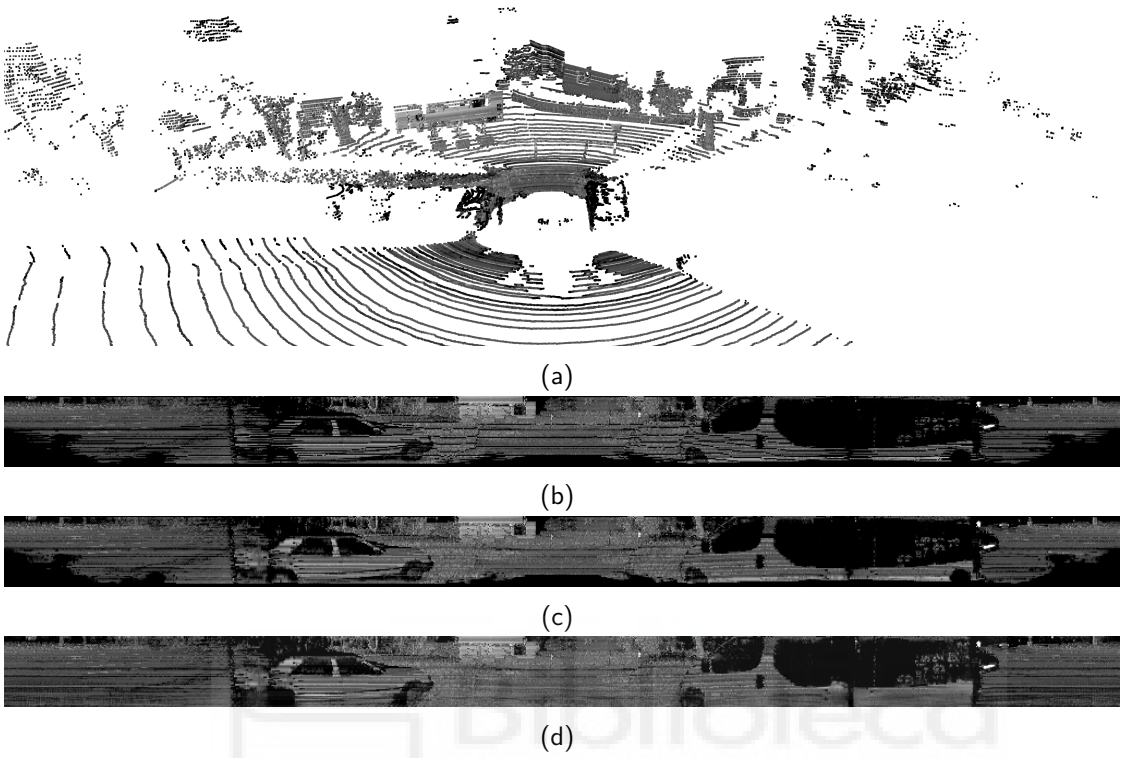


Figura 6.5: Proceso de transformación de nubes de puntos LiDAR a imágenes de intensidad. (a) Nube de puntos LiDAR con intensidad; (b) Imagen panorámica generada a partir de la nube de puntos con intensidad; (c) Imagen panorámica de intensidad interpolada verticalmente para completar los píxeles faltantes; y (d) Imagen panorámica de intensidad interpolada verticalmente e *inpainting* del vehículo que transporta los sensores.

ting LaMa [17], para eliminar oclusiones provocadas por el vehículo que transporta las cámaras *fisheye* (Figura 6.2 (c), Figura 6.3 (c)) y en el caso de la imagen segmentada semánticamente el vehículo se elimina directamente cambiando la categoría semántica de la zona de la imagen donde aparece el coche por la clase "carretera" (Figura 6.4 (c)). Además, la imagen de profundidad se eleva a la cuarta para conseguir un mayor contraste en las zonas más alejadas, lo que mejora la discriminación de edificios, vegetación y otros objetos distantes en el entorno (Figura 6.3 (d)).

6.3.2 Transformación de nubes de puntos LiDAR al espacio de la intensidad, profundidad y semántico

Las lecturas del sensor LiDAR (Figura 6.6 (a)) se proyectan en una representación panorámica bidimensional mediante una proyección esférica, con el objetivo de generar imágenes de profundidad equivalentes a partir de la nube de puntos tridimensional (Figura 6.6 (b)). Este procedimiento transforma las coordenadas cartesianas (x, y, z) en coordenadas angulares, en las que el eje horizontal representa el ángulo azimutal y el eje vertical representa el ángulo de elevación. El módulo de cada punto se utiliza como valor de profundidad (distancia al sensor). Las ecuaciones que definen esta proyección

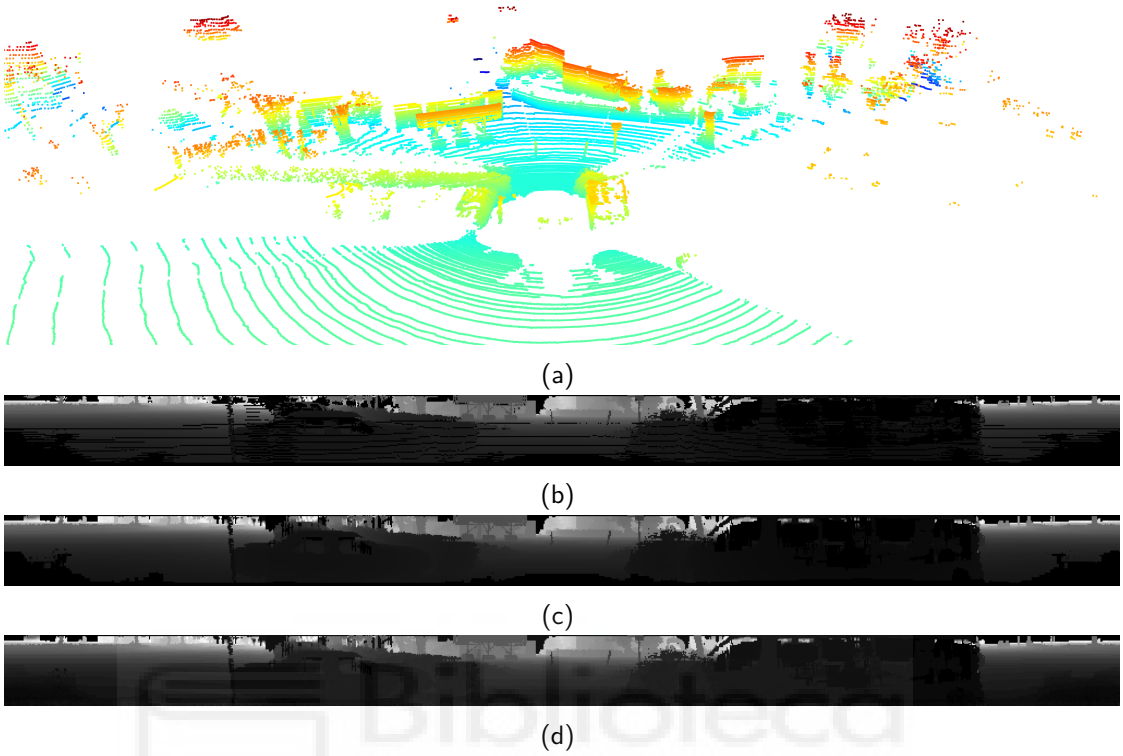


Figura 6.6: Proceso de transformación de nubes de puntos LiDAR a imágenes de rango. (a) Nube de puntos LiDAR; (b) Imagen panorámica generada a partir de la nube de puntos; (c) Imagen panorámica de profundidad interpolada verticalmente para completar los píxeles faltantes; y (d) Imagen panorámica de profundidad interpolada verticalmente e *inpainting* del vehículo que transporta los sensores.

son:

$$\begin{cases} r = \sqrt{x^2 + y^2 + z^2} \\ \phi = -\arctan 2(y, x) \\ \theta = \arcsin(z, r) \end{cases} \tag{6.1}$$

donde r representa la distancia euclídea al sensor, ϕ es el ángulo azimutal (horizontal), y θ es el ángulo de elevación (vertical). Esta proyección esférica es especialmente adecuada para sensores LiDAR con escaneo rotacional, ya que preserva la continuidad angular del entorno circundante.

Posteriormente, los valores angulares se discretizan de acuerdo con las resoluciones vertical y horizontal del sensor para construir una imagen panorámica en coordenadas (u, v), donde u y v son índices de píxel horizontales y verticales, respectivamente:

$$\begin{cases} u = \left\lfloor \dfrac{\phi}{\Delta\phi} \right\rfloor \\ v = \left\lfloor \dfrac{\theta}{\Delta\theta} \right\rfloor \end{cases} \tag{6.2}$$

Parámetro	Valor
Campo de visión horizontal	360 grados
Campo de visión vertical	26.8 grados
Número de canales	64
Resolución horizontal del LiDAR ($\Delta\phi$)	0.35 grados/pixel
Resolución vertical del LiDAR ($\Delta\theta$)	0.42 grados/pixel
Ancho de la imagen obtenida del LiDAR	1024 píxeles
Alto de la imagen obtenida del LiDAR	64 píxeles

Tabla 6.1: Parámetros y valores empleados para la proyección esférica del LiDAR.

donde $\lfloor \cdot \rfloor$ denota un redondeo al entero más cercano, y $\Delta\phi$ y $\Delta\theta$ son las resoluciones angulares del sensor LiDAR en grados por píxel en los ejes horizontal y vertical, respectivamente. Estas resoluciones dependen de la configuración del sensor y determinan la densidad de puntos en la imagen panorámica resultante. La Tabla 6.1 muestra los parámetros y valores empleados para la proyección esférica del LiDAR, que se han seleccionado en función de las especificaciones del sensor Velodyne VLP-16 utilizado en la base de datos KITTI-360 [3]. Estos parámetros son cruciales para garantizar una proyección precisa y coherente de las lecturas LiDAR en la imagen panorámica.

En la obtención del mapa de profundidad, cada píxel (u, v) contiene el valor de profundidad r correspondiente (Figura 6.6 (b)). Sin embargo, cuando se trata del mapa de intensidad, cada píxel (u, v) toma el valor de la intensidad de la señal reflejada por el objeto más cercano al sensor LiDAR (Figura 6.5 (b)). Esta intensidad se calcula como la remisión del láser, que es proporcional a la cantidad de luz reflejada por los objetos en la escena. En este caso, cada píxel (u, v) toma el valor de intensidad correspondiente al punto más cercano en la nube de puntos de intensidad (Figura 6.5 (a)). Este proceso permite generar una imagen panorámica que representa la intensidad de la señal reflejada por los objetos en el entorno. En cuanto a la segmentación semántica de la nube de puntos LiDAR, primero se segmenta la misma por medio de MinkUNet34C [11], que clasifica cada punto de la nube entre 20 categorías semánticas posibles (Figura 6.7 (a)). Posteriormente, se proyecta la nube de puntos segmentada a una imagen panorámica, donde cada píxel (u, v) toma el color de la categoría semántica correspondiente al punto más cercano en la nube de puntos segmentada (Figura 6.7 (b)). Cabe destacar que las clases semánticas proporcionadas por MinkUNet no coinciden exactamente con las categorías empleadas por SegFormer [15]. Para construir un espacio común para el reconocimiento de lugares basado en información semántica, se unifican las categorías considerando únicamente las siguientes: coche, bicicleta, motocicleta, camión, persona, conductor, carretera, acera, terreno, edificio, muro/valla, vegetación, poste, señal de tráfico y objeto desconocido.

Por último, existen píxeles que no tienen correspondencia directa con las lecturas LiDAR y esto se debe a: (1) la baja resolución angular del sensor LiDAR, (2) puntos que no intersectan con ningún objeto (puntos en el infinito) y por el contrario, (3) puntos que no pueden ser capturados por estar el objeto demasiado cerca del sensor. Para

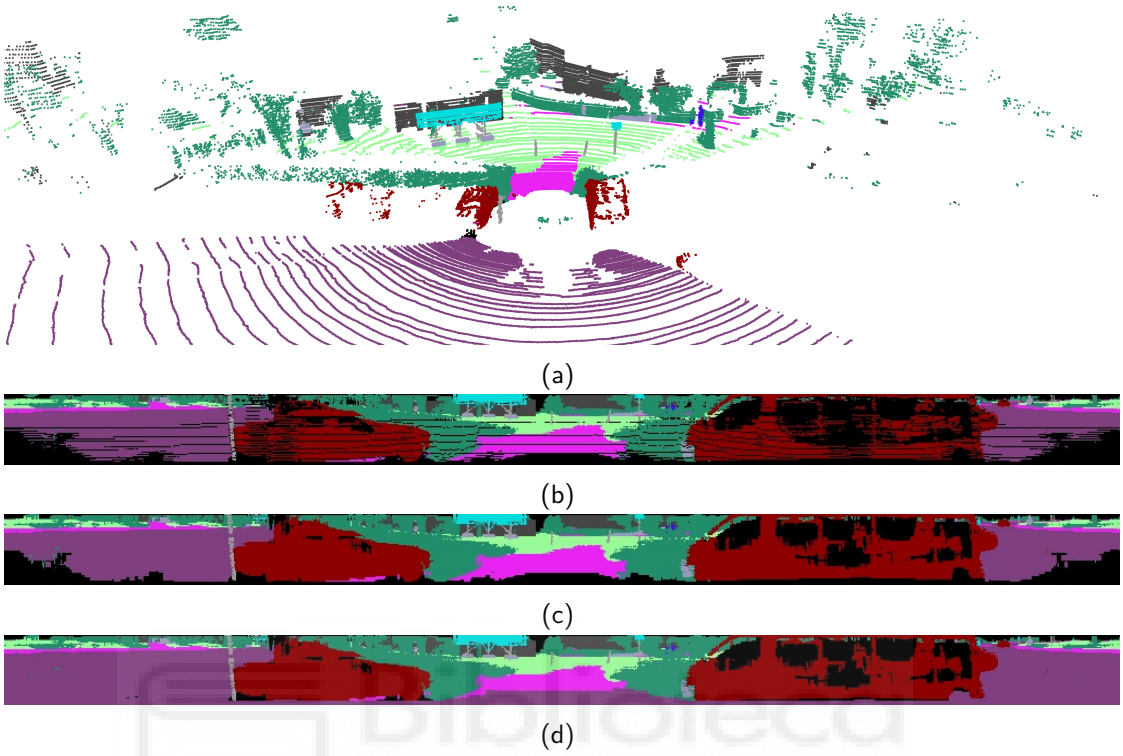


Figura 6.7: Proceso de transformación de nubes de puntos LiDAR a imágenes segmentadas semánticamente. (a) Nube de puntos LiDAR segmentada; (b) Imagen panorámica generada a partir de la nube de puntos segmentada; (c) Imagen panorámica segmentada e interpolada verticalmente para completar los píxeles faltantes; y (d) Imagen panorámica segmentada e interpolada verticalmente con filtrado del vehículo que transporta los sensores.

abordar el problema de los píxeles sin correspondencia debido a la resolución del sensor LiDAR, se aplica un algoritmo de interpolación vertical de manera que se obtienen las Figuras 6.5 (c), 6.6 (c) y 6.7 (c) a partir de las Figuras 6.5 (b), 6.6 (b) y 6.7 (b), respectivamente. Este algoritmo de interpolación vertical se basa en el hecho de que los puntos adyacentes verticales en las nubes de puntos suelen pertenecer al mismo objeto, por lo que se generan puntos interpolados considerando los píxeles adyacentes verticales para completar la imagen de intensidad, profundidad y semántica. En cuanto a los puntos que no intersectan con ningún elemento en la escena, se les asigna un valor de cero tanto en la imagen de intensidad, profundidad y semántica, lo que indica que no hay información disponible para esos píxeles. Por último, los puntos que no tienen correspondencia directa con las lecturas LiDAR debido a que están demasiado cerca del sensor se completan mediante un proceso de *inpainting* con LaMa (Figuras 6.5 (d) y 6.6 (d)) y en el caso de la imagen segmentada semánticamente, se eliminan los píxeles correspondientes al vehículo que transporta los sensores y se les asigna la clase "carretera" (Figura 6.7 (d)).

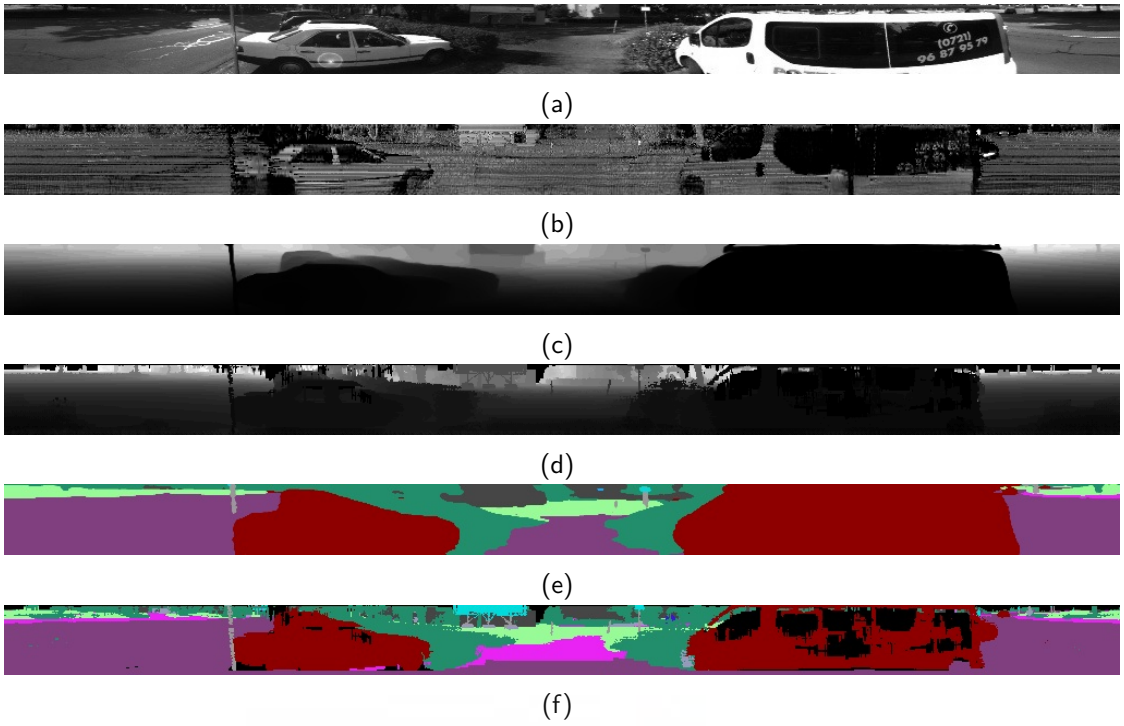


Figura 6.8: Comparación cualitativa de las imágenes generadas a partir de LiDAR y cámaras *fisheye* en el espacio común de la intensidad, profundidad e información semántica. (a) Imagen equirectangular de intensidad generada a partir de las imágenes *fisheye*, (b) Imagen de intensidad generada a partir de las lecturas del LiDAR, (c) Imagen equirectangular de profundidad generada a partir de las imágenes *fisheye*, (d) Imagen de profundidad generada a partir de las lecturas del LiDAR, (e) Imagen equirectangular segmentada semánticamente generada a partir de las imágenes *fisheye*, (f) Imagen segmentada semánticamente generada a partir de las lecturas del LiDAR.

6.3.3 Comparación cualitativa de las imágenes generadas a partir de LiDAR y cámaras *fisheye*

Para ilustrar la efectividad del proceso de transformación de las imágenes *fisheye* y las nubes de puntos LiDAR al espacio común de la intensidad, profundidad e información semántica, se presentan ejemplos cualitativos en la Figura 6.8. En esta figura se muestran las imágenes generadas a partir de las lecturas del LiDAR y las imágenes *fisheye* para cada uno de los espacios mencionados. En la primera y segunda fila se observan las imágenes de intensidad, donde se aprecia como tanto la imagen equirectangular generada a partir de las cámaras *fisheye* (Figura 6.8 (a)) como la imagen de intensidad obtenida a partir de la proyección esférica de las lecturas del LiDAR (Figura 6.8 (b)) ofrecen una representación coherente de la distribución de la intensidad en la escena. Sin embargo, existen diferencias inherentes debidas a la naturaleza de los sensores: mientras que las cámaras *fisheye* dependen de la iluminación ambiental y capturan la intensidad luminosa reflejada en el espectro visible, los sensores LiDAR son autoiluminados y miden la intensidad en función de la cantidad de energía láser reflejada por las superficies, lo que los hace menos sensibles a las condiciones de iluminación externas. Además, las cámaras *fisheye* presentan distorsiones ópticas que afectan a la

representación de la imagen, mientras que las imágenes generadas a partir del LiDAR son más precisas en términos de geometría y escala. En la tercera y cuarta fila se muestran las imágenes de profundidad, donde se observa una imagen de profundidad generada a partir de la imagen equirectangular anterior por medio de Depth Anything V2 Large [12] (Figura 6.8 (c)) y una imagen de profundidad generada a partir de la proyección esférica de las lecturas del LiDAR (Figura 6.8 (d)). Al igual que antes, las imágenes de profundidad generadas a partir de las imágenes *fisheye* presentan cierta distorsión y el grado de detalle en la estimación de profundidad por parte de Depth Anything V2 es inferior al obtenido por el LiDAR. Por último, en la quinta y sexta fila se presentan las imágenes de segmentación semántica, donde se puede observar la imagen equirectangular segmentada semánticamente por SegFormer [15] (Figura 6.8 (e)) y la nube de puntos segmentada semánticamente por MinkUNet34C [11] proyectada a imagen (Figura 6.8 (f)). Tal y como se indicó en el apartado anterior, se unifican las categorías semánticas predichas por cada uno de los modelos de segmentación en las siguientes clases: coche, bicicleta, motocicleta, camión, persona, conductor, carretera, acera, terreno, edificio, muro/valla, vegetación, poste, señal de tráfico y objeto desconocido.

Con estos ejemplos cualitativos, se demuestra que el proceso de transformación de las imágenes *fisheye* y las nubes de puntos LiDAR al espacio común de la intensidad, profundidad e información semántica permite obtener representaciones visualmente similares y comparables en un mismo espacio, independientemente del tipo de sensor de origen. Sin embargo, para que el reconocimiento de lugares sea efectivo, es necesario contar con un sistema capaz de embeber las imágenes de cada sensor en descriptores comunes a partir de estas representaciones unificadas. En este contexto, la arquitectura CosPlace [16] se presenta como una candidata idónea. A continuación se describe su adaptación para el reconocimiento *cross-modal* de lugares entre LiDAR y cámaras *fisheye*.

6.3.4 Arquitectura de red unificada para reconocimiento *cross-modal*

El método propuesto utiliza una única arquitectura para procesar los datos provenientes tanto del LiDAR como de las cámaras fisheye. Esta arquitectura está compuesta a su vez por tres ramas basadas en el modelo CosPlace [16], que ha demostrado ser efectivo en el reconocimiento de lugares a partir de imágenes estándar. Cada una de las tres ramas que conforman la arquitectura se encarga de procesar un tipo de imagen diferente: intensidad, profundidad y semántica, lo cual da lugar a un modelo CosPlace de intensidad, un modelo CosPlace de profundidad y un modelo CosPlace semántico. Estos modelos se entrenan de manera independiente según el tipo de imagen de entrada, pero se utiliza el mismo modelo tanto para LiDAR como para cámaras *fisheye*. De esta manera, se logra una representación coherente y útil para el reconocimiento *cross-modal* de lugares entre LiDAR y cámaras *fisheye*. Como se ha mencionado anteriormente, las imágenes de intensidad, profundidad y semántica se generan a partir de las lecturas del LiDAR y las imágenes *fisheye*. En el caso del LiDAR, la intensidad y la profundidad se obtienen directamente de las lecturas del sensor, mientras que la segmentación semántica se realiza mediante un modelo MinkUNet34C [11]. En el caso de las cámaras *fisheye*, las imágenes de intensidad se generan a partir de la conver-

sión a escala de grises de las imágenes equirectangulares, mientras que la profundidad y la segmentación semántica se obtienen mediante Depth Anything V2 Large [12] y SegFormer [15], respectivamente.

Los modelos CosPlace de intensidad, profundidad y semántica se adaptan en función de su información de entrada. De esta manera, cuando la imagen de entrada es de un solo canal (intensidad y profundidad), se modifica el número de canales de entrada de la primera capa convolucional del modelo CosPlace para que acepte imágenes de un único canal en lugar de tres canales (RGB). Además, se aprovecha el conocimiento previo del modelo original computando la media de los pesos de la primera capa convolucional original y transfiriéndolos a la nueva capa convolucional para que el modelo pueda procesar imágenes de profundidad e intensidad de manera efectiva desde el inicio del entrenamiento. Por otro lado, cuando la imagen de entrada es de tres canales (semántica), se mantiene la primera capa convolucional tal cual. De esta manera, se garantiza que el modelo pueda aprender representaciones discriminantes a partir de imágenes de intensidad, profundidad y semántica desde el inicio del entrenamiento en todas las capas del modelo. A lo largo del resto del capítulo, cada uno de estos tres tipos de información se denominará "fuente de unión", dado que es la información que se usará para obtener descriptores que puedan ser comparados en un mismo espacio de características independientemente del tipo de sensor (cámara o LiDAR) del que provengan.

Finalmente, los vectores descriptores extraídos por cada una de las ramas del modelo se fusionan por medio de una concatenación. Esta fusión tardía permite combinar las características aprendidas por cada rama del modelo, lo que mejora la capacidad del modelo para reconocer lugares a partir de diferentes tipos de imágenes. La arquitectura general del enfoque CrossPlace se muestra en la Figura 6.9. En esta figura se puede observar cómo las imágenes de intensidad, profundidad y semántica se procesan a través de sus respectivas ramas de la técnica CrossPlace, y cómo los descriptores resultantes se fusionan para obtener una representación final que se utiliza para el reconocimiento *cross-modal* de lugares.

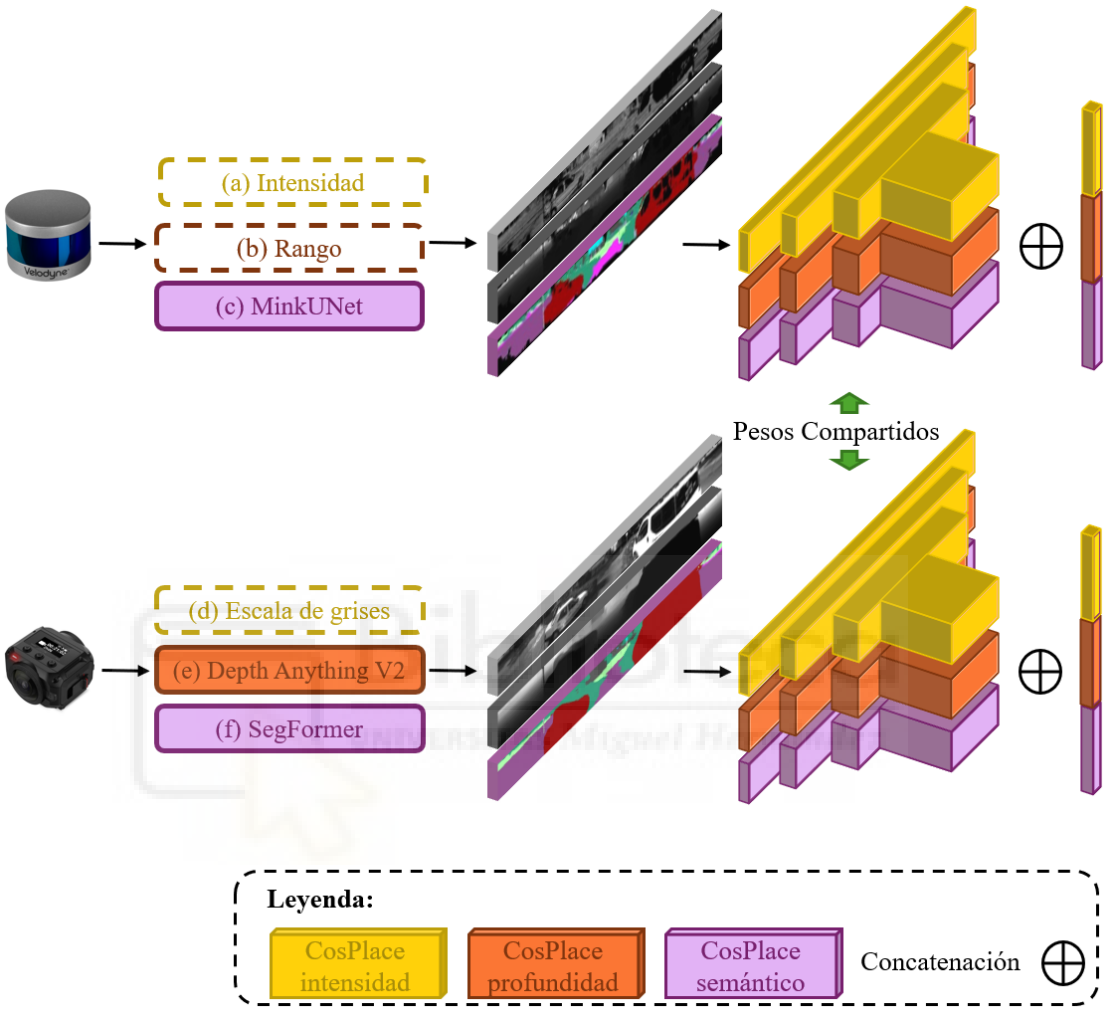


Figura 6.9: Arquitectura general del método CrossPlace, un enfoque unificado para el reconocimiento *cross-modal* de lugares entre LiDAR y cámaras omnidireccionales *fisheye* basado en el espacio común de la intensidad, profundidad e información semántica. La nube de puntos LiDAR se convierte en formato imagen por medio de una proyección esférica. En función de la información representada en cada pixel se obtiene: (a) la imagen de intensidad, (b) la imagen de rango y (c) la imagen segmentada por MinkUNet34C [11]. Por otro lado, las imágenes *fisheye* se transforman a un espacio equirectangular, donde se obtiene: (d) la imagen de intensidad mediante una conversión a escala de grises, (e) la imagen de profundidad estimada por Depth Anything V2 Large [12] y (f) la imagen semántica obtenida mediante SegFormer [15]. Cada una de estas fuentes de unión se emplea para entrenar una arquitectura independiente de CosPlace [16], pero con pesos compartidos entre modalidades de sensor. Posteriormente, los descriptores resultantes de cada fuente de unión se fusionan mediante una concatenación, lo que permite combinar las características aprendidas.

6.4 Experimentos

6.4.1 Conjuntos de datos

El método propuesto se evalúa mediante la base de datos KITTI-360 [3], que contiene información de diversos sensores y ha sido capturada en varias zonas de Karlsruhe (Alemania), incluyendo tanto zonas urbanas como autovías. Este conjunto de datos contiene más de 320000 imágenes y 100000 lecturas láser a lo largo de una distancia de 73.7 km. Entre los diferentes sensores de los que dispone el vehículo, en este capítulo únicamente se emplean los datos capturados por las dos cámaras *fisheye* y el sensor LiDAR Velodyne HDL-64E. Las cámaras *fisheye* cuentan con un campo de visión de 185º y fueron instaladas en ambos lados del vehículo, proporcionando una cobertura total de 360º del entorno. El sensor LiDAR, montado en el techo del vehículo, dispone de 64 canales de resolución y ofrece nubes de puntos que abarcan un rango de 120 metros en un campo de visión de 360º horizontal y 26.8º vertical. Además, dispone de información geolocalizada precisa, incluyendo coordenadas GPS y orientación del vehículo, lo que permite identificar las lecturas de diferentes sensores en un espacio común.

6.4.2 Entrenamiento y evaluación

Para llevar a cabo los experimentos, el conjunto de datos se ha dividido siguiendo el protocolo establecido por [278], separando las diferentes secuencias que conforman el conjunto de datos en zonas urbanas y zonas de autovía. Cabe destacar que existen secuencias puramente urbanas (*00*, *02*, *04*, *05*, *06*, *08*, *09*, *10* y *18*), otras en las que el vehículo ha recorrido únicamente zonas de autovía (*03*) y otras en las que el vehículo ha recorrido tanto zonas urbanas como de autovía (*07*). Para el entrenamiento, se han utilizado únicamente las zonas urbanas de las secuencias *00*, *02*, *04*, *05*, *06*, *07*, *08*, *09* y *10*. Este conjunto de entrenamiento presenta una gran variedad de entornos urbanos, incluyendo calles residenciales, zonas comerciales y áreas con alta densidad de tráfico. En total, se han generado 74232 pares de imágenes *fisheye* y nubes de puntos LiDAR para el entrenamiento del modelo.

Para la evaluación, se dispone de dos conjuntos de test que, a su vez, se dividen en entorno urbano y autovía. El primer conjunto de evaluación, denominado "Test", incluye una zona diferente a la de entrenamiento de la secuencia urbana *00* y la secuencia de autovía *03*, con un total de 1501 y 1010 pares de datos, respectivamente. El segundo conjunto de evaluación, denominado "Extra test", incluye la secuencia urbana *18* y la zona de autovía de la secuencia *07*, con un total de 3447 y 842 pares de datos, respectivamente. Ambos conjuntos de evaluación se emplean para analizar el rendimiento del modelo durante el análisis comparativo (Sección 6.4.5). Además, estos conjuntos se utilizan para comparar el método CrossPlace propuesto con el resto de estudios del estado del arte (Sección 6.4.6), permitiendo medir el rendimiento ante diferentes escenarios y condiciones. Cabe destacar que los datos de consulta (tanto en el entrenamiento como en el test) se seleccionan cada 3 metros (siguiendo los trabajos [274, 278]), mientras que las lecturas restantes conforman la base de datos. Esto garantiza unas condiciones de evaluación realistas, en las que las lecturas de consulta se

Conjunto	Tipo	Secuencia										
		00	02	03	04	05	06	07	08	09	10	18
Entren.	Consulta	2096	3349	-	2854	1382	2305	818	1592	3058	900	-
	Base de datos	6917	10320	-	8198	4909	6881	1230	5108	10189	2126	-
Test	Consulta	332	-	404	-	-	-	-	-	-	-	-
	Base de datos	1169	-	606	-	-	-	-	-	-	-	-
Extra test	Consulta	-	-	-	-	-	-	376	-	-	-	902
	Base de datos	-	-	-	-	-	-	466	-	-	-	2545

Tabla 6.2: Distribución de pares de datos por secuencia en el conjunto de datos KITTI-360.

toman en ubicaciones diferentes respecto a las lecturas de la base de datos. La Tabla 6.2 muestra la distribución de pares de datos por secuencia en el conjunto de datos KITTI-360, mientras que la Tabla 6.3 resume el número total de pares de datos en cada conjunto. En total, se dispone de 74232 pares de datos para entrenamiento y 6800 pares de datos para evaluación, lo que proporciona una base sólida para entrenar y evaluar el método propuesto.

En cuanto a las métricas de evaluación, se han utilizado de nuevo las métricas estándar en el reconocimiento de lugares: *recall at 1* (R@1), que mide la proporción de consultas para las cuales el elemento más cercano en el espacio del descriptor es un positivo verdadero si se encuentra dentro de un umbral de distancia de d metros. Este umbral de distancia d tomará un valor de 10 metros en la Sección 6.4.5 y de 20 metros en la Sección 6.4.6, siguiendo los trabajos [4, 58, 159]. En esta última sección, también se emplearán las variantes R@5 y R@20, que miden la proporción de consultas para las cuales al menos uno de los k elementos más cercanos en el espacio del descriptor es un positivo verdadero ($k = [1, 5, 20]$ en este capítulo). Estas métricas son fundamentales para evaluar la efectividad del modelo en el reconocimiento de lugares entre diferentes modalidades de sensor, dada la complejidad que supone este problema en el estado del arte actual.

Además, se realizan dos tipos de evaluaciones: (1) reconocimiento de lugares entre imágenes *fisheye* como consultas y nubes de puntos LiDAR proyectadas como base de datos (2D-3D) y (2) reconocimiento de lugares utilizando nubes de puntos LiDAR como consultas e imágenes *fisheye* como base de datos (3D-2D). Cabe destacar que el reconocimiento de lugares entre imágenes (2D-2D) y entre nubes de puntos (3D-3D) no se aborda en este capítulo, ya que estas modalidades se han tratado en los Capítulos 3 y 4, respectivamente. El capítulo actual se centra únicamente en el reconocimiento *cross-modal* de lugares entre imágenes *fisheye* y nubes de puntos LiDAR, lo que permite evaluar la capacidad del modelo para generalizar y reconocer lugares a partir de diferentes modalidades de sensor.

6.4.3 Etiquetado y similitud

Para el etiquetado de similitud entre pares de imágenes *fisheye* y nubes de puntos LiDAR, aprovechamos la información geolocalizada proporcionada por el conjunto de datos KITTI-360 [3], que está alineada con *OpenStreetMaps*. Siguiendo un enfoque

Conjunto	Entorno urbano			Entorno de autovía		
	Consulta	Base de datos	Total	Consulta	Base de datos	Total
Entrenamiento	18354	55878	74232	-	-	-
Test	332	1169	1501	404	606	1010
Extra test	902	2545	3447	376	466	842

Tabla 6.3: Resumen del conjunto de datos KITTI-360 utilizado en los experimentos.

similar al presentado en la Sección 4.4.2, definimos la similitud entre una lectura de consulta (ya provenga de las cámaras *fisheye* o del LiDAR) y un elemento de la base de datos basándonos en la distancia euclídea entre las posiciones GPS de captura.

En concreto, dos capturas (independientemente de la modalidad del sensor) se consideran positivas (estructuralmente similares) si fueron tomadas a una distancia euclídea menor a $p = 10$ metros entre sí. Por el contrario, se consideran negativas (estructuralmente diferentes) si la distancia entre ellas es mayor a $n = 50$ metros. De esta manera, dada una lectura de referencia, los ejemplos positivos y negativos se seleccionan con independencia de la modalidad del sensor, ya sea imagen *fisheye* o nube de puntos LiDAR. Este criterio permite crear pares de entrenamiento que fomentan un aprendizaje unificado en un sólo modelo de red para ambas modalidades, facilitando la codificación de ambos tipos de dato en un mismo espacio del descriptor y por tanto, posibilitando el reconocimiento *cross-modal* de lugares.

6.4.4 Detalles de implementación

Para el entrenamiento de CosPlace se vuelve a utilizar la función de pérdida *Truncated Smooth-AP* definida en la Sección 4.4.4. Para un rendimiento efectivo, esta función de pérdida necesita un tamaño de lote grande [227]. En este caso, se ha seleccionado un tamaño de lote de 1024 imágenes, lo que permite una mejor convergencia y estabilidad durante el entrenamiento. El modelo se entrena durante 4 épocas, utilizando el optimizador Adam con una tasa de aprendizaje inicial de 1×10^{-3}. Esta tasa de aprendizaje se reduce por un factor de 10 en las épocas 2 y 3, para una convergencia más fina hacia el final del entrenamiento. Los parámetros y valores utilizados para generar imágenes de intensidad, profundidad y semánticas para posteriormente entrenar el modelo se resume en la Tabla 6.4.

Todos los experimentos se llevan a cabo en una GPU NVIDIA GeForce RTX 3090 con 24 GB. Nuestro código está disponible públicamente en la página web del proyecto `https://juanjo-cabrera.github.io/projects-CrossPlace/`.

6.4.5 Análisis comparativo

En esta sección, se estudia tanto la elección del modelo a utilizar como la fuente de unión entre diferentes modalidades de sensor, el preprocesamiento de las imágenes y la fusión de características. En concreto, en la Sección 6.4.5.1 se evalúa el rendimiento del modelo CosPlace utilizando diferentes *backbones* y tamaños de descriptor, con el

Parámetro	Valor
Distancia positiva (p)	10 m
Distancia negativa (n)	50 m
Tamaño del lote	1024
Número de épocas	4
Tasa de aprendizaje inicial (LR)	1×10^{-3}
Épocas de reducción de LR	2, 3
Umbral de distancia (d)	10 m

Tabla 6.4: Parámetros y valores utilizados para entrenar el modelo CosPlace.

objetivo de determinar la mejor configuración para el reconocimiento *cross-modal* de lugares. En la Sección 6.4.5.2, se compara el rendimiento del método propuesto en función de la fuente de unión entre imágenes *fisheye* y nubes de puntos LiDAR, es decir, si se utilizan imágenes de profundidad, semánticas o intensidad separadamente. Seguidamente, en la Sección 6.4.5.3, se analiza el impacto del preprocesamiento de las imágenes de profundidad y semánticas en el rendimiento del modelo, evaluando cómo diferentes técnicas de preprocesamiento afectan a la calidad de las características extraídas y, por ende, al rendimiento del reconocimiento *cross-modal*. Por último, en la Sección 6.4.5.4 se estudia el impacto de la fusión temprana y tardía de las características extraídas de las imágenes de profundidad y semánticas, para determinar cuál de estas estrategias proporciona un mejor rendimiento en el reconocimiento *cross-modal* de lugares.

6.4.5.1 Estudio preliminar del modelo y tamaño del descriptor de CosPlace

En esta sección se estudia el rendimiento del modelo CosPlace en función del *backbone* de red utilizado y el tamaño del descriptor de salida del modelo. En concreto, se han probado las siguientes arquitecturas: VGG16 [5], ResNet-18 [6], ResNet-50 [6], ResNet-101 [6] y ResNet-152 [6], y los tamaños de descriptor de 32, 64, 128, 256, 512, 1024 y 2048. De manera preliminar, únicamente se emplea la información de intensidad para llevar a cabo el reconocimiento de lugares cruzado entre modalidades de sensor.

En la Tabla 6.5 se muestran los resultados de R@1 (%) para los diferentes modelos de red en los que se basa CosPlace para un tamaño de descriptor inicial de 512. Además, los resultados se diferencian entre entorno urbano y autovía, y entre modalidad 2D-3D y 3D-2D. En esta tabla se observa que VGG16 obtiene resultados modestos, especialmente bajos en autovía pues no superan el 30 % en ninguna de las modalidades. Este hecho evidencia la dificultad de este modelo para extraer características discriminantes en entornos repetitivos. Por su lado, ResNet-18 mejora notablemente el rendimiento de VGG16 alcanzando valores entorno al 90 % en entorno urbano y alrededor del 50-60 % en autovía, lo que indica una mayor capacidad de generalización. Al incrementar la profundidad de la red, ResNet-50 muestra mejoras adicionales obteniendo un 78.16 % en promedio, aunque ResNet-101 obtiene resultados ligeramente inferiores al obtener un 73.53 % en promedio frente al 76.67 % de ResNet-18. Finalmente, ResNet-152 obtiene los mejores resultados globales, superando el 95 % en urbano y el 60 % en autovía,

	Test				**Extra test**				
	Urbano (00)		Autovía (03)		Urbano (18)		Autovía (07)		
Modelo	2D-3D	3D-2D	2D-3D	3D-2D	2D-3D	3D-2D	2D-3D	3D-2D	**Total**
VGG16	72.59	78.98	18.92	29.98	57.11	54.82	25.18	21.84	44.93
ResNet-18	94.50	95.19	53.98	60.94	91.29	89.25	56.10	56.09	74.67
ResNet-50	96.60	96.10	69.53	64.24	88.78	93.21	60.18	56.68	78.16
ResNet-101	94.50	94.89	57.61	65.41	88.42	87.71	51.60	48.03	73.52
ResNet-152	**99.60**	**99.10**	**71.66**	**69.50**	**95.99**	**96.38**	**61.22**	**60.56**	**81.75**

Tabla 6.5: Evaluación de los diferentes *backbones* de CosPlace para un tamaño de descriptor de 512 con intensidad. Se muestran los resultados de R@1 (%) para una distancia $d = 10m$ en los conjuntos de test y extra test, diferenciando entre entorno urbano y autovía, y entre consultas 2D-3D y 3D-2D. La última columna muestra la media total de los 8 valores.

tanto en 2D-3D como en 3D-2D, y alcanzando una media total del 81.75 %. Estos resultados confirman que el uso de arquitecturas más profundas permite capturar mejor la complejidad de las representaciones de intensidad, especialmente en entornos desafiantes como las autovías (escenario no usado en el entrenamiento), donde la variabilidad visual es menor y la discriminación resulta más compleja. Por tanto, ResNet-152 se selecciona como *backbone* óptimo para el resto de experimentos.

Posteriormente, se estudia el impacto del tamaño del descriptor en el rendimiento del modelo CosPlace con ResNet-152 como *backbone* (Tabla 6.6). Se evalúan diferentes tamaños de descriptor: 32, 64, 128, 256, 512, 1024 y 2048. Esto permite analizar cómo afecta la dimensionalidad del descriptor a la capacidad de discriminación del modelo ante diferentes entornos y modalidades. Como se puede apreciar en la Tabla 6.6, los descriptores de tamaño intermedio (256, 512 y 1024) ofrecen los mejores resultados globales, superando en promedio el 80 %. En particular, el descriptor de 1024 componentes alcanza un 82.16 % siendo el valor más alto, seguido muy de cerca por el 81.75 % del descriptor de tamaño 512. Por el contrario, tamaños muy pequeños (64) o excesivamente grandes (2048) presentan una disminución en el rendimiento, probablemente debido a la pérdida de información discriminativa o a un sobreajuste a las condiciones de entrenamiento, respectivamente. En el caso del descriptor de 2048, el sobreajuste a entornos urbanos (entorno de entrenamiento) es evidente ya que se obtienen resultados muy altos en este tipo de entornos (98.99 %, 99.70 %, 96.53 % y 95.60 %) pero muy bajos en autovía (67.83 %, 64.00 %, 53.25 % y 50.28 %). Por tanto, se concluye que un tamaño de descriptor de 512 o 1024 componentes son los más adecuados para garantizar el rendimiento y robustez en el reconocimiento *cross-modal* de lugares, siendo 1024 el tamaño seleccionado para el resto de experimentos.

6.4.5.2 Intensidad, profundidad y semántica como fuentes de unión entre LiDAR y cámara

Anteriormente, en la Sección 6.4.5.1, se ha estudiado el rendimiento del modelo CosPlace utilizando únicamente la información de intensidad como fuente de unión entre las modalidades de cámara y LiDAR. Sin embargo, existen otras fuentes de unión

	Test				Extra test				
	Urbano (00)		Autovía (03)		Urbano (18)		Autovía (07)		
Tamaño	2D-3D	3D-2D	2D-3D	3D-2D	2D-3D	3D-2D	2D-3D	3D-2D	**Total**
32	96.59	96.10	63.14	62.40	87.58	91.34	52.63	53.13	75.36
64	90.94	91.85	51.21	48.92	85.94	88.60	38.71	48.65	68.10
128	98.71	96.99	69.53	64.24	95.33	94.18	62.25	54.21	79.43
256	98.99	98.80	74.65	72.26	91.63	93.53	61.64	56.27	80.85
512	**99.60**	99.10	71.66	69.50	95.99	96.38	61.22	60.56	81.75
1024	98.19	97.89	**75.25**	**72.03**	94.12	94.24	**63.30**	**62.23**	**82.16**
2048	98.99	**99.70**	67.83	64.00	**96.53**	**95.60**	53.25	50.28	78.27

Tabla 6.6: Evaluación de los diferentes tamaños de descriptor para CosPlace ResNet-152 con intensidad. Se muestran los resultados de R@1 (%) para una distancia $d = 10m$ en los conjuntos de test y extra test, diferenciando entre entorno urbano y autovía, y entre consultas 2D-3D y 3D-2D. La última columna muestra la media total de los 8 valores.

que pueden mejorar el rendimiento del reconocimiento *cross-modal* de lugares, como son la información de profundidad y semántica tal y como se ha descrito en los apartados anteriores. Por tanto, en esta sección se comparan estas tres fuentes de unión para determinar cuál es la más adecuada para unir las modalidades de cámara y LiDAR en la tarea de reconocimiento de lugares. Para ello, se ha utilizado el modelo CosPlace con ResNet-152 como *backbone* y un tamaño de descriptor de 1024 componentes. Se han evaluado separadamente las tres fuentes de unión: intensidad, profundidad y semántica, en términos de R@1 (%) para los conjuntos de test y extra test, diferenciando entre entorno urbano y autovía, y entre consultas 2D-3D y 3D-2D. Cabe destacar que la información de intensidad se obtiene directamente de la image equirectangular obtenida a partir de las cámaras *fisheye* y de la intensidad del LiDAR. Además, la información de profundidad también se obtiene directamente de la proyección del LiDAR, pero para la estimación de la profundidad en las imágenes equirectangulares, se hace uso Depth Anything V2 Large [12]. Por último, la información semántica se estima a partir de la segmentación semántica de las imágenes equirectangulares utilizando el modelo SegFormer [15] y de la segmentación semántica del LiDAR utilizando el modelo MinkUNet34C [11].

Los resultados mostrados en la Tabla 6.7 indican que la información de intensidad presenta el rendimiento más bajo en todos los casos, especialmente en autovía, donde no supera el 76 % en ninguna de las modalidades. Sin embargo, como se ha indicado anteriormente, es la única fuente de unión que no requiere de ningún modelo de aprendizaje previo, ya que se basa únicamente en la intensidad de las lecturas del LiDAR y las imágenes en escala de grises de las cámaras. En cuanto a la información semántica como fuente de unión, es la más robusta en entornos urbanos alcanzando un rendimiento cercano al 100 % en la mayoría de los casos. Sin embargo, en autovía, este tipo de información presenta un rendimiento inferior. En concreto, si se comparan los resultados de la segmentación semántica con los de profundidad en entornos de autovía, se aprecia que la profundidad supera a la semántica en todos los casos, siendo una fuente de unión más adecuada para el reconocimiento en entornos más

	Test				**Extra test**				
	Urbano (00)		Autovía (03)		Urbano (18)		Autovía (07)		
Fuente	2D-3D	3D-2D	2D-3D	3D-2D	2D-3D	3D-2D	2D-3D	3D-2D	**Total**
Intensidad	98.19	97.89	75.25	72.03	94.12	94.24	63.30	62.23	82.16
Profundidad	98.80	98.19	**87.62**	**88.61**	95.57	95.23	**82.18**	**80.32**	**90.82**
Semántica	**99.70**	**100.00**	78.22	85.40	**98.67**	**97.23**	76.06	72.12	88.68

Tabla 6.7: Evaluación de las diferentes fuentes de unión (profundidad, intensidad y semántica) entre LiDAR y cámaras *fisheye*. Se muestran los resultados de R@1 (%) para una distancia $d = 10m$ en los conjuntos de test y extra test, diferenciando entre entorno urbano y autovía, y entre consultas 2D-3D y 3D-2D. La última columna muestra la media total de los 8 valores.

homogéneos y repetitivos como las autovías, donde la geometría de la escena es crucial para la identificación de lugares. En general, la información de profundidad muestra un rendimiento superior al de intensidad y semántica en todos los casos, alcanzando una media total del 90.82 %, pero cuando se trata de entornos urbanos, la semántica supera a la profundidad.

6.4.5.3 Preprocesamiento de la intensidad, profundidad y semántica

En esta sección se estudia la influencia del preprocesamiento de las imágenes de intensidad, profundidad y semántica en el rendimiento del modelo CosPlace. El preprocesamiento es una etapa crucial en el *pipeline* de reconocimiento *cross-modal* de lugares, ya que permite asemejar aún más las dos modalidades de sensor. En concreto, se evalúan dos técnicas de preprocesamiento: (1) *inpainting* del vehículo que transporta los sensores tanto en las imágenes *fisheye* como LiDAR y (2) interpolación vertical de las imágenes de LiDAR.

En la Tabla 6.8 se muestran los resultados de R@1 (%) para diferentes técnicas de preprocesamiento aplicadas a las imágenes de intensidad del LiDAR y las imágenes equirectangulares en escala de grises. Para comenzar, se muestra el punto de partida con las imágenes de intensidad y escala de grises sin ningún tipo de preprocesamiento, obteniendo un rendimiento del 82.16 % en total. A continuación, se aplica la técnica de *inpainting* para la eliminación del vehículo que transporta los sensores a la imagen equirectangular en escala de grises, mientras que las imágenes provenientes del LiDAR se dejan en crudo. Así, se produce una ligera mejora a nivel global del 0.46 % (82.62 %). A partir de ahora, esta técnica de *inpainting* se aplicará a las imágenes de intensidad 2D. Posteriormente, se aplica la técnica de interpolación vertical sobre la imagen de intensidad del LiDAR, lo que mejora aún más el rendimiento global hasta un 85.03 %. Esta técnica es especialmente efectiva en las secuencias de autovía y en concreto, en la modalidad 3D-2D, donde la imagen de consulta proviene del LiDAR y las imágenes que conforman el mapa se capturan con las cámaras *fisheye*, alcanzando un 76.49 % y un 72.87 en las secuencias *03* y *07*, respectivamente. Finalmente, además de la interpolación vertical, se aplica un *inpainting* sobre la imagen de intensidad LiDAR para eliminar el vehículo que transporta los sensores, lo que empeora el rendimiento global a un 83.40 %. Por ello, de aquí en adelante, se utilizará únicamente la técnica

		Test				Extra test				
		Urbano (00)		Autovía (03)		Urbano (18)		Autovía (07)		
	Intensidad	2D-3D	3D-2D	2D-3D	3D-2D	2D-3D	3D-2D	2D-3D	3D-2D	**Total**
	En crudo	98.19	97.89	75.25	72.03	94.12	**94.24**	63.30	62.23	82.16
2D	+ *Inpainting* †	**98.49**	**98.49**	**77.23**	74.50	92.79	91.80	66.76	60.90	82.62
3D	+ Interpolación	98.19	97.89	76.98	**76.49**	**96.34**	92.57	68.88	**72.87**	**85.03**
	+ *Inpainting* ‡	97.89	98.19	73.02	72.28	94.12	90.24	**71.81**	69.68	83.40

† Eliminación del vehículo sobre el que van montados los sensores en la imagen equirectangular en escala de grises.
‡ Eliminación del vehículo sobre el que van montados los sensores en la imagen de intensidad del LiDAR.

Tabla 6.8: Influencia del preprocesamiento de imágenes en escala de grises (2D) e intensidad LiDAR (3D) en el rendimiento de CosPlace. Se muestran los resultados de R@1 (%) para una distancia $d = 10m$ en los conjuntos de test y extra test, diferenciando entre entorno urbano y autovía, y entre consultas 2D-3D y 3D-2D. La última columna muestra la media total de los 8 valores.

de interpolación vertical para las imágenes de intensidad del LiDAR y la técnica de *inpainting* para las imágenes equirectangulares en escala de grises.

En la Tabla 6.9 se aplican las técnicas de procesamiento sobre las imágenes de profundidad obtenidas mediante Depth Anything V2 y la proyección esférica del LiDAR. Para comenzar, se muestra el punto de partida sin ningún tipo de preprocesamiento, obteniendo un rendimiento promedio del 90.82 %. A continuación, se aplica la técnica de *inpainting* para la eliminación del vehículo que transporta los sensores a la imagen 2D equirectangular de profundidad, produciendo una mejoría a nivel global del 1.05 % (91.87 %). Esta mejora es especialmente notable en los entornos urbanos, donde se alcanza valores del 99.70 % y 100.00 % en la secuencia *00* y valores del 98.12 % y 98.34 % en la secuencia *18*, respectivamente. Posteriormente, se aplica sobre la imagen 2D con *inpainting* la potencia a la cuarta, lo que mejora aún más el rendimiento global hasta un 93.57 %. Esta técnica es especialmente efectiva en la secuencia de autovía *03*, donde se alcanza un 94.06 % en la modalidad 3D-2D. A partir de ahora, se utilizará la técnica de *inpainting* y la potencia a la cuarta para las imágenes de profundidad equirectangulares. En cuanto al procesamiento de la imagen 3D de rango LiDAR, se aplica la técnica de interpolación vertical, lo que mejora el rendimiento global hasta un 94.82 %. Por último, se aplica sobre la imagen 3D interpolada el *inpainting* para eliminar el vehículo que transporta los sensores, lo que al igual que en el caso de la intensidad, no termina de mejorar el rendimiento global. Por ello, se utilizará únicamente la técnica de interpolación vertical para las imágenes de rango LiDAR y la técnica de *inpainting* y potencia para las imágenes equirectangulares de profundidad.

En cuanto al procesamiento de las imágenes de segmentación semántica, la Tabla 6.10 muestra los resultados de R@1 (%) para diferentes técnicas de preprocesamiento aplicadas a las imágenes semánticas. En este caso, se comienza de nuevo con las imágenes semánticas sin ningún tipo de preprocesamiento, obteniendo un rendimiento

		Test				**Extra test**				
		Urbano (00)		Autovía (03)		Urbano (18)		Autovía (07)		
	Profundidad	2D-3D	3D-2D	2D-3D	3D-2D	2D-3D	3D-2D	2D-3D	3D-2D	**Total**
	En crudo	98.80	98.19	87.62	88.61	95.57	95.23	82.18	80.32	90.82
2D	+ *Inpainting* †	**99.70**	**100.00**	87.87	91.09	98.12	98.34	80.32	79.52	91.87
	+ Potencia	99.40	99.70	92.08	**94.06**	98.67	99.00	83.51	82.18	93.57
3D	+ Interpolación	**99.70**	99.40	**95.54**	92.08	**99.22**	**99.22**	**88.03**	**85.37**	**94.82**
	+ *Inpainting* ‡	99.40	99.70	92.08	93.32	98.89	99.11	85.11	82.45	93.76

† Eliminación del vehículo sobre el que van montados los sensores en la imagen equirectangular de profundidad.

‡ Eliminación del vehículo sobre el que van montados los sensores en la imagen de rango del LiDAR.

Tabla 6.9: Influencia del preprocesamiento de imágenes de profundidad (2D) y rango LiDAR (3D) en el rendimiento de CosPlace. Se muestran los resultados de R@1 (%) para una distancia $d = 10m$ en los conjuntos de test y extra test, diferenciando entre entorno urbano y autovía, y entre consultas 2D-3D y 3D-2D. La última columna muestra la media total de los 8 valores.

del 88.68 % en total. A continuación, se aplica la técnica de *inpainting* sobre la imagen equirectangular segmentada, eliminando el vehículo que transporta los sensores, lo que produce una mejora marginal del 0.01 % (88.69 %). Posteriormente, se aplica la técnica de interpolación vertical a las imágenes de LiDAR segmentadas, lo que sí mejora el rendimiento global hasta un 89.59 %. Esta técnica es especialmente efectiva en la secuencia de autovía *07*, donde se alcanza un 80.05 % en la modalidad 2D-3D y un 76.06 % en la modalidad 3D-2D. Por último, se aplica el *inpainting* a la imagen de LiDAR segmentada, lo que termina de mejorar el rendimiento global hasta un 90.25 %. Por este motivo, de aquí en adelante se utilizará la técnica de interpolación vertical e *inpainting* para las imágenes de LiDAR segmentadas y la técnica de *inpainting* para las imágenes equirectangulares segmentadas.

		Test				Extra test				
		Urbano (00)		Autovía (03)		Urbano (18)		Autovía (07)		
	Semántica	2D-3D	3D-2D	2D-3D	3D-2D	2D-3D	3D-2D	2D-3D	3D-2D	**Total**
	En crudo	99.70	**100.00**	78.22	85.40	98.67	97.23	76.06	72.12	88.68
2D	+ *Inpainting* †	**100.00**	**100.00**	80.20	84.16	97.01	97.12	78.19	72.87	88.69
3D	+ Interpolación	98.49	99.70	80.20	85.45	**99.00**	97.78	**80.05**	**76.06**	89.59
	+ *Inpainting* ‡	98.80	**100.00**	**83.17**	**87.87**	98.34	**98.23**	79.79	75.80	**90.25**

† Eliminación del vehículo sobre el que van montados los sensores en la imagen equirectangular de semántica.
‡ Eliminación del vehículo sobre el que van montados los sensores en la imagen semántica del LiDAR.

Tabla 6.10: Influencia del preprocesamiento de imágenes segmentadas (2D) y LiDAR segmentado (3D) en el rendimiento de CosPlace. Se muestran los resultados de R@1 (%) para una distancia $d = 10m$ en los conjuntos de test y extra test, diferenciando entre entorno urbano y autovía, y entre consultas 2D-3D y 3D-2D. La última columna muestra la media total de los 8 valores.

6.4.5.4 Fusión temprana *vs.* fusión tardía

En esta sección se estudia el impacto de la fusión temprana y tardía en el rendimiento del modelo CosPlace para el reconocimiento cruzado de lugares entre cámaras *fisheye* y LiDAR. La fusión temprana implica combinar las imágenes de intensidad, profundidad y segmentación semántica antes de ser procesadas por el modelo CosPlace, mientras que la fusión tardía combina las características extraídas por la arquitectura después de procesar cada imagen por separado. Por su parte, la fusión tardía se plantea de dos maneras: (1) mediante la adición o (2) mediante la concatenación de los descriptores de intensidad, profundidad y semánticos.

En la Tabla 6.11 se muestran los resultados de R@1 (%) para las diferentes fuentes de unión (intensidad, profundidad y semántica) después del procesamiento, tanto de manera individual como combinada, utilizando CosPlace-152 con un tamaño de descriptor de 1024 componentes. Se evalúan las modalidades 2D-3D y 3D-2D en los conjuntos de test y extra test, diferenciando entre entorno urbano y autovía. Como se ha visto anteriormente, de manera individual, el R@1(%) con la intensidad toma un valor promedio del 85.03 %, con la profundidad alcanza un rendimiento del 94.82 %, mientras que con la semántica presenta un rendimiento del 88.68 %. En cuanto a la fusión temprana, se observa que el rendimiento global empeora tomando un valor de 92.94 %, superando tanto a la semántica como a intensidad, pero no a la profundidad de manera individual. Sin embargo, la fusión tardía presenta mejores resultados que los individuales. En concreto, la adición de los descriptores de intensidad, profundidad y semánticos mejora el rendimiento global hasta un 96.96 %. Por otro lado, la concatenación de los descriptores de intensidad, profundidad y semánticos alcanza el mejor rendimiento global con un 97.45 %, superando a todas las demás modalidades y fuentes de unión. En particular, destaca especialmente en las secuencias de autovía. Por tanto, se concluye que la fusión tardía mediante concatenación de información de intensidad, profundidad y semántica es la más adecuada para el reconocimiento cruzado de lugares

	Test				**Extra test**				
	Urbano (00)		Autovía (03)		Urbano (18)		Autovía (07)		
Fuente	2D-3D	3D-2D	2D-3D	3D-2D	2D-3D	3D-2D	2D-3D	3D-2D	**Total**
Intensidad	98.19	97.89	76.98	76.49	96.34	92.57	68.88	72.87	85.03
Profundidad	99.70	99.40	95.54	92.08	99.22	99.22	88.03	85.37	94.82
Semántica	98.80	**100.00**	83.17	87.87	98.34	98.23	79.79	75.80	90.25
Fusión temprana	99.70	99.70	90.59	93.81	99.11	98.78	83.51	78.19	92.94
Fusión tardía †	**100.00**	**100.00**	95.30	96.53	**99.89**	99.67	93.88	90.43	96.96
Fusión tardía ‡	**100.00**	**100.00**	**96.29**	**98.02**	99.78	**99.89**	**94.41**	**91.22**	**97.45**

† Adición de los descriptores de intensidad, profundidad y semánticos.
‡ Concatenación de los descriptores de intensidad, profundidad y semánticos.

Tabla 6.11: Evaluación de las diferentes técnicas de fusión de las fuentes de unión intensidad, profundidad y semántica. Se muestran los resultados de R@1 (%) para una distancia $d = 10m$ en los conjuntos de test y extra test, diferenciando entre entorno urbano y autovía, y entre consultas 2D-3D y 3D-2D. La última columna muestra la media total de los 8 valores.

entre cámaras *fisheye* y LiDAR, ya que permite aprovechar al máximo los beneficios de todas las fuentes de información y mejorar el rendimiento global del modelo. Con ello, se define CrossPlace como método para el reconocimiento cruzado de lugares entre cámaras *fisheye* y LiDAR, utilizando la fusión tardía mediante concatenación de descriptores de intensidad, profundidad y semántica.

6.4.6 Comparación con el estado del arte

En esta sección se realiza una comparación del rendimiento de CrossPlace con aquellos trabajos del estado del arte que utilizan la base de datos KITTI-360 [3] para llevar a cabo el reconocimiento de lugares entre modalidades de sensores diferentes (Cámara-LiDAR). En concreto, se comparan los resultados obtenidos con los de AE-Spherical [274], LIP-Loc [161] y SaliencyI2PLoc [278]. En general, resulta difícil comparar los resultados con todos los trabajos del estado del arte, pero los autores de SaliencyI2PLoc [278] trataron de unificar las diferentes soluciones a la hora de utilizar el KITTI-360, definiendo un nuevo *benchmark* de comparación. Para ello, entrenaron los modelos LIP-Loc [161] y AE-Spherical [274] siguiendo la división que establecieron en su propio trabajo, y los evaluaron en igualdad de condiciones. Por este motivo, el método propuesto en este capítulo se ha entrenado y evaluado siguiendo la división de la base de datos KITTI-360 propuesta por Li *et al.* [278], que está descrita en la Sección 6.4.1. Cabe destacar que estos trabajos utilizan una distancia d de 20 metros para evaluar el rendimiento de sus modelos, aunque en la Sección 6.4.5 se ha definido una distancia d de 10 metros para la evaluación del método CrossPlace para evitar saturar los resultados y poder extraer conclusiones más significativas. Además, estos trabajos consideran únicamente el reconocimiento de imágenes capturadas por las cámaras en mapas conformados por nubes de puntos del sensor LiDAR, es decir, únicamente plantean la modalidad 2D-3D. A continuación, en la Tabla 6.12 se presentan los resultados de CrossPlace utilizando una distancia d de 10 y 20 metros para poder compararlos con los del estado del arte y seguir la traza de los experimentos realizados anteriormente.

En la Tabla 6.13 se muestran los resultados de R@1 (%), R@5 (%) y R@20 (%)

	Test				Extra test				
	Urbano (00)		Autovía (03)		Urbano (18)		Autovía (07)		
Distancia d	2D-3D	3D-2D	2D-3D	3D-2D	2D-3D	3D-2D	2D-3D	3D-2D	**Total**
CrossPlace 10 m	100.00	100.00	96.29	98.02	99.78	99.89	94.41	91.22	97.45
CrossPlace 20 m	100.00	100.00	99.50	99.50	100.00	100.00	98.67	96.54	99.28

Tabla 6.12: Resultados de CrossPlace en términos de R@1 (%) para una distancia $d = 10m$ y $d = 20m$. Se muestran los resultados para las modalidades 2D-3D y 3D-2D en los conjuntos de test y extra test, diferenciando entre entorno urbano y autovía. La última columna muestra la media total de los 8 valores.

	Test						Extra test						
	Urbano (00)			Autovía (03)			Urbano (18)			Autovía (07)			**Total**
Método	R@1	R@5	R@20	R@1	R@5	R@20	R@1	R@5	R@20	R@1	R@5	R@20	**R@1**
AE-Spherical [274]	41.57	60.54	79.52	10.64	22.28	45.30	37.14	60.42	82.59	4.79	20.74	43.09	23.54
LIP-Loc [161]	64.46	79.52	91.27	37.87	58.42	79.46	-	-	-	-	-	-	-
SaliencyI2PLoc [278]	78.92	86.75	97.59	30.94	49.26	75.99	61.86	81.15	95.90	22.34	50.27	72.07	48.52
CrossPlace	**100.00**	**100.00**	**100.00**	**99.50**	**99.75**	**100.00**	**100.00**	**100.00**	**100.00**	**98.67**	**100.00**	**100.00**	**99.54**

Tabla 6.13: Resultados de CrossPlace en comparación con el estado del arte. Se muestran los resultados de R@1 (%), R@5 (%) y R@20 (%) para una distancia $d = 20m$ en los conjuntos de test y extra test, diferenciando entre entorno urbano y autovía en la modalidad 2D-3D. La última columna muestra la media total de los resultados de R@1 (%).

de CrossPlace y los trabajos del estado del arte en la modalidad 2D-3D, diferenciando entre entorno urbano y autovía. En general, CrossPlace supera a todos los trabajos del estado del arte en todas las métricas y secuencias, alcanzando un rendimiento global de 99.54 % en R@1. En concreto, destaca especialmente en las secuencias de autovía, donde propuestas como AE-Spherical [274], LIP-Loc [161] y SaliencyI2PLoc [278] obtienen un rendimiento muy bajo, con valores de R@1 del 10.64 %, 37.87 % y 30.94 %, respectivamente en la secuencia *03*. Sin embargo, CrossPlace alcanza un 99.50 % y 98.67 % en las secuencias de autovía *03* y *07*, respectivamente. En el entorno urbano, CrossPlace también supera a los trabajos del estado del arte, saturando los resultados a un 100.00 % en R@1 % en las secuencias *00* y *18*. En general, el método propuesto en este capítulo supera a los trabajos del estado del arte en todas las secuencias y métricas, alcanzando un rendimiento global de 99.54 % en R@1. Por tanto, se concluye que CrossPlace es el método más efectivo para el reconocimiento cruzado de lugares entre cámaras *fisheye* y LiDAR en la base de datos KITTI-360.

6.5 Resultados cualitativos de la tarea de reconocimiento de lugares

En esta sección se presentan ejemplos visuales de los resultados obtenidos por la técnica CrossPlace propuesta en el presente capítulo para desempeñar la tarea de reconocimiento de lugares entre imágenes *fisheye* y LiDAR. Estos ejemplos ilustran la capacidad del método en entornos urbanos (Figuras 6.10, 6.11 y 6.12) y entornos de autovía (Figuras 6.13, 6.14 y 6.15) no usados en el entrenamiento. En cada figura, se muestra un ejemplo capturado en una de las diferentes secuencias del conjunto de datos, donde se observa la imagen de intensidad, profundidad y semántica de LiDAR

o cámara *fisheye* y la predicción de la imagen de intensidad, profundidad y semántica más cercana en el espacio del descriptor de la base de datos, la cual está conformada por el sensor contrario al empleado en test. Además se comprueba si coincide con la imagen de intensidad, profundidad y semántica más cercana en el espacio métrico de la posición. Las posiciones del mapa se representan con puntos azules, la posición actual con una cruz roja, la posición predicha con un círculo amarillo y la posición real con un anillo verde.

Los ejemplos de las Figuras 6.10, 6.11 y 6.12 muestran el rendimiento del método en el entorno urbano dado por las secuencias *00* y *18*. En la Figura 6.10, se observa que el método logra una predicción correcta en la modalidad 2D-3D, donde las segmentaciones semánticas obtenidas a partir del LiDAR y cámaras difieren relativamente. En la Figura 6.11, se muestra un ligero error en la modalidad 3D-2D, donde la nube de puntos LiDAR está ligeramente desplazada respecto a la imagen. En la Figura 6.12, la predicción es correcta en la modalidad 2D-3D, aunque se observa que el método tiene más facilidad para asociar la instancia de test con una del mapa que haya sido capturada con la misma orientación, aunque la instancia más cercana del mapa fuera capturada en sentido contrario.

En las Figuras 6.13, 6.14 y 6.15, se presentan ejemplos del entorno de autovía, dado por las secuencias *03* y *07*. En la Figura 6.13, se observa que el método logra una predicción correcta en la modalidad 2D-3D, aún cuando la segmentación semántica obtenida a partir del LiDAR y cámaras difiere en la categorización de la acera. En la Figura 6.14, se muestra un error en la modalidad 3D-2D, donde en la imagen de test hay presencia de un vehículo que viene en sentido contrario y el método la asocia a otra instancia en la que también hay un vehículo, pero en este caso, no se trata del mismo coche. Además, en este tipo de entornos tan repetitivos, resulta ligeramente difícil para el método asociar correctamente las instancias de test con las del mapa, ya que la mayoría de las instancias del mapa son muy similares entre sí. Sin embargo, de manera general el método funciona bien en este tipo de entornos, tal y como se muestra en la Figura 6.15 donde se observa que el método logra una predicción correcta en la modalidad 3D-2D, en una zona donde el *visual aliasing* es muy alto, ya que las instancias del mapa son muy similares entre sí. Para ver más ejemplos, visite la página del proyecto `https://juanjo-cabrera.github.io/projects-CrossPlace/`.

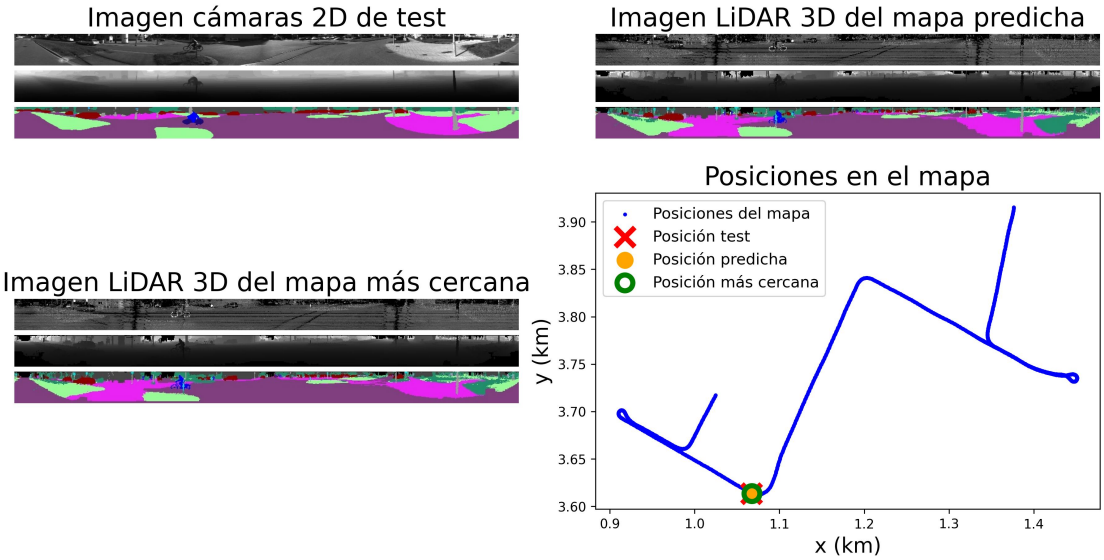


Figura 6.10: Ejemplo de acierto en la modalidad 2D-3D en el entorno *00* con el método CrossPlace en la tarea de reconocimiento de lugares entre imágenes *fisheye* y LiDAR en entornos urbanos.

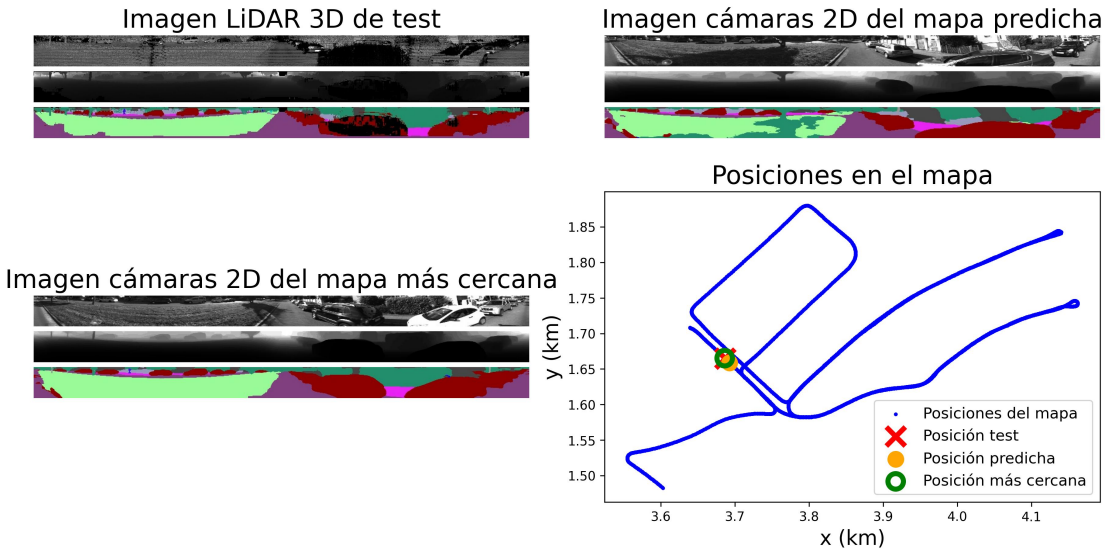


Figura 6.11: Ejemplo de ligero error en la modalidad 3D-2D en el entorno *18* con el método CrossPlace en la tarea de reconocimiento de lugares entre imágenes *fisheye* y LiDAR en entornos urbanos.

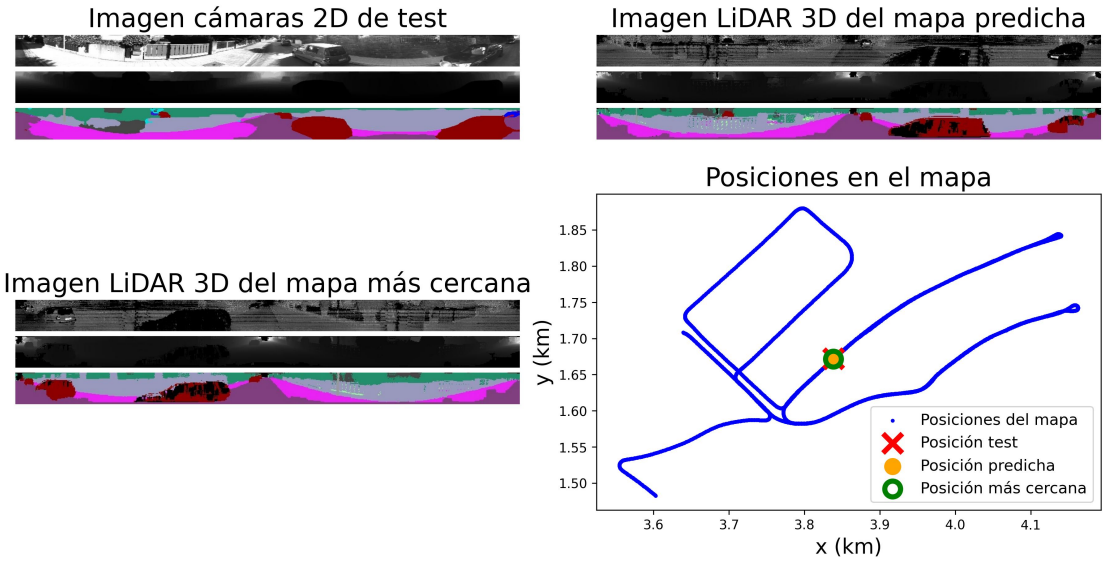


Figura 6.12: Ejemplo de acierto en la modalidad 2D-3D en el entorno *18* con el método CrossPlace en la tarea de reconocimiento de lugares entre imágenes *fisheye* y LiDAR en entornos urbanos.

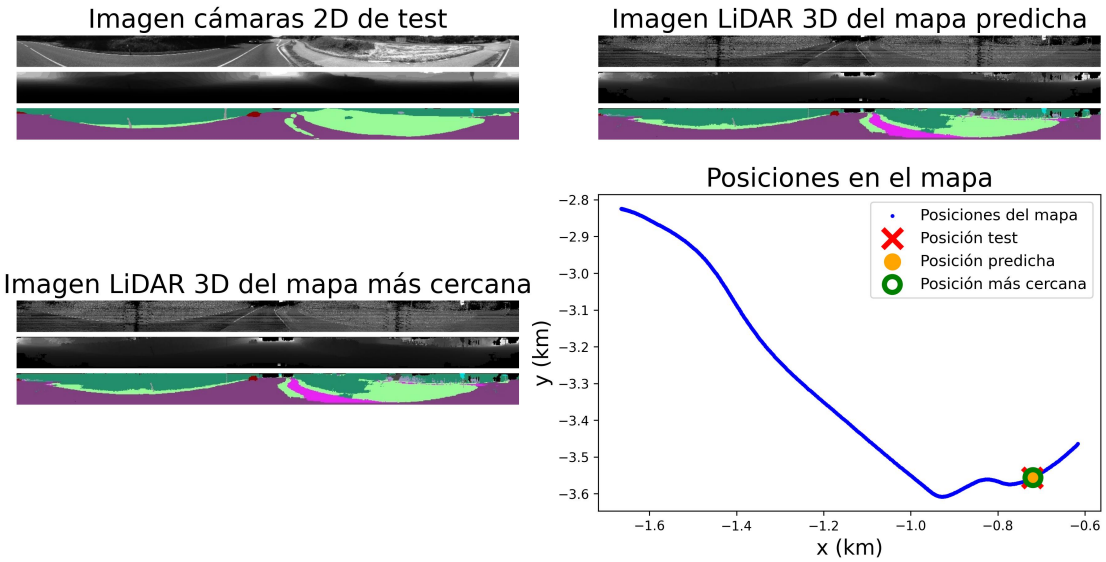


Figura 6.13: Ejemplo de acierto en la modalidad 2D-3D en el entorno *07* con el método CrossPlace en la tarea de reconocimiento de lugares entre imágenes *fisheye* y LiDAR en entornos de autovía.

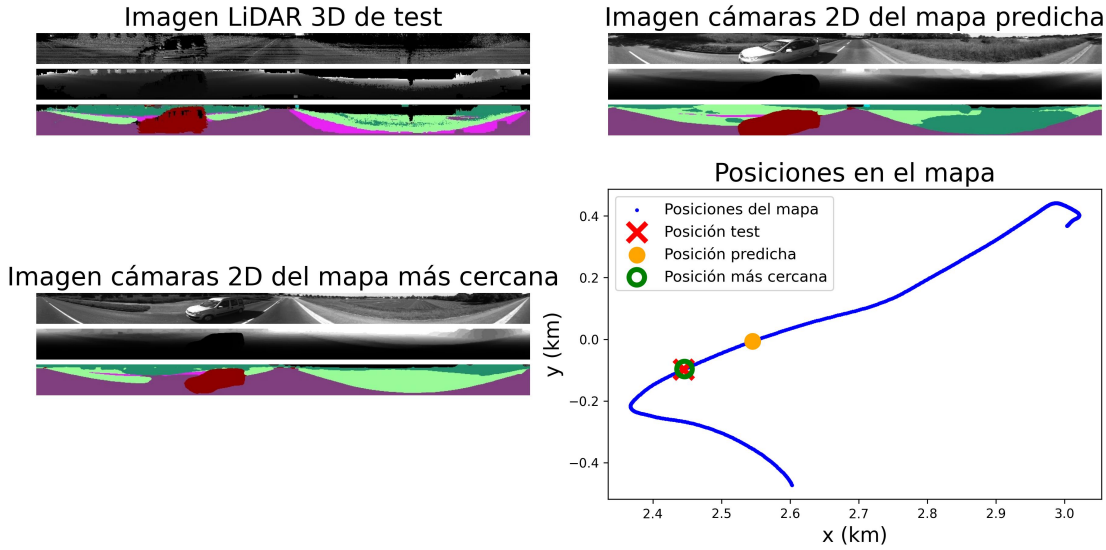


Figura 6.14: Ejemplo de error en la modalidad 3D-2D en el entorno *03* con el método CrossPlace en la tarea de reconocimiento de lugares entre imágenes *fisheye* y LiDAR en entornos de autovía.

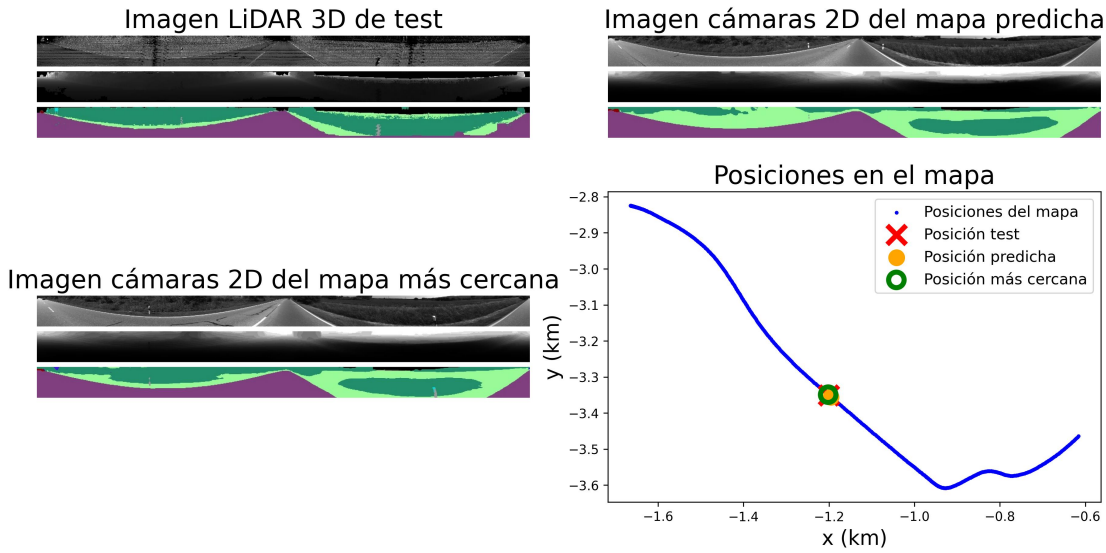


Figura 6.15: Ejemplo de acierto en la modalidad 3D-2D en el entorno *07* con el método CrossPlace en la tarea de reconocimiento de lugares entre imágenes *fisheye* y LiDAR en entornos de autovía.

6.6 Conclusiones

En este capítulo se ha presentado un método innovador para el reconocimiento de lugares entre modalidades de sensores heterogéneos, específicamente entre cámaras *fisheye* y LiDAR. El método propuesto, denominado CrossPlace, transforma las lecturas de ambos sensores al espacio común de intensidad, profundidad y semántica, permitiendo el uso de una única arquitectura de red para ambas modalidades. Este enfoque elimina la necesidad de utilizar el mismo tipo de sensor para capturar la base de datos y, posteriormente, realizar la consulta y tarea de reconocimiento, lo que resulta en una solución más flexible y práctica para sistemas multi-robot o plataformas con configuraciones sensoriales diversas.

Los resultados experimentales en la base de datos KITTI-360 han demostrado la efectividad de CrossPlace, superando a los métodos del estado del arte en todas las métricas en escenarios urbanos y de autovía. En particular, la integración de información de profundidad y semántica ha mostrado ser clave para mejorar la discriminación en entornos homogéneos y repetitivos, como las autovías, mientras que la información de intensidad ha demostrado ser una solución eficiente y sin dependencia de modelos de aprendizaje previos.

Además, se ha evaluado el impacto de diferentes técnicas de preprocesamiento, como la interpolación vertical y el *inpainting*, así como estrategias de fusión temprana y tardía de características. Los resultados indican que la fusión tardía mediante concatenación de descriptores de intensidad, profundidad y semántica proporciona el mejor rendimiento global, destacando la importancia de combinar múltiples fuentes de información para el reconocimiento *cross-modal* de lugares.

En conclusión, este capítulo introduce una solución robusta y eficiente para el reconocimiento de lugares entre modalidades de sensores diferentes, abriendo nuevas posibilidades para aplicaciones en robótica móvil y sistemas autónomos. Como trabajo futuro, se plantea explorar la integración de otras modalidades sensoriales, como sensores térmicos, así como la extensión del enfoque a escenarios más complejos y dinámicos.

7 Conclusiones y trabajos futuros

En los anteriores capítulos se han presentado diversas contribuciones al campo del reconocimiento de lugares utilizando diferentes modalidades sensoriales. Se han realizado propuestas novedosas de arquitecturas de red que han resultado en métodos de reconocimiento robustos ante variaciones ambientales. En este capítulo final, se resumen las principales conclusiones y se proponen líneas de investigación futuras que podrían ampliar y mejorar los enfoques desarrollados.

7.1 Contribuciones y conclusiones

En esta tesis se ha abordado el reconocimiento de lugares desde una perspectiva integral, desarrollando métodos que aprovechan diferentes modalidades de sensores y técnicas de aprendizaje profundo para proporcionar soluciones robustas, eficientes y prácticas. Los avances presentados en los capítulos anteriores representan contribuciones significativas al estado del arte en robótica móvil y navegación autónoma. A continuación, se enumeran los hitos y las contribuciones fundamentales que se han alcanzado en cada uno de los capítulos de esta tesis.

Capítulo 3

En este capítulo se aborda el reconocimiento visual de lugares mediante el desarrollo de dos enfoques complementarios: un método jerárquico que combina la clasificación de estancias con la estimación fina de la posición, y un método global basado en redes neuronales siamesas con aprendizaje por contraste. Los resultados experimentales demuestran que el enfoque jerárquico logra alta precisión en la clasificación de estancias y realiza una estimación efectiva de la posición dentro de cada habitación. Por su

parte, el método global también presenta un rendimiento competitivo, destacando su capacidad para generalizar a nuevos entornos y condiciones de iluminación.

Una contribución fundamental es el desarrollo de técnicas de aumento de datos específicas para imágenes omnidireccionales. Las técnicas propuestas, particularmente la modificación del contraste, así como la alteración del brillo/oscuridad general y la adición de focos de luz y sombras, son especialmente beneficiosas para mejorar la robustez ante condiciones cambiantes del entorno. Estos efectos visuales han sido diseñados para simular fenómenos que ocurren en condiciones reales de operación. La evaluación exhaustiva en el conjunto de datos COLD bajo diferentes condiciones de iluminación (nublado, soleado y noche) confirma la efectividad de estos enfoques.

Los métodos descritos en este capítulo se han publicado en:

- *J.J. Cabrera, O. J. Céspedes, S. Cebollada, O. Reinoso, L. Payá.* ***An evaluation of CNN models and data augmentation techniques in hierarchical localization of mobile robots****. Evolving Systems (2024). Ed. Springer-Verlag. ISSN: 1868-6486. DOI: https://doi.org/10.1007/s12530-024-09604-6* ***(Q3, JCR)***
- *J.J. Cabrera, V. Román, A. Gil, O. Reinoso, L. Payá.* ***An experimental evaluation of Siamese Neural Networks for robot localization using omnidirectional imaging in indoor environments****. Artificial Intelligence Review (2024). Ed. Springer. ISSN: 1573-7462. DOI: https://doi.org/ 10.1007/s10462-024-10840-0* ***(Q1, JCR)***
- *O. J. Céspedes, S. Cebollada, J.J. Cabrera, O. Reinoso, L. Payá.* ***Analysis of Data Augmentation Techniques for Mobile Robots Localization by Means of Convolutional Neural Network****. Artificial Intelligence, Applications and Innovations (2023). Ed. Springer. ISBN: 978-3-031-34110-6. ISSN: 1868-4238.*

Capítulo 4

En este capítulo se propone MinkUNeXt, una nueva arquitectura de red neuronal basada en convoluciones 3D dispersas que establece un nuevo hito en el estado del arte para el reconocimiento de lugares basado en LiDAR. La contribución principal consiste en la propuesta de esta arquitectura, cuyo diseño parte de MinkUNet [11], una red de segmentación semántica que se modifica drásticamente para obtener una arquitectura optimizada específicamente para el reconocimiento de lugares. Este proceso incluye el desarrollo del bloque residual MinkNeXt 3D, que incorpora elementos de diseño actuales como cuellos de botella invertidos, activaciones GeLU y normalización por capas, adaptados al dominio 3D.

Los resultados experimentales validan cada decisión de diseño mediante un análisis exhaustivo que justifica tanto la configuración macro (uso de arquitectura U-Net con tres conexiones de salto) como el diseño micro (configuración específica del bloque residual). Testada en los datasets Oxford RobotCar e In-house, MinkUNeXt logró un R@1 de 98.3 % y R@1 % de 99.5 %, mejorando significativamente con respecto a métodos previos.

Capítulo 5

En este capítulo se plantea una solución novedosa que combina las ventajas económicas de las cámaras con la robustez geométrica de la información tridimensional. La contribución principal consiste en validar la viabilidad del reconocimiento de lugares utilizando nubes de puntos 3D sintéticas generadas a partir de imágenes omnidireccionales mediante estimadores de profundidad de última generación.

El enfoque propuesto, pL-MinkUNeXt, parte de la generación de nubes de puntos 3D sintéticas a partir de imágenes panorámicas, con el objetivo principal de dotar al reconocimiento de lugares de una mayor invariancia frente a cambios de iluminación. Para ello, se emplea Distill Any Depth [13] para transformar las imágenes en mapas de profundidad, que posteriormente se convierten en nubes de puntos pseudo-LiDAR procesadas por la arquitectura MinkUNeXt. Una contribución clave de este trabajo es la propuesta de la técnica de aumento de datos *Distilled Depth Variations*, que consiste en utilizar diferentes variantes destiladas de modelos de estimación de profundidad, incluyendo estimadores menos robustos. Esta estrategia permite simular las inexactitudes presentes en las predicciones de profundidad y, de este modo, mejorar la resiliencia del modelo de reconocimiento de lugares MinkUNeXt frente a los errores inherentes en la generación de nubes de puntos pseudo-LiDAR. Los resultados experimentales presentan un R@1 promedio de 88.30 % y R@1 % de 87.31 % en el conjunto de datos COLD, superando consistentemente a métodos del estado del arte y mostrando una especial robustez en condiciones de iluminación desafiantes.

Capítulo 6

En este último capítulo se aborda uno de los desafíos más relevantes dentro de la robótica móvil: el reconocimiento de lugares entre diferentes modalidades de sensor. Para ello, se propone CrossPlace, un método de reconocimiento de lugares que integra información de LiDAR y sistemas de visión omnidireccionales con lentes *fisheye*. La principal contribución es la combinación de intensidad, profundidad y semántica como fuentes de unión entre las dos modalidades. Cada fuente de unión se procesa mediante un modelo CosPlace [16] especializado, que aprende representaciones compartidas entre modalidades LiDAR y cámara. Este enfoque evita la necesidad de múltiples modelos especializados para cada modalidad, permitiendo llevar efectivamente al mismo espacio de características la información LiDAR y de imagen. Los descriptores obtenidos por cada fuente de unión (intensidad, profundidad y semántica) se fusionan tardíamente mediante la concatenación, lo que permite una integración efectiva de la información.

La evaluación en el conjunto de datos KITTI-360 muestra la efectividad del enfoque propuesto, alcanzando un rendimiento global del 99.54 % en R@1 para la modalidad 2D-3D con una distancia *threshold* de 20 metros, superando significativamente a los otros métodos del estado del arte. Además, también se muestran resultados para distancias más estrictas, como 10 metros, y para ambas modalidades 2D-3D y 3D-2D donde CrossPlace aún logra un R@1 promedio de 97.45 %. Estos resultados demuestran la efectividad del método propuesto para el reconocimiento de lugares entre diferentes modalidades de sensor, tanto en entornos urbanos como de autovía.

7.2 Trabajos futuros

Durante la presente tesis, se han abordado diferentes líneas de investigación, pero no todas se han podido explorar en profundidad debido a limitaciones de tiempo. Sin embargo, el doctorando cuenta aún con unos meses de contrato predoctoral y con una anualidad adicional de Periodo de Orientación Postdoctoral (POP) en su contrato FPU. Aquellas líneas que se iniciaron y que se pretenden retomar en el futuro inmediato son las siguientes:

- Reconocimiento de lugares multimodal con fusión de imágenes y LiDAR: se inició un estudio preliminar en la mejora de la fusión de datos de imágenes y LiDAR para el reconocimiento de lugares y se plantea estudiarlo en profundidad si se le concede al doctorando la ayuda de movilidad asociada a su contrato FPU en la Polytechnique Montréal, donde se encuentra el profesor Pierre-Yves Lajoie. En concreto, se pretende explorar el uso de técnicas de atención y Transformers para la extracción de características en imagen para después fusionarlas con las características de LiDAR, mejorando así la robustez y precisión del reconocimiento de lugares en entornos complejos.

- Además, el doctorando inició una nueva línea de investigación durante su estancia en la Universidad de Oxford, junto con Daniele De Martini y Benjamin Ramtoula donde se propuso avanzar en el campo de la interpretabilidad de los modelos de percepción como Transformers Visuales (ViTs). En concreto, se analiza el DNA (*Distributions of Neuron Activations*) [283] del modelo tras procesar una imagen para identificar qué neuronas son más relevantes en función del tipo de objeto que se encuentra en dicha imagen. La posibilidad de identificar qué neuronas son más relevantes para cada tipo de objeto y/o elemento abre nuevas vías para modificar las activaciones de las neuronas y así alterar el comportamiento del modelo. Esto permitiría, por ejemplo, ignorar objetos no deseados que en el caso del reconocimiento de lugares se trata de aquellos elementos dinámicos que aparecen y desaparecen de la escena. Esta línea de investigación se pretende retomar en el futuro inmediato, ya que se considera de gran interés para la comunidad científica y puede aportar una nueva perspectiva sobre cómo los modelos de percepción aprenden y procesan la información visual.

Por otro lado, los avances presentados en esta tesis abren múltiples líneas de investigación futuras que pueden ampliar y mejorar las capacidades de los sistemas de reconocimiento de lugares. Una línea natural de investigación futura consiste en extender los enfoques desarrollados a otras modalidades de sensores. Los métodos unimodales y *cross-modal* presentados pueden adaptarse para incluir sensores radar, que han demostrado efectividad en condiciones climáticas adversas, y cámaras térmicas, que proporcionan información complementaria especialmente valiosa en condiciones de baja visibilidad. Por otro lado, una limitación actual de los métodos propuestos es su evaluación principalmente en entornos relativamente estáticos. Un posible trabajo futuro debería abordar el reconocimiento de lugares en entornos altamente dinámicos, donde elementos móviles como vehículos, peatones y cambios estructurales temporales pueden afectar significativamente el rendimiento. Esto requiere el desarrollo de técnicas similares a las comentadas anteriormente de manera que el modelo pueda aprender

a ignorar o adaptarse a estos elementos dinámicos, posiblemente mediante el uso de técnicas de segmentación semántica para identificar y filtrar objetos no relevantes, o la modificación directa de las activaciones del modelo. Además, los sistemas robóticos que operan durante períodos prolongados se enfrentan al desafío de adaptarse a cambios graduales en el entorno (construcciones y cambios estacionales extremos). El desarrollo de técnicas de aprendizaje continuo que permitan actualizar los modelos sin olvidar el conocimiento previo representa una línea de investigación crucial para aplicaciones prácticas a largo plazo.

Finalmente, la validación de los métodos propuestos en aplicaciones específicas como sistemas de vigilancia en entornos reales, proporcionaría evidencia adicional de su utilidad práctica y ayudaría a identificar requisitos específicos de cada dominio de aplicación. Esto conllevaría la implementación de los modelos y la adaptación de las arquitecturas para aprovechar los aceleradores *hardware* incorporados en robots como la NVIDIA Jetson Nano, así como la optimización del consumo energético y la latencia de inferencia.

7 Conclusions and Future Work

In the previous chapters, various contributions to the field of place recognition using different sensory modalities have been presented. Novel network architectures have been proposed, resulting in place recognition methods that are robust against environmental variations. In this final chapter, the main conclusions are summarized and future research directions are proposed that could expand and improve the developed approaches.

7.1 Contributions and Conclusions

This thesis addresses place recognition from a comprehensive perspective, developing methods that leverage different sensor modalities and deep learning techniques to provide robust, efficient and practical solutions. The advances presented in the previous chapters represent significant contributions to the state of the art in mobile robotics and autonomous navigation. Below, the main contributions achieved in each chapter of this thesis are listed.

Chapter 3

This chapter addresses visual place recognition through the development of two complementary approaches: a hierarchical method that combines room classification with fine position estimation and a global method based on Siamese neural networks with contrastive learning. Experimental results show that the hierarchical approach achieves high accuracy in room classification and provides effective position estimation within each room. The global method also achieves competitive performance, highlighting its ability to generalize to new environments and lighting conditions.

A fundamental contribution is the development of data augmentation techniques specific to omnidirectional images. The proposed techniques, particularly the contrast variation effect, as well as changes in global brightness/darkness and the addition of light spots and shadows, are especially beneficial for improving robustness against changing environmental conditions. These visual effects are designed to simulate phenomena that occur under real operation conditions. Exhaustive evaluation on the COLD dataset under different lighting conditions (cloudy, sunny and night) confirms the effectiveness of these approaches.

The methods described in this chapter have been published in:

- *J.J. Cabrera, O. J. Céspedes, S. Cebollada, O. Reinoso, L. Payá.* ***An evaluation of CNN models and data augmentation techniques in hierarchical localization of mobile robots****. Evolving Systems (2024). Springer-Verlag. ISSN: 1868-6486. DOI: https://doi.org/10.1007/s12530-024-09604-6* ***(Q3, JCR)***
- *J.J. Cabrera, V. Román, A. Gil, O. Reinoso, L. Payá.* ***An experimental evaluation of Siamese Neural Networks for robot localization using omnidirectional imaging in indoor environments****. Artificial Intelligence Review (2024). Springer. ISSN: 1573-7462. DOI: https://doi.org/ 10.1007/s10462-024-10840-0* ***(Q1, JCR)***
- *O. J. Céspedes, S. Cebollada, J.J. Cabrera, O. Reinoso, L. Payá.* ***Analysis of Data Augmentation Techniques for Mobile Robots Localization by Means of Convolutional Neural Network****. Artificial Intelligence, Applications and Innovations (2023). Springer. ISBN: 978-3-031-34110-6. ISSN: 1868-4238.*

Chapter 4

This chapter proposes MinkUNeXt, a new neural network architecture based on sparse 3D convolutions that sets a new milestone in the state of the art for LiDAR-based place recognition. The main contribution is the proposal of this architecture, whose design is based on MinkUNet [11], a semantic segmentation network that is drastically modified to obtain an architecture specifically optimized for place recognition. This process includes the development of the MinkNeXt 3D residual block, which incorporates state-of-the-art design elements such as inverted bottlenecks, GeLU activations and Layer Normalizations, adapted to the 3D domain.

Experimental results validate each design decision through an exhaustive analysis that justifies both the macro configuration (use of a U-Net architecture with three skip connections) and the micro design (specific configuration of the residual block). Tested on the Oxford RobotCar and In-house datasets, MinkUNeXt achieved a R@1 of 98.3% and R@1% of 99.5%, significantly improving upon previous methods.

Chapter 5

This chapter presents a novel solution that combines the economic advantages of cameras with the geometric robustness of three-dimensional information. The main contribution is validating the feasibility of place recognition using synthetic 3D point clouds generated from omnidirectional images using state-of-the-art depth estimators.

The proposed approach, pL-MinkUNeXt, is based on generating synthetic 3D point clouds from panoramic images, with the main objective of improving place recognition with greater invariance to lighting changes. To this end, Distill Any Depth [13] is used to transform the images into depth maps, which are subsequently converted into pseudo-LiDAR point clouds processed by the MinkUNeXt architecture. A key contribution of this work is the proposal of the Distilled Depth Variations data augmentation technique, which consists in using different distilled variants of depth estimation models, including less robust estimators. This strategy allows simulating the inaccuracies present in depth predictions and thus, improving the resilience of the MinkUNeXt place recognition model to the inherent errors in the generated pseudo-LiDAR point clouds. Experimental results show an average R@1 of 88.30% and R@1% of 87.31% on the COLD dataset, consistently outperforming state-of-the-art methods and showing particular robustness under challenging lighting conditions.

Chapter 6

This final chapter addresses one of the most relevant challenges in mobile robotics: cross-modal place recognition. To this end, CrossPlace is proposed, a place recognition method that integrates information from LiDAR and omnidirectional vision systems with fisheye lenses. The main contribution is the combination of intensity, depth and semantics as union sources between the two modalities. Each union source is processed by a specialized CosPlace model [16], which learns shared representations between LiDAR and camera modalities. This approach avoids the need for multiple specialized models for each modality, effectively bringing LiDAR and image information into the same feature space. The descriptors obtained from each union source (intensity, depth and semantics) are fused by a concatenation, resulting in a highly effective integration of the information.

The experiments carried out on the KITTI-360 dataset demonstrate the effectiveness of the proposed approach, achieving an overall performance of 99.54% in R@1 for the 2D-3D modality with a threshold distance of 20 meters, significantly outperforming other state-of-the-art methods. In addition, results are also shown for stricter distances, such as 10 meters, and for both 2D-3D and 3D-2D modalities, where CrossPlace still achieves an average R@1 of 97.45%. These results demonstrate the effectiveness of the proposed method for place recognition between different sensor modalities, both in urban and highway environments.

7.2 Future work

Throughout this thesis, different lines of research have been addressed, but not all of them could be explored in depth due to time constraints. However, the PhD candidate still has a few months left on the predoctoral contract and an additional year of the Postdoctoral Orientation Period (POP) in the FPU contract. The lines of research that were started and are intended to be resumed in the immediate future are as follows:

- Multimodal place recognition with image and LiDAR fusion: a preliminary study was started to improve the fusion of image and LiDAR data for place recognition

and it is planned to be studied in depth if the PhD candidate is granted the mobility grant associated with the FPU contract at Polytechnique Montréal, where Professor Pierre-Yves Lajoie is based. Specifically, the aim is to explore the use of attention mechanisms and Transformers for feature extraction in images and then fuse them with LiDAR features, thus improving the robustness and accuracy of place recognition in complex environments.

- In addition, the PhD candidate started a new line of research during a stay at the University of Oxford, together with Daniele De Martini and Benjamin Ramtoula, where the goal was to advance in the field of interpretability of perception models such as Visual Transformers (ViTs). Specifically, the DNA (Distribution of Neuron Activations) [283] of the model is analyzed after processing an image to identify which neurons are most relevant depending on the type of object present in that image. The ability to identify which neurons are most relevant for each type of object and/or element opens new avenues for modifying neuron activations and thus steering the model's behavior. This would allow, for example, ignoring unwanted objects, which in the case of place recognition refers to those dynamic elements that appear and disappear from the scene. This research line is intended to be resumed in the immediate future, as it is considered of great interest to the scientific community and may provide a new perspective on how perception models learn and process visual information.

Furthermore, the advances presented in this thesis open up multiple future research lines that can expand and improve the capabilities of place recognition systems. A natural future research direction is to extend the developed approaches to other sensor modalities. The unimodal and cross-modal methods presented can be adapted to include radar sensors, which have proven effective in adverse weather conditions, and thermal cameras, which provide especially valuable complementary information in low-visibility conditions. Also, a current limitation of the proposed methods is their evaluation mainly in relatively static environments. A possible future work should address place recognition in highly dynamic environments, where moving elements such as vehicles, pedestrians and temporary structural changes can significantly affect the performance. This requires the development of techniques similar to those previously discussed, so that the model can learn to ignore or adapt to these dynamic elements, possibly through the use of semantic segmentation techniques to identify and filter irrelevant objects, or by directly modifying the model's activations. In addition, robotic systems operating over long periods face the challenge of adapting to gradual changes in the environment (construction of buildings and extreme seasonal changes). The development of continuous learning techniques that allow models to be updated without forgetting previous knowledge represents a crucial line of research for long-term practical applications.

Finally, validating the proposed methods in specific applications such as surveillance systems in real environments would provide additional evidence of their practical utility and help identify specific requirements for each application domain. This would involve implementing the models and adapting the architectures to take advantage of hardware accelerators built into robots such as the NVIDIA Jetson Nano, as well as optimizing energy consumption and inference latency.

[1] A. Pronobis y B. Caputo, «COLD: COsy Localization Database», *The International Journal of Robotics Research (IJRR)*, vol. 28, n.º 5, págs. 588-594, 2009. DOI: 10.1177/0278364909103912.

[2] W. Maddern, G. Pascoe, C. Linegar y P. Newman, «1 year, 1000 km: The Oxford robotcar dataset», *The International Journal of Robotics Research*, vol. 36, n.º 1, págs. 3-15, 2017.

[3] Y. Liao, J. Xie y A. Geiger, «Kitti-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d», *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, n.º 3, págs. 3292-3310, 2022.

[4] M. A. Uy y G. H. Lee, «PointNetVLAD: Deep point cloud based retrieval for large-scale place recognition», en *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, págs. 4470-4479.

[5] K. Simonyan y A. Zisserman, «Very deep convolutional networks for large-scale image recognition», *arXiv preprint arXiv:1409.1556*, 2014.

[6] K. He, X. Zhang, S. Ren y J. Sun, «Deep residual learning for image recognition», en *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, págs. 770-778.

[7] S. Xie, R. Girshick, P. Dollár, Z. Tu y K. He, «Aggregated residual transformations for deep neural networks», en *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, págs. 1492-1500.

[8] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan y col., «Searching for mobilenetv3», en *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, págs. 1314-1324.

[9] M. Tan y Q. Le, «Efficientnetv2: Smaller models and faster training», en *International conference on machine learning*, PMLR, 2021, págs. 10 096-10 106.

[10] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell y S. Xie, «A convnet for the 2020s», en *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, págs. 11 976-11 986.

[11] C. Choy, J. Gwak y S. Savarese, «4D Spatio-Temporal ConvNets: Minkowski Convolutional Neural Networks», en *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, págs. 3075-3084.

[12] L. Yang, B. Kang, Z. Huang, Z. Zhao, X. Xu, J. Feng y H. Zhao, «Depth Anything v2», *Advances in Neural Information Processing Systems*, vol. 37, págs. 21 875-21 911, 2025. DOI: 10.48550/arXiv.2406.09414.

[13] X. He, D. Guo, H. Li, R. Li, Y. Cui y C. Zhang, «Distill Any Depth: Distillation Creates a Stronger Monocular Depth Estimator», *arXiv preprint arXiv:2502.19204*, 2025.

[14] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby y col., «DINOv2: Learning robust visual features without supervision», *arXiv preprint arXiv:2304.07193*, 2023. DOI: 10.48550/arXiv.2304.07193.

[15] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez y P. Luo, «SegFormer: Simple and efficient design for semantic segmentation with transformers», *Advances in neural information processing systems*, vol. 34, págs. 12 077-12 090, 2021.

[16] G. Berton, C. Masone y B. Caputo, «Rethinking Visual Geo-Localization for Large-Scale Applications», en *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, págs. 4878-4888. DOI: 10.48550/arXiv.2204.02287.

[17] R. Suvorov, E. Logacheva, A. Mashikhin, A. Remizova, A. Ashukha, A. Silvestrov, N. Kong, H. Goka, K. Park y V. Lempitsky, «Resolution-robust Large Mask Inpainting with Fourier Convolutions», *arXiv preprint arXiv:2109.07161*, 2021. DOI: 10.48550/arXiv.2109.07161.

[18] E. Rublee, V. Rabaud, K. Konolige y G. Bradski, «ORB: An efficient alternative to SIFT or SURF», en *2011 International conference on computer vision*, Ieee, 2011, págs. 2564-2571.

[19] M. A. Fischler y R. C. Bolles, «Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography», *Communications of the ACM*, vol. 24, n.º 6, págs. 381-395, 1981.

[20] I. Lluvia, E. Lazkano y A. Ansuategi, «Active mapping and robot exploration: A survey», *Sensors*, vol. 21, n.º 7, pág. 2445, 2021.

[21] I. Ullah, D. Adhikari, H. Khan, M. S. Anwar, S. Ahmad y X. Bai, «Mobile robot localization: Current challenges and future prospective», *Computer Science Review*, vol. 53, pág. 100 651, 2024.

[22] Á. Madridano, A. Al-Kaff, D. Martín y A. De La Escalera, «Trajectory planning for multi-robot systems: Methods and applications», *Expert Systems with Applications*, vol. 173, pág. 114 660, 2021.

[23] X. Cao, L. Ren y C. Sun, «Research on obstacle detection and avoidance of autonomous underwater vehicle based on forward-looking sonar», *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, n.º 11, págs. 9198-9208, 2022.

[24] S. Stavridis, P. Falco y Z. Doulgeri, «Pick-and-place in dynamic environments with a mobile dual-arm robot equipped with distributed distance sensors», en *2020 IEEE-RAS 20th International Conference on Humanoid Robots (Humanoids)*, IEEE, 2021, págs. 76-82.

[25] C. Masone y B. Caputo, «A survey on deep visual place recognition», *IEEE Access*, vol. 9, págs. 19 516-19 547, 2021.

[26] Z. Zhou, L. Li, A. Fürsterling, H. J. Durocher, J. Mouridsen y X. Zhang, «Learning-based object detection and localization for a mobile robot manipulator in SME production», *Robotics and Computer-Integrated Manufacturing*, vol. 73, pág. 102 229, 2022.

[27] O. H. Jafari, D. Mitzel y B. Leibe, «Real-time RGB-D based people detection and tracking for mobile robots and head-worn cameras», en *2014 IEEE international conference on robotics and automation (ICRA)*, IEEE, 2014, págs. 5636-5643.

[28] S. Badrloo, M. Varshosaz, S. Pirasteh y J. Li, «Image-based obstacle detection methods for the safe navigation of unmanned vehicles: A review», *Remote Sensing*, vol. 14, n.º 15, pág. 3824, 2022.

[29] Z. Chen, A. Jacobson, N. Sünderhauf, B. Upcroft, L. Liu, C. Shen, I. Reid y M. Milford, «Deep learning features at scale for visual place recognition», en *2017 IEEE international conference on robotics and automation (ICRA)*, IEEE, 2017, págs. 3223-3230.

[30] K. Li, Y. Ma, X. Wang, L. Ji y N. Geng, «Evaluation of Global Descriptor Methods for Appearance-Based Visual Place Recognition», *Journal of Robotics*, vol. 2023, n.º 1, pág. 9 150 357, 2023.

[31] Y. Yan, H. Zhang, C. Zhao, X. Liu y S. Fu, «LiDAR-based place recognition for mobile robots in ground/water surface multiple scenes», *Journal of Field Robotics*,

[32] M. Gadd, D. De Martini y P. Newman, «Contrastive learning for unsupervised radar place recognition», en *2021 20th International Conference on Advanced Robotics (ICAR)*, IEEE, 2021, págs. 344-349.

[33] W. N. Street e Y. Kim, «A streaming ensemble algorithm (SEA) for large-scale classification», en *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, 2001, págs. 377-382.

[34] H. Bay, T. Tuytelaars y L. Van Gool, «Surf: Speeded up robust features», en *Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part I 9*, Springer, 2006, págs. 404-417.

[35] D. Nistér, «An efficient solution to the five-point relative pose problem», *IEEE transactions on pattern analysis and machine intelligence*, vol. 26, n.º 6, págs. 756-770, 2004.

[36] X. Tang, W. Fu, M. Jiang, G. Peng, Z. Wu, Y. Yue y D. Wang, «Place Recognition Using Line-Junction-Lines in Urban Environments», en *2019 IEEE International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM)*, 2019, págs. 530-535. DOI: 10.1109/CIS-RAM47153.2019.9095776.

[37] S. Arshad y G.-W. Kim, «A Robust Feature Matching Strategy for Fast and Effective Visual Place Recognition in Challenging Environmental Conditions», *International Journal of Control, Automation and Systems*, vol. 21, n.º 3, págs. 948-962, 2023.

[38] M. Cummins y P. Newman, «FAB-MAP: Probabilistic localization and mapping in the space of appearance», *The International journal of robotics research*, vol. 27, n.º 6, págs. 647-665, 2008.

[39] A. Oliva y A. Torralba, «Building the gist of ascene: the role of global image features in recognition.», en *Progress in Brain Reasearch: Special Issue on Visual Perception.Vol. 155.*, 2006.

[40] N. Dalal y B. Triggs, «Histograms of Oriented Gradients for Human Detection.», en *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, San Diego, USA. Vol. II, pp. 886-893*, 2005.

[41] L. Payá, F. Amorós, L. Fernández y O. Reinoso, «Performance of global-appearance descriptors in map building and localization using omnidirectional vision», *Sensors*, vol. 14, n.º 2, págs. 3033-3064, 2014.

[42] K. M. Yi, E. Trulls, V. Lepetit y P. Fua, «Lift: Learned invariant feature transform», en *Computer Vision–ECCV 2016: 14th European Conference, Amster-*

dam, The Netherlands, October 11-14, 2016, Proceedings, Part VI 14, Springer, 2016, págs. 467-483.

[43] H. Noh, A. Araujo, J. Sim, T. Weyand y B. Han, «Large-scale image retrieval with attentive deep local features», en *Proceedings of the IEEE international conference on computer vision*, 2017, págs. 3456-3465.

[44] D. DeTone, T. Malisiewicz y A. Rabinovich, «Superpoint: Self-supervised interest point detection and description», en *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2018, págs. 224-236.

[45] M. Dusmanu, I. Rocco, T. Pajdla, M. Pollefeys, J. Sivic, A. Torii y T. Sattler, «D2-net: A trainable cnn for joint detection and description of local features», *arXiv preprint arXiv:1905.03561*, 2019.

[46] P.-E. Sarlin, D. DeTone, T. Malisiewicz y A. Rabinovich, «Superglue: Learning feature matching with graph neural networks», en *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, págs. 4938-4947.

[47] B. Cao, A. Araujo y J. Sim, «Unifying deep local and global features for image search», en *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XX 16*, Springer, 2020, págs. 726-743.

[48] A. Krizhevsky, I. Sutskever y G. E. Hinton, «Imagenet classification with deep convolutional neural networks», *Advances in neural information processing systems*, vol. 25, 2012.

[49] N. Sünderhauf, S. Shirazi, F. Dayoub, B. Upcroft y M. Milford, «On the performance of convnet features for place recognition», en *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, IEEE, 2015, págs. 4297-4304.

[50] Z. Chen, O. Lam, A. Jacobson y M. Milford, «Convolutional neural network-based place recognition», *arXiv preprint arXiv:1411.1509*, 2014.

[51] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus e Y. LeCun, «Overfeat: Integrated recognition, localization and detection using convolutional networks», *arXiv preprint arXiv:1312.6229*, 2013.

[52] P. Wozniak, H. Afrisal, R. G. Esparza y B. Kwolek, «Scene recognition for indoor localization of mobile robots using deep CNN», en *International Conference on Computer Vision and Graphics*, Springer, 2018, págs. 137-147.

[53] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla y J. Sivic, «NetVLAD: CNN architecture for weakly supervised place recognition», en *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, págs. 5297-5307. DOI: 10.48550/arXiv.1511.07247.

[54] G. Berton, G. Trivigno, B. Caputo y C. Masone, «Eigenplaces: Training viewpoint robust models for visual place recognition», en *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, págs. 11 080-11 090. DOI: 10.48550/arXiv.2308.10832.

[55] A. Maćkiewicz y W. Ratajczak, «Principal components analysis (PCA)», *Computers & Geosciences*, vol. 19, n.º 3, págs. 303-342, 1993.

[56] A. Ali-Bey, B. Chaib-Draa y P. Giguere, «MixVPR: Feature mixing for visual place recognition», en *Proceedings of the IEEE/CVF winter conference on ap-*

plications of computer vision, 2023, págs. 2998-3007. DOI: 10.48550/arXiv.2303.02190.

[57] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly y col., «An image is worth 16x16 words: Transformers for image recognition at scale», *arXiv preprint arXiv:2010.11929*, 2020. DOI: 10.48550/arXiv.2010.11929.

[58] N. Keetha, A. Mishra, J. Karhade, K. M. Jatavallabhula, S. Scherer, M. Krishna y S. Garg, «AnyLoc: Towards universal visual place recognition», *IEEE Robotics and Automation Letters*, 2023. DOI: 10.1109/LRA.2023.3343602.

[59] S. Izquierdo y J. Civera, «Optimal transport aggregation for visual place recognition», en *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, págs. 17 658-17 668. DOI: 10.48550/arXiv.2311.15937.

[60] S. Hausler, S. Garg, M. Xu, M. Milford y T. Fischer, «Patch-netvlad: Multi-scale fusion of locally-global descriptors for place recognition», en *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, págs. 14 141-14 152.

[61] S. Zhu, L. Yang, C. Chen, M. Shah, X. Shen y H. Wang, «R2Former: Unified Retrieval and Reranking Transformer for Place Recognition», en *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, págs. 19 370-19 380.

[62] R. Wang, Y. Shen, W. Zuo, S. Zhou y N. Zheng, «TransVPR: Transformer-Based Place Recognition With Multi-Level Attention Aggregation», en *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, págs. 13 648-13 657.

[63] F. Lu, L. Zhang, X. Lan, S. Dong, Y. Wang y C. Yuan, *Towards Seamless Adaptation of Pre-trained Models for Visual Place Recognition*, 2024. arXiv: 2402.14505 [cs.CV]. dirección: https://arxiv.org/abs/2402.14505.

[64] S. Hausler y P. Moghadam, «Pair-VPR: Place-Aware Pre-Training and Contrastive Pair Classification for Visual Place Recognition With Vision Transformers», *IEEE Robotics and Automation Letters*, vol. 10, n.º 4, págs. 4013-4020, 2025. DOI: 10.1109/LRA.2025.3546512.

[65] S. Garg y M. Milford, «SeqNet: Learning Descriptors for Sequence-Based Hierarchical Place Recognition», *IEEE Robotics and Automation Letters*, vol. 6, n.º 3, págs. 4305-4312, 2021. DOI: 10.1109/LRA.2021.3067633.

[66] M. J. Milford y G. F. Wyeth, «SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights», en *2012 IEEE international conference on robotics and automation*, IEEE, 2012, págs. 1643-1649.

[67] A. D. Hines, P. G. Stratton, M. Milford y T. Fischer, «VPRTempo: A Fast Temporally Encoded Spiking Neural Network for Visual Place Recognition», en *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, págs. 10 200-10 207. DOI: 10.1109/ICRA57147.2024.10610918.

[68] B. Ferrarini, M. J. Milford, K. D. McDonald-Maier y S. Ehsan, «Binary Neural Networks for Memory-Efficient and Effective Visual Place Recognition in Changing Environments», *IEEE Transactions on Robotics*, vol. 38, n.º 4, págs. 2617-2631, 2022. DOI: 10.1109/TRO.2022.3148908.

[69] K. Cai, C. X. Lu y X. Huang, «STUN: Self-teaching uncertainty estimation for place recognition», en *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2022, págs. 6614-6621.

[70] D. Gao, C. Wang y S. Scherer, «AirLoop: Lifelong Loop Closure Detection», en *2022 International Conference on Robotics and Automation (ICRA)*, 2022, págs. 10 664-10 671. DOI: 10.1109/ICRA46639.2022.9811658.

[71] J. Li, Q. Liu, B. Wang, H. Liu e Y. Han, «RangePlace: A Hierarchical Range Image Transformer for LiDAR-Based Place Recognition», *IEEE Transactions on Intelligent Vehicles*, 2024.

[72] X. Chen, T. Läbe, A. Milioto, T. Röhling, J. Behley y C. Stachniss, «OverlapNet: A siamese network for computing LiDAR scan similarity with applications to loop closing and localization», *Autonomous Robots*, págs. 1-21, 2022.

[73] L. Luo, S. Zheng, Y. Li, Y. Fan, B. Yu, S.-Y. Cao, J. Li y H.-L. Shen, «BEVPlace: Learning LiDAR-based place recognition using bird's eye view images», en *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, págs. 8700-8709.

[74] L. Luo, S.-Y. Cao, B. Han, H.-L. Shen y J. Li, «BVMatch: Lidar-based place recognition using bird's-eye view images», *IEEE Robotics and Automation Letters*, vol. 6, n.º 3, págs. 6076-6083, 2021.

[75] C. R. Qi, H. Su, K. Mo y L. J. Guibas, «Pointnet: Deep learning on point sets for 3D classification and segmentation», en *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, págs. 652-660.

[76] M. Bosse y R. Zlot, «Place recognition using keypoint voting in large 3D lidar datasets», en *2013 IEEE international conference on robotics and automation*, IEEE, 2013, págs. 2677-2684.

[77] T. Röhling, J. Mack y D. Schulz, «A fast histogram-based similarity measure for detecting loop closures in 3-d lidar data», en *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, IEEE, 2015, págs. 736-741.

[78] Y. Wang, Z. Sun, C.-Z. Xu, S. E. Sarma, J. Yang y H. Kong, «LiDAR Iris for Loop-Closure Detection», en *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, págs. 5769-5775. DOI: 10.1109/IROS45743.2020.9341010.

[79] G. Kim y A. Kim, «Scan context: Egocentric spatial descriptor for place recognition within 3D point cloud map», en *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2018, págs. 4802-4809.

[80] T. Shan, B. Englot, F. Duarte, C. Ratti y D. Rus, «Robust Place Recognition using an Imaging Lidar», en *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, págs. 5469-5475. DOI: 10.1109/ICRA48506.2021.9562105.

[81] Y. Cui, X. Chen, Y. Zhang, J. Dong, Q. Wu y F. Zhu, «Bow3d: Bag of words for real-time loop closing in 3d lidar slam», *IEEE Robotics and Automation Letters*, vol. 8, n.º 5, págs. 2828-2835, 2022.

[82] Y. Cui, Y. Zhang, J. Dong, H. Sun, X. Chen y F. Zhu, «Link3d: Linear keypoints representation for 3d lidar point cloud», *IEEE Robotics and Automation Letters*, vol. 9, n.º 3, págs. 2128-2135, 2024.

[83] C. R. Qi, L. Yi, H. Su y L. J. Guibas, «Pointnet++: Deep hierarchical feature learning on point sets in a metric space», *Advances in neural information processing systems*, vol. 30, 2017.

[84] J. Komorowski, «Minkloc3D: Point cloud based large-scale place recognition», en *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021, págs. 1790-1799.

[85] Z. Zhou, C. Zhao, D. Adolfsson, S. Su, Y. Gao, T. Duckett y L. Sun, «NDT-transformer: Large-scale 3D point cloud localisation using the normal distribution transform representation», en *2021 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2021, págs. 5654-5660.

[86] L. Hui, H. Yang, M. Cheng, J. Xie y J. Yang, «Pyramid point cloud transformer for large-scale place recognition», en *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, págs. 6098-6107.

[87] T.-X. Xu, Y.-C. Guo, Z. Li, G. Yu, Y.-K. Lai y S.-H. Zhang, «TransLoc3D: Point cloud based large-scale place recognition using adaptive receptive fields», *arXiv preprint arXiv:2105.11605*, 2021.

[88] J. Ma, J. Zhang, J. Xu, R. Ai, W. Gu y X. Chen, «OverlapTransformer: An efficient and yaw-angle-invariant transformer network for LiDAR-based place recognition», *IEEE Robotics and Automation Letters*, vol. 7, n.º 3, págs. 6958-6965, 2022.

[89] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin y B. Guo, «Swin transformer: Hierarchical vision transformer using shifted windows», en *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, págs. 10 012-10 022.

[90] T. Cohen y M. Welling, «Group Equivariant Convolutional Networks», en *Proceedings of The 33rd International Conference on Machine Learning*, M. F. Balcan y K. Q. Weinberger, eds., ép. Proceedings of Machine Learning Research, vol. 48, New York, New York, USA: PMLR, 2016, págs. 2990-2999. dirección: https://proceedings.mlr.press/v48/cohenc16.html.

[91] Z. Liu, S. Zhou, C. Suo, P. Yin, W. Chen, H. Wang, H. Li e Y.-H. Liu, «LPD-Net: 3D point cloud learning for large-scale place recognition and environment analysis», en *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, págs. 2831-2840.

[92] T. G. Phillips, N. Guenther y P. R. McAree, «When the dust settles: The four behaviors of lidar in the presence of fine airborne particulates», *Journal of field robotics*, vol. 34, n.º 5, págs. 985-1009, 2017.

[93] W. Kuang, X. Zhao, Y. Shen, C. Wen, H. Lu, Z. Zhou y X. Chen, «Reslpr: A lidar data restoration network and benchmark for robust place recognition against weather corruptions», *arXiv preprint arXiv:2503.12350*, 2025.

[94] J. Knights, P. Moghadam, M. Ramezani, S. Sridharan y C. Fookes, «Incloud: Incremental learning for point cloud place recognition», en *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2022, págs. 8559-8566.

[95] G. Kim, Y. S. Park, Y. Cho, J. Jeong y A. Kim, «Mulran: Multimodal range dataset for urban place recognition», en *2020 IEEE international conference on robotics and automation (ICRA)*, IEEE, 2020, págs. 6246-6253.

[96] H. Jang, M. Jung y A. Kim, «Raplace: Place recognition for imaging radar using radon transform and mutable threshold», en *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2023, págs. 11 194-11 201.

[97] M. Gadd y P. Newman, «Open-radvlad: Fast and robust radar place recognition», en *2024 IEEE Radar Conference (RadarConf24)*, IEEE, 2024, págs. 1-6.

[98] B. Choi, H. Kim e Y. Cho, «Referee: Radar-based efficient global descriptor using a feature and free space for place recognition», *arXiv preprint arXiv:2403.14176*, 2024.

[99] S. Saftescu, M. Gadd, D. De Martini, D. Barnes y P. Newman, «Kidnapped Radar: Topological Radar Localisation using Rotationally-Invariant Metric Learning», en *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, págs. 4358-4364. DOI: 10.1109/ICRA40945.2020.9196682.

[100] D. Barnes, M. Gadd, P. Murcutt, P. Newman e I. Posner, «The Oxford Radar RobotCar Dataset: A Radar Extension to the Oxford RobotCar Dataset», en *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, págs. 6433-6438. DOI: 10.1109/ICRA40945.2020.9196884.

[101] M. Gadd, D. De Martini y P. Newman, «Look Around You: Sequence-based Radar Place Recognition with Learned Rotational Invariance», en *2020 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, 2020, págs. 270-276. DOI: 10.1109/PLANS46316.2020.9109951.

[102] D. De Martini, M. Gadd y P. Newman, «KRadar++: Coarse-to-fine FMCW scanning radar localisation», *Sensors*, vol. 20, n.º 21, pág. 6002, 2020.

[103] J. Komorowski, M. Wysoczanska y T. Trzcinski, «Large-scale topological radar localization using learned descriptors», en *Neural Information Processing: 28th International Conference, ICONIP 2021, Sanur, Bali, Indonesia, December 8–12, 2021, Proceedings, Part II 28*, Springer, 2021, págs. 451-462.

[104] M. Usuelli, M. Frosi, P. Cudrano, S. Mentasti y M. Matteucci, «RadarLCD: Learnable radar-based loop closure detection pipeline», en *2024 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2024, págs. 1-7.

[105] K. Burnett, D. J. Yoon, A. P. Schoellig y T. D. Barfoot, «Radar odometry combining probabilistic estimation and unsupervised feature learning», *arXiv preprint arXiv:2105.14152*, 2021.

[106] D. Cattaneo, M. Vaghi y A. Valada, «Lcdnet: Deep loop closure detection and point cloud registration for LiDAR SLAM», *IEEE Transactions on Robotics*, vol. 38, n.º 4, págs. 2074-2093, 2022.

[107] S. Agarwal, J. Yuan, P. Newman, D. De Martini y M. Gadd, «Bayesian Radar Cosplace: Directly estimating location uncertainty in radar place recognition», *IET Radar, Sonar & Navigation*, vol. 19, n.º 1, e70002, 2025.

[108] K. Cai, B. Wang y C. X. Lu, «Autoplace: Robust place recognition with single-chip automotive radar», en *2022 International Conference on Robotics and Automation (ICRA)*, IEEE, 2022, págs. 2222-2228.

[109] C. Meng, Y. Duan, C. He, D. Wang, X. Fan e Y. Zhang, «mmPlace: Robust Place Recognition With Intermediate Frequency Signal of Low-Cost Single-Chip Millimeter Wave Radar», *IEEE Robotics and Automation Letters*, vol. 9, n.º 6, págs. 4878-4885, 2024. DOI: 10.1109/LRA.2024.3377562.

[110] D. C. Herraez, L. Chang, M. Zeller, L. Wiesmann, J. Behley, M. Heidingsfeld y C. Stachniss, «Spr: Single-scan radar place recognition», *IEEE Robotics and Automation Letters*, 2024.

[111] G. Peng, H. Li, Y. Zhao, J. Zhang, Z. Wu, P. Zheng y D. Wang, «Transloc4d: Transformer-based 4d radar place recognition», en *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, págs. 17 595-17 605.

[112] S. Lu, G. Zhuo, H. Wang, Q. Zhou, H. Zhou, R. Huang, M. Huang, L. Zheng y Q. Shu, «TDFANet: Encoding Sequential 4D Radar Point Clouds Using Trajectory-Guided Deformable Feature Aggregation for Place Recognition», *arXiv preprint arXiv:2504.05103*, 2025.

[113] Y. Chen, Y. Zhuang, B. Wang y J. Huai, «4D RadarPR: Context-Aware 4D Radar Place Recognition in harsh scenarios», *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 221, págs. 210-223, 2025.

[114] M. Gadd, D. De Martini, O. Bartlett, P. Murcutt, M. Towlson, M. Widojo, V. Mușat, L. Robinson, E. Panagiotaki, G. Pramatarov y col., «Oord: The oxford offroad radar dataset», *IEEE Transactions on Intelligent Transportation Systems*, 2024.

[115] S. Garg, N. Suenderhauf y M. Milford, «Semantic–geometric visual place recognition: a new perspective for reconciling opposing views», *The International Journal of Robotics Research*, vol. 41, n.º 6, págs. 573-598, 2022.

[116] G. Lin, A. Milan, C. Shen e I. Reid, «Refinenet: Multi-path refinement networks for high-resolution semantic segmentation», en *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, págs. 1925-1934.

[117] T. Naseer, G. L. Oliveira, T. Brox y W. Burgard, «Semantics-aware visual localization under challenging perceptual conditions», en *2017 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2017, págs. 2614-2620.

[118] G. L. Oliveira, W. Burgard y T. Brox, «Efficient Deep Methods for Monocular Road Segmentation.», en *IEEE/RSJ international conference on intelligent robots and systems (IROS 2016)*, 2016.

[119] M. Larsson, E. Stenborg, C. Toft, L. Hammarstrand, T. Sattler y F. Kahl, «Fine-grained segmentation networks: Self-supervised segmentation for improved long-term visual localization», en *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, págs. 31-41.

[120] N. Merrill y G. Huang, «CALC2. 0: Combining appearance, semantic and geometric information for robust and efficient visual loop closure», en *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2019, págs. 4554-4561.

[121] Y. Zhang y X. Zhao, «Mesa: Matching everything by segmenting anything», en *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024, págs. 20 217-20 226.

[122] K. Garg, S. S. Puligilla, S. Kolathaya, M. Krishna y S. Garg, «Revisit Anything: Visual Place Recognition via Image Segment Retrieval», en *European Conference on Computer Vision*, Springer, 2024, págs. 326-343.

[123] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo y col., «Segment anything», en *Pro-*

ceedings of the IEEE/CVF international conference on computer vision, 2023, págs. 4015-4026.

[124] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark y col., «Learning transferable visual models from natural language supervision», en *International conference on machine learning*, PmLR, 2021, págs. 8748-8763.

[125] J. Chen, D. Barath, I. Armeni, M. Pollefeys y H. Blum, «"Where am I?" Scene Retrieval with Language», en *European Conference on Computer Vision*, Springer, 2024, págs. 201-220.

[126] P. Yin, L. Xu, Z. Feng, A. Egorov y B. Li, «Pse-match: A viewpoint-free place recognition method with parallel semantic embedding», *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, n.º 8, págs. 11 249-11 260, 2021.

[127] B. Wu, A. Wan, X. Yue y K. Keutzer, «Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud», en *2018 IEEE international conference on robotics and automation (ICRA)*, IEEE, 2018, págs. 1887-1893.

[128] G. Pramatarov, D. De Martini, M. Gadd y P. Newman, «BoxGraph: Semantic Place Recognition and Pose Estimation from 3D LiDAR», en *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, págs. 7004-7011. DOI: 10.1109/IROS47612.2022.9981266.

[129] M. Milford, C. Shen, S. Lowry, N. Suenderhauf, S. Shirazi, G. Lin, F. Liu, E. Pepperell, C. Lerma, B. Upcroft e I. Reid, «Sequence Searching With Deep-Learnt Depth for Condition- and Viewpoint-Invariant Route-Based Place Recognition», en *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2015.

[130] F. Liu, C. Shen y G. Lin, «Deep convolutional neural fields for depth estimation from a single image», en *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, págs. 5162-5170.

[131] E. Sizikova, V. K. Singh, B. Georgescu, M. Halber, K. Ma y T. Chen, «Enhancing Place Recognition Using Joint Intensity - Depth Analysis and Synthetic Data», en *Computer Vision – ECCV 2016 Workshops*, G. Hua y H. Jégou, eds., Cham: Springer International Publishing, 2016, págs. 901-908, ISBN: 978-3-319-49409-8.

[132] A. Handa, V. Patraucean, V. Badrinarayanan, S. Stent y R. Cipolla, «Understanding real world indoor scenes with synthetic data», en *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, págs. 4077-4085.

[133] S. Garg, M. Babu V, T. Dharmasiri, S. Hausler, N. Suenderhauf, S. Kumar, T. Drummond y M. Milford, «Look No Deeper: Recognizing Places from Opposing Viewpoints under Varying Scene Appearance using Single-View Depth Estimation», en *2019 International Conference on Robotics and Automation (ICRA)*, 2019, págs. 4916-4923. DOI: 10.1109/ICRA.2019.8794178.

[134] M. Babu V, A. Majumder, K. Das y S. Kumar, «A deeper insight into the undemon: Unsupervised deep network for depth and ego-motion estimation», *arXiv preprint arXiv:1809.00969*, 2018.

[135] A. Oertel, T. Cieslewski y D. Scaramuzza, «Augmenting Visual Place Recognition With Structural Cues», *IEEE Robotics and Automation Letters*, vol. 5, n.º 4, págs. 5534-5541, 2020. DOI: 10.1109/LRA.2020.3009077.

[136] A. Milioto, I. Vizzo, J. Behley y C. Stachniss, «Rangenet++: Fast and accurate lidar semantic segmentation», en *2019 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, IEEE, 2019, págs. 4213-4220.

[137] S. Xie, C. Pan, Y. Peng, K. Liu y S. Ying, «Large-scale place recognition based on camera-lidar fused descriptor», *Sensors*, vol. 20, n.º 10, pág. 2870, 2020.

[138] Y. Lu, F. Yang, F. Chen y D. Xie, «Pic-net: Point cloud and image collaboration network for large-scale place recognition», *arXiv preprint arXiv:2008.00658*, 2020.

[139] J. Komorowski, M. Wysoczańska y T. Trzcinski, «MinkLoc++: lidar and monocular image fusion for place recognition», en *2021 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2021, págs. 1-8.

[140] H. Lai, P. Yin y S. Scherer, «Adafusion: Visual-lidar fusion with adaptive weights for place recognition», *IEEE Robotics and Automation Letters*, vol. 7, n.º 4, págs. 12 038-12 045, 2022.

[141] Y. Pan, X. Xu, W. Li, Y. Cui, Y. Wang y R. Xiong, «Coral: Colored structural representation for bi-modal place recognition», en *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2021, págs. 2084-2091.

[142] Z. Zhou, J. Xu, G. Xiong y J. Ma, «LCPR: A Multi-Scale Attention-Based LiDAR-Camera Fusion Network for Place Recognition», *IEEE Robotics and Automation Letters*, 2023. DOI: 10.1109/LRA.2023.3346753.

[143] W. Liu, J. Fei y Z. Zhu, «MFF-PR: Point cloud and image multi-modal feature fusion for place recognition», en *2022 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, IEEE, 2022, págs. 647-655.

[144] J. Xu, J. Ma, Q. Wu, Z. Zhou, Y. Wang, X. Chen, W. Yu y L. Pei, «Explicit Interaction for Fusion-Based Place Recognition», en *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024, págs. 3318-3325. DOI: 10.1109/IROS58592.2024.10802665.

[145] Z. Qi, J. Ma, J. Xu, Z. Zhou, L. Cheng y G. Xiong, «GSPR: Multimodal Place Recognition Using 3D Gaussian Splatting for Autonomous Driving», *arXiv preprint arXiv:2410.00299*, 2024.

[146] Y. Wang, J. Deng, Y. Li, J. Hu, C. Liu, Y. Zhang, J. Ji, W. Ouyang e Y. Zhang, «Bi-lrfusion: Bi-directional lidar-radar fusion for 3d dynamic object detection», en *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, págs. 13 394-13 403.

[147] G. Bang, K. Choi, J. Kim, D. Kum y J. W. Choi, «Radardistill: Boosting radar-based object detection performance via knowledge distillation from lidar features», en *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, págs. 15 491-15 500.

[148] Z. Qi, L. Cheng, Z. Zhou y G. Xiong, «LRFusionPR: A Polar BEV-Based LiDAR-Radar Fusion Network for Place Recognition», *arXiv preprint arXiv:2504.19186*, 2025.

[149] H. Li, Y. Ma, Y. Gu, K. Hu, Y. Liu y X. Zuo, «Radarcam-depth: Radar-camera fusion for depth estimation with learned metric scale», en *2024 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2024, págs. 10 665-10 672.

[150] R. Nabati y H. Qi, «Centerfusion: Center-based radar and camera fusion for 3d object detection», en *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 2021, págs. 1527-1536.

[151] Y. Wang, Z. Jiang, Y. Li, J.-N. Hwang, G. Xing y H. Liu, «RODNet: A real-time radar object detection network cross-supervised by camera-radar fused object 3D localization», *IEEE Journal of Selected Topics in Signal Processing*, vol. 15, n.º 4, págs. 954-967, 2021.

[152] F. Nobis, M. Geisslinger, M. Weber, J. Betz y M. Lienkamp, «A deep learning-based radar and camera sensor fusion architecture for object detection», en *2019 Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, IEEE, 2019, págs. 1-7.

[153] S. Fu, Y. Duan, Y. Li, C. Meng, Y. Wang, J. Ji e Y. Zhang, «Crplace: Camera-radar fusion with bev representation for place recognition», en *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2024, págs. 8421-8427.

[154] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang y O. Beijbom, «Pointpillars: Fast encoders for object detection from point clouds», en *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, págs. 12 697-12 705.

[155] S. Wang, R. She, Q. Kang, X. Jian, K. Zhao, Y. Song y W. P. Tay, «Distilvpr: Cross-modal knowledge distillation for visual place recognition», en *Proceedings of the AAAI conference on artificial intelligence*, vol. 38, 2024, págs. 10 377-10 385.

[156] D. Cattaneo, M. Vaghi, S. Fontana, A. L. Ballardini y D. G. Sorrenti, «Global visual localization in LiDAR-maps through shared 2D-3D embedding space», en *2020 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2020, págs. 4365-4371.

[157] P. Yin, L. Xu, J. Zhang, H. Choset y S. Scherer, «i3dLoc: Image-to-range cross-domain localization robust to inconsistent environmental conditions», *arXiv preprint arXiv:2105.12883*, 2021.

[158] W. Xie, L. Luo, N. Ye, Y. Ren, S. Du, M. Wang, J. Xu, R. Ai, W. Gu y X. Chen, «ModaLink: Unifying Modalities for Efficient Image-to-PointCloud Place Recognition», en *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024, págs. 3326-3333. DOI: 10.1109/IROS58592.2024.10801556.

[159] Y.-J. Li, M. Gladkova, Y. Xia, R. Wang y D. Cremers, «VXP: Voxel-Cross-Pixel Large-scale Image-LiDAR Place Recognition», *arXiv preprint arXiv:2403.14594*, 2024. DOI: 10.48550/arXiv.2403.14594.

[160] Y. Zhou y O. Tuzel, «Voxelnet: End-to-end learning for point cloud based 3d object detection», en *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, págs. 4490-4499.

[161] S. Shubodh, M. Omama, H. Zaidi, U. S. Parihar y M. Krishna, «Lip-loc: Lidar image pretraining for cross-modal localization», en *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2024, págs. 948-957.

[162] Y. Xia, Z. Li, Y.-J. Li, L. Shi, H. Cao, J. F. Henriques y D. Cremers, «UniLoc: Towards Universal Place Recognition Using Any Single Modality», *arXiv preprint arXiv:2412.12079*, 2024.

[163] Y. Ma, X. Zhao, H. Li, Y. Gu, X. Lang e Y. Liu, «RoLM: Radar on LiDAR Map Localization», en *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, págs. 3976-3982. DOI: 10.1109/ICRA48891.2023.10161203.

[164] A. Nayak, D. Cattaneo y A. Valada, «RaLF: Flow-based Global and Metric Radar Localization in LiDAR Maps», en *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, págs. 5097-5103. DOI: 10.1109/ICRA57147.2024.10610626.

[165] Z. Teed y J. Deng, «Raft: Recurrent all-pairs field transforms for optical flow», en *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, Springer, 2020, págs. 402-419.

[166] D. Lisus, J. Laconte, K. Burnett, Z. Zhang y T. D. Barfoot, «Pointing the Way: Refining Radar-Lidar Localization Using Learned ICP Weights», *arXiv preprint arXiv:2309.08731*, 2023.

[167] H. Yin, Y. Wang, J. Wu y R. Xiong, «Radar style transfer for metric robot localisation on lidar maps», *CAAI Transactions on Intelligence Technology*, vol. 8, n/a-n/a, jun. de 2022. DOI: 10.1049/cit2.12112.

[168] P. Isola, J.-Y. Zhu, T. Zhou y A. A. Efros, «Image-to-image translation with conditional adversarial networks», en *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, págs. 1125-1134.

[169] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein y col., «Imagenet large scale visual recognition challenge», *International journal of computer vision*, vol. 115, págs. 211-252, 2015.

[170] H. Sinha, J. Patrikar, E. G. Dhekane, G. Pandey y M. Kothari, «Convolutional Neural Network Based Sensors for Mobile Robot Relocalization», en *2018 23rd International Conference on Methods & Models in Automation & Robotics (MMAR)*, IEEE, 2018, págs. 774-779.

[171] O. Moolan-Feroze, K. Karachalios, D. N. Nikolaidis y A. Calway, «Improving drone localisation around wind turbines using monocular model-based tracking», en *2019 International Conference on Robotics and Automation (ICRA)*, IEEE, 2019, págs. 7713-7719.

[172] S. Brahmbhatt, J. Gu, K. Kim, J. Hays y J. Kautz, «Geometry-aware learning of maps for camera localization», en *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, págs. 2616-2625.

[173] P. Weinzaepfel, G. Csurka, Y. Cabon y M. Humenberger, «Visual localization by learning objects-of-interest dense match regression», en *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, págs. 5634-5643.

[174] R. Li, Q. Liu, J. Gui, D. Gu y H. Hu, «Indoor relocalization in challenging environments with dual-stream convolutional neural networks», *IEEE Transactions on Automation Science and Engineering*, vol. 15, n.º 2, págs. 651-662, 2017.

[175] D. Cattaneo, M. Vaghi, A. L. Ballardini, S. Fontana, D. G. Sorrenti y W. Burgard, «CMRNet: Camera to LiDAR-Map Registration», en *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, págs. 1283-1289. DOI: 10.1109/ITSC.2019.8917470.

[176] Q. Zhao, B. Zhang, S. Lyu, H. Zhang, D. Sun, G. Li y W. Feng, «A CNN-SIFT hybrid pedestrian navigation method based on first-person vision», *Remote Sensing*, vol. 10, n.º 8, pág. 1229, 2018.

[177] L. Ma, J. Chen y col., «Using RGB image as visual input for mapless robot navigation», *arXiv preprint arXiv:1903.09927*, 2019.

[178] Y. Lu y G. Lu, «Deep unsupervised learning for simultaneous visual odometry and depth estimation», en *2019 IEEE International Conference on Image Processing (ICIP)*, IEEE, 2019, págs. 2571-2575.

[179] W. Liu, Y. Mo y J. Jiao, «An efficient edge-feature constraint visual SLAM», en *Proceedings of the International Conference on Artificial Intelligence, Information Processing and Cloud Computing*, 2019, págs. 1-7.

[180] S. Cebollada, L. Payá, M. Flores, A. Peidró y O. Reinoso, «A state-of-the-art review on mobile robotics tasks using artificial intelligence and visual data», *Expert Systems with Applications*, pág. 114 195, 2020.

[181] P. Sharma, H. Liu, H. Wang y S. Zhang, «Securing wireless communications of connected vehicles with artificial intelligence», en *2017 IEEE international symposium on technologies for homeland security (HST)*, IEEE, 2017, págs. 1-7.

[182] R. Polvara, S. Sharma, J. Wan, A. Manning y R. Sutton, «Obstacle avoidance approaches for autonomous navigation of unmanned surface vehicles», *The Journal of Navigation*, vol. 71, n.º 1, págs. 241-256, 2018.

[183] D. Organisciak, D. Sakkos, E. S. Ho, N. Aslam y H. P. Shum, «Unifying person and vehicle re-identification», *IEEE Access*, vol. 8, págs. 115 673-115 684, 2020.

[184] Y. Wang, T. Bao, C. Ding y M. Zhu, «Face recognition in real-world surveillance videos with deep learning method», en *2017 2nd International Conference on Image, Vision and Computing (ICIVC)*, IEEE, 2017, págs. 239-243.

[185] W. Jiang y W. Wang, «Face detection and recognition for home service robots with end-to-end deep neural networks», en *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2017, págs. 2232-2236.

[186] S. Hu, H. P. Shum, X. Liang, F. W. Li y N. Aslam, «Facial reshaping operator for controllable face beautification», *Expert Systems with Applications*, vol. 167, pág. 114 067, 2021.

[187] N. Nozawa, H. P. Shum, Q. Feng, E. S. Ho y S. Morishima, «3D car shape reconstruction from a contour sketch using GAN and lazy learning», *The Visual Computer*, págs. 1-14, 2021.

[188] H. F. Zaki, F. Shafait y A. Mian, «Viewpoint invariant semantic object and scene categorization with RGB-D sensors», *Autonomous Robots*, vol. 43, n.º 4, págs. 1005-1022, 2019.

[189] Q. Feng, H. P. Shum y S. Morishima, «Resolving hand-object occlusion for mixed reality with joint deep learning and model optimization», *Computer Animation and Virtual Worlds*, vol. 31, n.º 4-5, e1956, 2020.

[190] G. Tanzmeister, J. Thomas, D. Wollherr y M. Buss, «Grid-based mapping and tracking in dynamic environments using a uniform evidential environment representation», en *2014 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2014, págs. 6090-6095.

[191] A. Holliday y G. Dudek, «Scale-robust localization using general object landmarks», en *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2018, págs. 1688-1694.

[192] X. Ruan, D. Ren, X. Zhu y J. Huang, «Mobile robot navigation based on deep reinforcement learning», en *2019 Chinese control and decision conference (CCDC)*, IEEE, 2019, págs. 6174-6178.

[193] D. Bai, C. Wang, B. Zhang, X. Yi y X. Yang, «CNN feature boosted SeqSLAM for real-time loop closure detection», *Chinese Journal of Electronics*, vol. 27, n.º 3, págs. 488-499, 2018.

[194] T. Naseer, M. Ruhnke, C. Stachniss, L. Spinello y W. Burgard, «Robust visual SLAM across seasons», en *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2015, págs. 2529-2535.

[195] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke y A. Rabinovich, «Going deeper with convolutions», en *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, págs. 1-9.

[196] S. Cebollada, L. Payá, V. Román y O. Reinoso, «Hierarchical localization in topological models under varying illumination using holistic visual descriptors», *IEEE Access*, vol. 7, págs. 49 580-49 595, 2019.

[197] S. Xu, W. Chou y H. Dong, «A Robust Indoor Localization System Integrating Visual Localization Aided by CNN-Based Image Retrieval with Monte Carlo Localization», *Sensors*, vol. 19, n.º 2, pág. 249, 2019.

[198] M. Leyva-Vallina, N. Strisciuglio, M. Lopez-Antequera, R. Tylecek, M. Blaich y N. Petkov, «TB-Places: A Data Set for Visual Place Recognition in Garden Environments.», *IEEE Access*, vol. 7, págs. 52 277-52 287, 2019.

[199] M. Ballesta, L. Payá, S. Cebollada, O. Reinoso y F. Murcia, «A CNN Regression Approach to Mobile Robot Localization Using Omnidirectional Images», *Applied Sciences*, vol. 11, n.º 16, pág. 7521, 2021.

[200] L. V. Utkin, V. S. Zaborovsky y S. G. Popov, «Siamese neural network for intelligent information security control in multi-robot systems», *Automatic Control and Computer Sciences*, vol. 51, n.º 8, págs. 881-887, 2017.

[201] A. Zeng, S. Song, K.-T. Yu, E. Donlon, F. R. Hogan, M. Bauza, D. Ma, O. Taylor, M. Liu, E. Romo y col., «Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching», en *2018 IEEE international conference on robotics and automation (ICRA)*, IEEE, 2018, págs. 3750-3757.

[202] Y. Li y X. Zhang, «SiamVGG: Visual tracking using deeper siamese networks», *arXiv preprint arXiv:1902.02804*, 2019.

[203] M. Leyva-Vallina, N. Strisciuglio y N. Petkov, «Place recognition in gardens by learning visual representations: data set and benchmark analysis», en *International Conference on Computer Analysis of Images and Patterns*, Springer, 2019, págs. 324-335.

[204] M. Leyva-Vallina, N. Strisciuglio y N. Petkov, «Generalized Contrastive Optimization of Siamese Networks for Place Recognition», *arXiv preprint arXiv:2103.06638*, 2021.

[205] H. Yin, L. Tang, X. Ding, Y. Wang y R. Xiong, «LocNet: Global localization in 3D point clouds for mobile vehicles», en *2018 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2018, págs. 728-733.

[206] J. Cabrera, O. Céspedes, S. Cebollada, O. Reinoso y L. Payá, «An evaluation of CNN models and data augmentation techniques in hierarchical localization of mobile robots», *Evolving Systems*, 2024, ISSN: 1868-6486. DOI: 10.1007/s12530-024-09604-6.

[207] J. J. Cabrera, V. Román, A. Gil, O. Reinoso y L. Payá, «An experimental evaluation of Siamese Neural Networks for robot localization using omnidirectional imaging in indoor environments», *Artificial Intelligence Review*, vol. 57, n.º 198, 2024, ISSN: 1573-7462. DOI: 10.1007/s10462-024-10840-0.

[208] M. Ballesta, L. Payá, S. Cebollada, O. Reinoso y F. Murcia, «A CNN Regression Approach to Mobile Robot Localization Using Omnidirectional Images», *Applied Sciences*, vol. 11, n.º 16, pág. 7521, 2021.

[209] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell y S. Xie, «A convnet for the 2020s», en *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, págs. 11 976-11 986.

[210] J. J. Cabrera, S. Cebollada, M. Flores, Ó. Reinoso y L. Payá, «Training, Optimization and Validation of a CNN for Room Retrieval and Description of Omnidirectional Images», *SN Computer Science*, vol. 3, n.º 4, págs. 1-13, 2022.

[211] G. Grisetti, C. Stachniss y W. Burgard, «Improving grid-based slam with rao-blackwellized particle filters by adaptive proposals and selective resampling», en *Proceedings of the 2005 IEEE international conference on robotics and automation*, IEEE, 2005, págs. 2432-2437.

[212] ——, «Improved techniques for grid mapping with rao-blackwellized particle filters», *IEEE transactions on Robotics*, vol. 23, n.º 1, págs. 34-46, 2007.

[213] Y. Wang, Z. Sun, C.-Z. Xu, S. E. Sarma, J. Yang y H. Kong, «LiDAR iris for loop-closure detection», en *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2020, págs. 5769-5775.

[214] Y. Tian, F. Liu, H. Liu, Y. Liu, H. Suwoyo, T. Jin, L. Li y J. Wang, «A Real-Time and Fast LiDAR–IMU–GNSS SLAM System with Point Cloud Semantic Graph Descriptor Loop-Closure Detection», *Advanced Intelligent Systems*, vol. 5, n.º 10, pág. 2 300 138, 2023.

[215] Y. Li, C. P. Chen, N. Maitlo, L. Mi, W. Zhang y J. Chen, «Deep Neural Network–Based Loop Detection for Visual Simultaneous Localization and Mapping Featuring Both Points and Lines», *Advanced Intelligent Systems*, vol. 2, n.º 1, pág. 1 900 107, 2020.

[216] F. Radenović, G. Tolias y O. Chum, «Fine-tuning CNN image retrieval with no human annotation», *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, n.º 7, págs. 1655-1668, 2018.

[217] H. Song, W. Choi y H. Kim, «Robust Vision-Based Relative-Localization Approach Using an RGB-Depth Camera and LiDAR Sensor Fusion», *IEEE Transactions on Industrial Electronics*, vol. 63, n.º 6, págs. 3725-3736, 2016. DOI: 10.1109/TIE.2016.2521346.

[218] K. Żywanowski, A. Banaszczyk, M. R. Nowicki y J. Komorowski, «MinkLoc3D-SI: 3D LiDAR place recognition with sparse convolutions, spherical coordinates, and intensity», *IEEE Robotics and Automation Letters*, vol. 7, n.º 2, págs. 1079-1086, 2021.

[219] Q. Sun, H. Liu, J. He, Z. Fan y X. Du, «Dagc: Employing dual attention and graph convolution for point cloud based place recognition», en *Proceedings of the 2020 International Conference on Multimedia Retrieval*, 2020, págs. 224-232.

[220] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser e I. Polosukhin, «Attention is all you need», *Advances in neural information processing systems*, vol. 30, 2017.

[221] Y. Xia, Y. Xu, S. Li, R. Wang, J. Du, D. Cremers y U. Stilla, «SOE-Net: A self-attention and orientation encoding network for point cloud based place recognition», en *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, 2021, págs. 11 348-11 357.

[222] L. Wiesmann, R. Marcuzzi, C. Stachniss y J. Behley, «Retriever: Point cloud retrieval in compressed 3D maps», en *2022 International Conference on Robotics and Automation (ICRA)*, IEEE, 2022, págs. 10 925-10 932.

[223] Z. Fan, Z. Song, H. Liu, Z. Lu, J. He y X. Du, «SVT-Net: Super light-weight sparse voxel transformer for large scale place recognition», en *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, 2022, págs. 551-560.

[224] Z. Hou, Y. Yan, C. Xu y H. Kong, en *2022 International Conference on Robotics and Automation (ICRA)*, IEEE, 2022, págs. 2612-2618.

[225] F. Radenović, G. Tolias y O. Chum, «Fine-tuning CNN image retrieval with no human annotation», *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, n.º 7, págs. 1655-1668, 2018.

[226] O. Ronneberger, P. Fischer y T. Brox, «U-net: Convolutional networks for biomedical image segmentation», en *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, Springer, 2015, págs. 234-241.

[227] J. Komorowski, «Improving point cloud based place recognition with ranking-based loss and large batch training», en *2022 26th International Conference on Pattern Recognition (ICPR)*, IEEE, 2022, págs. 3699-3705.

[228] N. Sünderhauf, F. Dayoub, S. McMahon, B. Talbot, R. Schulz, P. Corke, G. Wyeth, B. Upcroft y M. Milford, «Place categorization and semantic mapping on a mobile robot», en *2016 IEEE international conference on robotics and automation (ICRA)*, IEEE, 2016, págs. 5729-5736.

[229] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov y L.-C. Chen, «Mobilenetv2: Inverted residuals and linear bottlenecks», en *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, págs. 4510-4520.

[230] D. Hendrycks y K. Gimpel, «Gaussian error linear units (gelus)», *arXiv preprint arXiv:1606.08415*, 2016.

[231] J. L. Ba, J. R. Kiros y G. E. Hinton, «Layer normalization», *arXiv preprint arXiv:1607.06450*, 2016.

[232] S. Ioffe, «Batch renormalization: Towards reducing minibatch dependence in batch-normalized models», *Advances in neural information processing systems*, vol. 30, 2017.

[233] V. Nair y G. E. Hinton, «Rectified linear units improve restricted boltzmann machines», en *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, págs. 807-814.

[234] J. Devlin, M.-W. Chang, K. Lee y K. Toutanova, «Bert: Pre-training of deep bidirectional transformers for language understanding», *arXiv preprint arXiv:1810.04805*, 2018.

[235] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat y col., «Gpt-4 technical report», *arXiv preprint arXiv:2303.08774*, 2023.

[236] W. Zhang y C. Xiao, «PCAN: 3D attention map learning using contextual information for point cloud based retrieval», en *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, págs. 12 436-12 445.

[237] Z. Hou, Y. Shang, T. Gao e Y. Yan, «BPT: binary point cloud transformer for place recognition», *arXiv preprint arXiv:2303.01166*, 2023.

[238] Z. Fan, Z. Song, W. Zhang, H. Liu, J. He y X. Du, «RPR-Net: A point cloud-based rotation-aware large scale place recognition network», en *European Conference on Computer Vision*, Springer, 2022, págs. 709-725.

[239] L. Hui, M. Cheng, J. Xie, J. Yang y M.-M. Cheng, «Efficient 3D point cloud feature learning for large-scale place recognition», *IEEE Transactions on Image Processing*, vol. 31, págs. 1258-1270, 2022.

[240] C. E. Lin, J. Song, R. Zhang, M. Zhu y M. Ghaffari, «Se (3)-equivariant point cloud-based place recognition», en *Conference on Robot Learning*, PMLR, 2023, págs. 1520-1530.

[241] D. W. Shu y J. Kwon, «Hierarchical bidirected graph convolutions for large-scale 3D point cloud place recognition», *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[242] L. Wiesmann, L. Nunes, J. Behley y C. Stachniss, «KPPR: Exploiting momentum contrast for point cloud-based place recognition», *IEEE Robotics and Automation Letters*, vol. 8, n.º 2, págs. 592-599, 2022.

[243] M. Gadd, B. Ramtoula, D. De Martini y P. Newman, «What you see is what you get: Experience ranking with deep neural dataset-to-dataset similarity for topological localisation», en *International Symposium on Experimental Robotics*, Springer, 2023, págs. 595-607. DOI: 10.1007/978-3-031-63596-0_53.

[244] M. Flores, D. Valiente, A. Gil, O. Reinoso y L. Paya, «Efficient probability-oriented feature matching using wide field-of-view imaging», *Engineering Appli-*

cations of Artificial Intelligence, vol. 107, pág. 104 539, 2022. DOI: 10.1016/j.engappai.2021.104539.

[245] R. Kalita, A. K. Talukdar y K. K. Sarma, «Real-Time Human Detection with Thermal Camera Feed using YOLOv3», en *2020 IEEE 17th India Council International Conference (INDICON)*, IEEE, 2020, págs. 1-5. DOI: 10.1109/INDICON49873.2020.9342089.

[246] Y. Wang, W.-L. Chao, D. Garg, B. Hariharan, M. Campbell y K. Q. Weinberger, «Pseudo-LiDAR from visual depth estimation: Bridging the gap in 3D object detection for autonomous driving», en *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, págs. 8445-8453. DOI: 10.48550/arXiv.1812.07179.

[247] M. Zaffar, S. Garg, M. Milford, J. Kooij, D. Flynn, K. McDonald-Maier y S. Ehsan, «VPR-bench: An open-source visual place recognition evaluation framework with quantifiable viewpoint and appearance change», *International Journal of Computer Vision*, vol. 129, n.º 7, págs. 2136-2174, 2021. DOI: 10.1007/s11263-021-01469-5.

[248] M. Humenberger, Y. Cabon, N. Pion, P. Weinzaepfel, D. Lee, N. Guérin, T. Sattler y G. Csurka, «Investigating the role of image retrieval for visual localization: An exhaustive benchmark», *International Journal of Computer Vision*, vol. 130, n.º 7, págs. 1811-1836, 2022. DOI: 10.1007/s11263-022-01615-7.

[249] Z. Liu, H. Hu, Y. Lin, Z. Yao, Z. Xie, Y. Wei, J. Ning, Y. Cao, Z. Zhang, L. Dong y col., «Swin transformer v2: Scaling up capacity and resolution», en *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, págs. 12 009-12 019. DOI: 10.48550/arXiv.2111.09883.

[250] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski y A. Joulin, «Emerging Properties in Self-Supervised Vision Transformers», en *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, págs. 9650-9660. DOI: 10.48550/arXiv.2104.14294.

[251] M. Rostkowska y P. Skrzypczyński, «Optimizing Appearance-Based Localization with Catadioptric Cameras: Small-Footprint Models for Real-Time Inference on Edge Devices», *Sensors*, vol. 23, n.º 14, 2023, ISSN: 1424-8220. DOI: 10.3390/s23146485. dirección: https://www.mdpi.com/1424-8220/23/14/6485.

[252] M. Alfaro, J. J. Cabrera, L. M. Jiménez, O. Reinoso y L. Payá, «Triplet Neural Networks for the Visual Localization of Mobile Robots», en *Proceedings of the 21st International Conference on Informatics in Control, Automation and Robotics (ICINCO 2024)*, vol. 2, 2024, págs. 125-132. DOI: 10.5220/0012927400003822.

[253] H. Yin, X. Xu, S. Lu, X. Chen, R. Xiong, S. Shen, C. Stachniss e Y. Wang, «A survey on global LiDAR localization: Challenges, advances and open problems», *International Journal of Computer Vision*, vol. 132, n.º 8, págs. 3139-3171, 2024. DOI: 10.1007/s11263-024-02019-5.

[254] J. J. Cabrera, A. Santo, A. Gil, C. Viegas y L. Payá, «MinkUNeXt: point cloud-based large-scale place recognition using 3D sparse convolutions», *arXiv preprint arXiv:2403.07593*, 2024.

[255] H. Zhao, L. Jiang, J. Jia, P. H. Torr y V. Koltun, «Point transformer», en *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, págs. 16 259-16 268. DOI: 10.48550/arXiv.2012.09164.

[256] Z. Zhao, H. Yu, C. Lyu, W. Yang y S. Scherer, «Attention-Enhanced Cross-Modal Localization Between Spherical Images and Point Clouds», *IEEE Sensors Journal*, vol. 23, n.º 19, págs. 23 836-23 845, 2023. DOI: 10.1109/JSEN.2023.3306377.

[257] E. Karypidis, I. Kakogeorgiou, S. Gidaris y N. Komodakis, «DINO-Foresight: Looking into the Future with DINO», *arXiv preprint arXiv:2412.11673*, 2024. DOI: 10.48550/arXiv.2412.11673.

[258] K. Xian, Z. Cao, C. Shen y G. Lin, «Towards robust monocular depth estimation: A new baseline and benchmark», *International Journal of Computer Vision*, vol. 132, n.º 7, págs. 2401-2419, 2024. DOI: 10.1007/s11263-023-01979-4.

[259] M. Gui, J. Schusterbauer, U. Prestel, P. Ma, D. Kotovenko, O. Grebenkova, S. A. Baumann, V. T. Hu y B. Ommer, «DepthFM: Fast monocular depth estimation with flow matching», *arXiv preprint arXiv:2403.13788*, 2024. DOI: 10.48550/arXiv.2403.13788.

[260] B. Ke, A. Obukhov, S. Huang, N. Metzger, R. C. Daudt y K. Schindler, «Repurposing diffusion-based image generators for monocular depth estimation», en *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, págs. 9492-9502. DOI: 10.48550/arXiv.2312.02145.

[261] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler y V. Koltun, «Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer», *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, n.º 3, págs. 1623-1637, 2020. DOI: 10.1109/TPAMI.2020.3019967.

[262] S. F. Bhat, R. Birkl, D. Wofk, P. Wonka y M. Müller, «ZoeDepth: Zero-shot transfer by combining relative and metric depth», *arXiv preprint arXiv:2302.12288*, 2023. DOI: 10.48550/arXiv.2302.12288.

[263] L. Yang, B. Kang, Z. Huang, X. Xu, J. Feng y H. Zhao, «Depth Anything: Unleashing the Power of Large-Scale Unlabeled Data», en *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, págs. 10 371-10 381. DOI: 10.48550/arXiv.2401.10891.

[264] W. Hu, X. Gao, X. Li, S. Zhao, X. Cun, Y. Zhang, L. Quan e Y. Shan, «Depthcrafter: Generating consistent long depth sequences for open-world videos», *arXiv preprint arXiv:2409.02095*, 2024. DOI: 10.48550/arXiv.2409.02095.

[265] S. Chen, H. Guo, S. Zhu, F. Zhang, Z. Huang, J. Feng y B. Kang, «Video Depth Anything: Consistent Depth Estimation for Super-Long Videos», *arXiv preprint arXiv:2501.12375*, 2025. DOI: 10.48550/arXiv.2501.12375.

[266] Y. Guo, S. Garg, S. M. H. Miangoleh, X. Huang y L. Ren, «Depth Any Camera: Zero-Shot Metric Depth Estimation from Any Camera», *arXiv preprint arXiv:2501.02464*, 2025. DOI: 10.48550/arXiv.2501.02464.

[267] G. Xu, Y. Ge, M. Liu, C. Fan, K. Xie, Z. Zhao, H. Chen y C. Shen, «What Matters When Repurposing Diffusion Models for General Dense Perception Tasks?», *arXiv preprint arXiv:2403.06090*, 2024.

[268] L. Yan, P. Yan, S. Xiong, X. Xiang e Y. Tan, «MonoCD: Monocular 3D Object Detection with Complementary Depths», en *Proceedings of the IEEE/CVF Con-*

ference on Computer Vision and Pattern Recognition (CVPR), 2024, págs. 10 248-10 257. DOI: 10.48550/arXiv.2404.03181.

[269] A. Ganj, Y. Zhao, H. Su y T. Guo, «Mobile AR depth estimation: Challenges & prospects», en *Proceedings of the 25th International Workshop on Mobile Computing Systems and Applications*, 2024, págs. 21-26. DOI: 10.1145/3638550.3641122.

[270] J. J. Han, A. Acar, C. Henry y J. Y. Wu, «Depth anything in medical images: A comparative study», *arXiv preprint arXiv:2401.16600*, 2024. DOI: 10.48550/arXiv.2401.16600.

[271] D. Hettiarachchi, Y. Tian, H. Yu y S. Kamijo, «Depth as attention to learn image representations for visual localization, using monocular images», *Journal of Visual Communication and Image Representation*, vol. 98, pág. 104 012, 2024. DOI: 10.1016/j.jvcir.2023.104012.

[272] A. Bochkovskii, A. Delaunoy, H. Germain, M. Santos, Y. Zhou, S. R. Richter y V. Koltun, «Depth Pro: Sharp monocular metric depth in less than a second», *arXiv preprint arXiv:2410.02073*, 2024. DOI: doi.org/10.48550/arXiv.2410.02073.

[273] A. J. Lee, S. Song, H. Lim, W. Lee y H. Myung, «LC2: LiDAR-Camera Loop Constraints for Cross-Modal Place Recognition», *IEEE Robotics and Automation Letters*, vol. 8, n.º 6, págs. 3589-3596, 2023.

[274] Z. Zhao, H. Yu, C. Lyu, W. Yang y S. Scherer, «Attention-enhanced cross-modal localization between spherical images and point clouds», *IEEE Sensors Journal*, vol. 23, n.º 19, págs. 23 836-23 845, 2023.

[275] S. Zheng, Y. Li, Z. Yu, B. Yu, S.-Y. Cao, M. Wang, J. Xu, R. Ai, W. Gu, L. Luo y col., «I2P-Rec: Recognizing Images on Large-Scale Point Cloud Maps Through Bird's Eye View Projections», en *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2023, págs. 1395-1400.

[276] A. Geiger, P. Lenz y R. Urtasun, «Are we ready for autonomous driving? the kitti vision benchmark suite», en *2012 IEEE conference on computer vision and pattern recognition*, IEEE, 2012, págs. 3354-3361.

[277] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark y col., «Learning transferable visual models from natural language supervision», en *International conference on machine learning*, PmLR, 2021, págs. 8748-8763.

[278] Y. Li, J. Li, Z. Dong, Y. Wang y B. Yang, «SaliencyI2PLoc: saliency-guided image-point cloud localization using contrastive learning», *Information Fusion*, pág. 103 015, 2025.

[279] X. Cai, Y. Wang, Z. Huang, Y. Shao y D. Li, «VOLoc: Visual Place Recognition by Querying Compressed Lidar Map», en *2024 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2024, págs. 10 192-10 199.

[280] Á. Martínez, A. Santo, M. Ballesta, A. Gil y L. Payá, «A Method for the Calibration of a LiDAR and Fisheye Camera System», *Applied Sciences*, vol. 15, n.º 4, pág. 2044, 2025.

[281] M. Flores, D. Valiente, A. Peidró, O. Reinoso y L. Payá, «Generating a full spherical view by modeling the relation between two fisheye images», *The Visual Computer*, vol. 40, n.º 10, págs. 7107-7132, 2024.

[282] C. Mei y P. Rives, «Single view point omnidirectional camera calibration from planar grids», en *Proceedings 2007 IEEE International Conference on Robotics and Automation*, IEEE, 2007, págs. 3945-3950.

[283] B. Ramtoula, M. Gadd, P. Newman y D. De Martini, «Visual dna: Representing and comparing images using distributions of neuron activations», en *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, págs. 11 113-11 123.

Evolving Systems (2024) 15:1991–2003
https://doi.org/10.1007/s12530-024-09604-6

ORIGINAL PAPER

An evaluation of CNN models and data augmentation techniques in hierarchical localization of mobile robots

Juan José Cabrera[1] · Orlando José Céspedes[1] · Sergio Cebollada[1] · Oscar Reinoso[1,2] · Luis Payá[1]

Received: 31 January 2024 / Accepted: 25 June 2024 / Published online: 8 July 2024

Abstract
This work presents an evaluation of CNN models and data augmentation to carry out the hierarchical localization of a mobile robot by using omnidirectional images. In this sense, an ablation study of different state-of-the-art CNN models used as backbone is presented and a variety of data augmentation visual effects are proposed for addressing the visual localization of the robot. The proposed method is based on the adaption and re-training of a CNN with a dual purpose: (1) to perform a rough localization step in which the model is used to predict the room from which an image was captured, and (2) to address the fine localization step, which consists in retrieving the most similar image of the visual map among those contained in the previously predicted room by means of a pairwise comparison between descriptors obtained from an intermediate layer of the CNN. In this sense, we evaluate the impact of different state-of-the-art CNN models such as ConvNeXt for addressing the proposed localization. Finally, a variety of data augmentation visual effects are separately employed for training the model and their impact is assessed. The performance of the resulting CNNs is evaluated under real operation conditions, including changes in the lighting conditions. Our code is publicly available on the project website https://github.com/juanjo-cabrera/IndoorLocalizationSingleCNN.git.

Keywords Mobile robotics · Omnidirectional vision · Hierarchical localization · Deep learning · Data augmentation

1 Introduction

In the ever-evolving landscape of Artificial Intelligence (AI), Convolutional Neural Networks (CNNs) have become a fundamental pillar of the technology, with disruptive problem-solving capabilities. This kind of neural networks were originally conceived for image recognition tasks, but have quickly transcended their initial boundaries, establishing themselves as a versatile and powerful tool for tackling a wide range of challenges in a variety of domains (LeCun and Bengio 1995).

The increasing use of CNNs can be attributed to their high ability to recognise patterns from different sources of information. This ability has made them essential in a wide variety of applications, from image recognition (Krizhevsky et al. 2012; Simonyan and Zisserman 2014) and object detection (Redmon et al. 2016; Ren et al. 2015) to semantic segmentation (Ronneberger et al. 2015) and even natural language processing (Kim 2014). The success of such architectures is based on their ability to extract features from data, which allows solving high-level problems such as visual localization.

In this sense, some researchers have addressed visual localization by means of 360° vision sensors due to its relatively low cost and the wide range of information they provide. When capturing images in real-world scenarios, especially in robotics applications, the environmental conditions can vary significantly. Consequently, addressing the visual localization could be particularly challenging due to

✉ Juan José Cabrera
juan.cabreram@umh.es

Orlando José Céspedes
orlando.cespedes@goumh.umh.es

Sergio Cebollada
s.cebollada@umh.es

Oscar Reinoso
o.reinoso@umh.es

Luis Payá
lpaya@umh.es

[1] Institute for Engineering Research (I3E), Miguel Hernandez University, Elche, Spain

[2] Valencian Graduate School and Research Network for Artificial Intelligence (valgrAI), Valencia, Spain

different phenomena such as changes in illumination conditions. For this reason, understanding and addressing the effects of illumination changes are crucial for developing robust CNN models.

Related with the above information, the main objective of this work is to analyze the influence of different visual effects applied to the training data in order to carry out the mapping and localization of a mobile robot, which moves in an indoor environment under real operation conditions. For this purpose, the omnidirectional images captured by a catadioptric vision sensor are used to train a CNN. Both the raw images, and some sets of images obtained after introducing visual effects to the original images in a data augmentation process are considered during the training. In this paper, we have also evaluated the performance of state-of-the-art CNN models when addressing localization through a hierarchical approach. In this sense, the CNN will be adapted and re-trained with a dual purpose: (1) to perform a rough localization step in which the model is used to predict the room from which a test image was captured, and (2) to address the fine localization step, which consists in retrieving the most similar image of the visual map among those contained in the previously predicted room by means of a pairwise comparison between descriptors obtained from an intermediate layer of the CNN. The main contributions of this paper can be summarized as follows.

- A CNN is adapted and re-trained to predict the room from which the robot captured an omnidirectional image which is transformed into panoramic. This approach enhances robotic localization by initially performing room recognition.
- We use the re-trained CNN to embed panoramic images into holistic descriptors by extracting the activation of an intermediate layer. These descriptors are compared to the visual model of the retrieved room via nearest neighbour search, providing an efficient method for scene recognition and position retrieval.
- We conduct a thorough study of the individual influence of different data augmentation visual effects when training a model to perform hierarchical localization. This analysis contributes to improve the robustness of the model and its generalization ability in localization tasks.
- We evaluate the performance of different state-of-the-art CNN models that are used as the backbone for the proposed localization task. This comparative evaluation provides valuable insights for selecting the most suitable CNN architecture for real-world localization applications.

This work is an extension of the initial developments presented in Céspedes et al. (2023). In this previous work, we used a basic CNN model (Places, Zhou et al. 2014) to perform the rough localization. However, our present proposal addresses both rough and fine localization steps and studies more exhaustively different state-of-the-art models such as AlexNet (Krizhevsky et al. 2012), ResNet-152 (He et al. 2016), ResNeXt-101 64x4d (Xie et al. 2017), MobileNetV3 (Howard et al. 2019), EfficientNetV2 (Tan and Le 2021) and ConvNeXt Large (Liu et al. 2022). Also, an ablation study of a variety of data augmentation visual effects are carried out with the aim of analysing the performance of the proposed tools under real operation conditions.

The following sections are structured as follows. First, in Sect. 2 we present a review of the state of the art on visual place-recognition and localization by means of artificial intelligence techniques. Second, in Sect. 3 we describe the proposed hierarchical localization method, the different CNN architectures which are evaluated and the proposed data augmentation visual effects. After that, we present in Sect. 4 the dataset used and the experiments carried out to test and validate the proposed method. Finally, conclusions and future works are outlined in Sect. 5.

2 State of the art

Artificial intelligence (AI) techniques are commonly proposed to address computer vision and robotics problems. Recent works, such as Aguilar et al. (2017), propose a pedestrian detector for Unmanned Aerial Vehicles (UAVs) based on Haar-LBP features combined with Adaboost and cascade classifiers with Meanshift. Another example is Wang et al. (2018), which utilizes an autoencoder for the fusion and extraction of multiple visual features from different sensors with the aim of carrying out motion planning based on deep reinforcement learning.

CNNs have proven to be successful in many practical applications. Well-known architectures, such as GoogLeNet (Szegedy et al. 2015), AlexNet (Krizhevsky et al. 2012) and VGG16 (Simonyan and Zisserman 2014) have been used as starting points to address new computer vision tasks. Regarding place-recognition, CNN models were firstly proposed to address this problem in Chen et al. (2014), where a pre-trained model called Overfeat (Sermanet et al. 2013) is used to extract features from images. Sünderhauf et al. (2015) provided a thorough investigation on the performance of extracted features for place recognition. In fact, they found out that the features extracted from convolutional layers were more robust against different lighting conditions than those extracted from fully connected layers which outperformed towards viewpoint changes. Bai et al. (2018) propose the SeqCNNSLAM method, which consists in using the pre-trained AlexNet (Krizhevsky et al. 2012) to extract features and feed the SeqSLAM algorithm (Milford and Wyeth 2012). Also Naseer et al. (2015) proposed a similar

approach, but using GoogleNet (Szegedy et al. 2015). Some of the works have not only used images as source of information, but also point clouds (Uy and Lee 2018) and both combined (Komorowski et al. 2021).

In the context of robot localization, Kopitkov and Indelman (2018) propose using CNN holistic descriptors to estimate the robot position by learning a generative viewpoint-dependent model of CNN features with a spatially-varying Gaussian distribution. Sarlin et al. (2019) carry out a hierarchical modeling using a CNN, which extracts local features and holistic descriptors for 6-DOF localization. In that paper, a coarse localization is solved by using global descriptors, while a fine localization is solved by matching local features. Recent works (Cebollada et al. 2022) have proposed hierarchical visual models for efficient localization. This method involves arranging visual information hierarchically in different layers so that localization can be solved in two main steps. The first step involves coarse localization to roughly determine the area where the robot is located, and the second step involves fine localization within this pre-selected area.

Regarding the training of CNNs, a large and varied dataset is essential. Since a lack of a large enough datasets is quite common, Data Augmentation (DA) can be used to increase the training instances to avoid overfitting. As for the DA for a mobile robot localization task, it is essential to apply visual effects that may occur in real operation conditions to make the model robust against those effects. Considering as many effects as possible would increase the effectiveness of the CNN, but this would imply more processing power and memory. Numerous researchers have leveraged the data augmentation technique as a valuable tool to enhance the efficacy of their models. For example, Ding et al. (2016) train a CNN with three distinct types of data augmentation operations. Their investigation aims to enhance the performance of Synthetic Aperture Radar target recognition by achieving invariance against pose variations. Similarly, Salamon and Bello (2017) present a CNN designed for environmental sound classification, accompanied by an audio data augmentation strategy. This augmentation approach is useful to mitigate the scarcity of data in this domain, contributing to improved model performance. Furthermore, Perez and Wang (2017) present a study about the effectiveness of data augmentation to solve the classification task. Shorten and Khoshgoftaar (2019) present a survey about the existing methods for data augmentation and related developments. Nonetheless, the previously proposed data augmentation methods do not exactly analyze the visual phenomena that can occur when the mobile robot moves through the target environment under real-operation conditions. Therefore, the present work performs a data augmentation analysis that focuses on a wide range of those specific visual effects.

In light of the above information, the aim of this work is to analyze the influence of some visual effects to carry out data augmentation for CNN training to address a hierarchical localization (Cebollada et al. 2022). Hence, the efficiency of each visual effect will be assessed through the ability of the CNN model to robustly estimate the position where the image was captured. In addition, this work focuses on evaluating the performance of different well-known CNN models for both the coarse and fine localization steps. The first one consists in estimating the room where the image was taken by means of a classification final layer. The second one is addressed by extracting a global descriptor from an intermediate layer of the CNN and used to retrieve the most similar image that conforms the visual map. To address the proposed evaluation, the unique source of information is the set of images obtained by an omnidirectional vision sensor installed on the mobile robot, which moves in an indoor environment under real operation conditions.

3 Methodology

3.1 Hierarchical localization approach

This study aims to tackle visual localization through a hierarchical methodology by means of deep learning. The proposed approach (Fig. 1) consists of two main steps: an initial stage for rough localization, which consist in identifying the room from which the test image has been captured, and a subsequent phase for fine localization where the position of the robot is obtained by a pairwise comparison between the test image and the visual model that conforms the preselected room.

The initial step of rough localization is performed using the output of a CNN. The output layer of that CNN is composed by R neurons, each one corresponding to a room (R is the number of rooms or relevant areas in the target environment). Then, a SoftMax activation function is applied and the room prediction is obtained. However, before training the CNN, a dataset of labelled images captured along the target environment is needed. In this case, each image is labelled with the corresponding room information. The CNN is then trained to address the room retrieval task. Once the CNN is appropriately trained for the room classification task, the coarse localization step is performed: a test image im_{test} is fed into the CNN and the output indicates the room c_i in which the image was captured.

Simultaneously, a holistic descriptor is extracted by flattening the activation map from the last convolutional layer. This descriptor $\mathbf{d}_{test}$ is compared with the descriptors $D_{c_i} = \{\mathbf{d}_{c_i,1}, \mathbf{d}_{c_i,2}, \ldots, \mathbf{d}_{c_i,N_i}\}$ from the visual map of

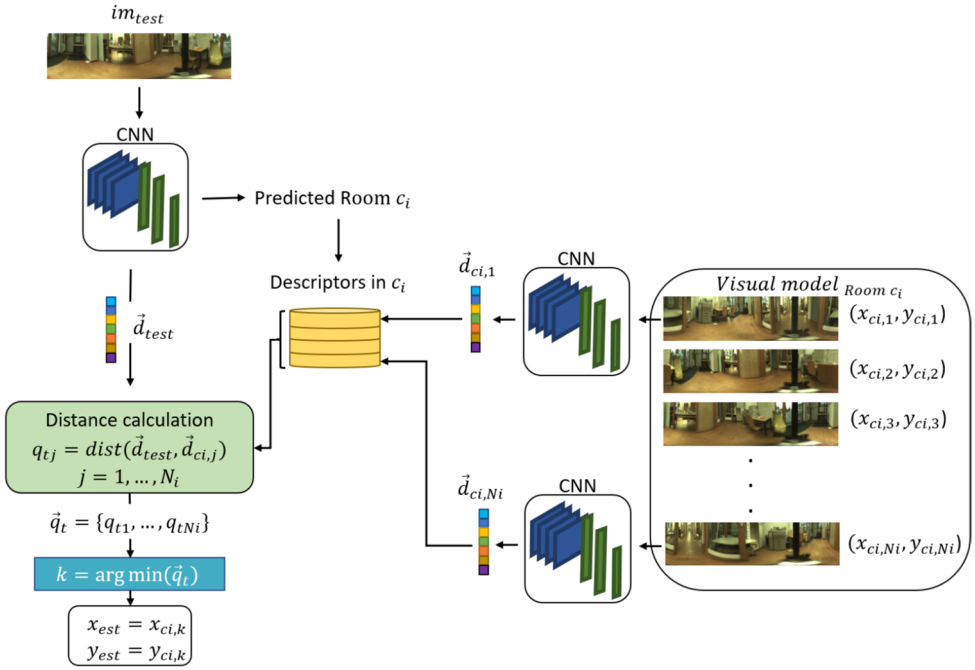


Fig. 1 Diagram of the proposed hierarchical localization. The test image im_{test} is the input of the CNN, which predicts the most likely room c_i and embeds the image into a global descriptor $\mathbf{d}_{test}$ by flattening the last activation map. This descriptor is compared with the descriptors from the training dataset included in the retrieved room by means of a nearest neighbour search. Consequently, the capture point of the image that corresponds to the most similar descriptor ($im_{c_i,k}$) is considered an estimation of the position where im_{test} was captured

the predicted room c_i, where N_i is the number of images in the room c_i. Note that the visual map descriptors are also obtained by flattening the last activation map of the same CNN. Then, the distance between the test descriptor $\mathbf{d}_{test}$ and each descriptor $\mathbf{d}_{c_i,j}\, \epsilon\, D_{c_i}$ in the room c_i is calculated (Eq. 1).

$$q_{t_j} = dist(\mathbf{d}_{test}, \mathbf{d}_{c_i,j}), \quad j = 1, \ldots, N_i \tag{1}$$

where N_i is the number of descriptors in room c_i and *dist* is the Euclidean distance (Eq. 2)

$$dist(\mathbf{d}_{test}, \mathbf{d}_{c_i,j}) = \sqrt{\sum_{i=1}^{m} (d_{test,i} - d_{c_i,j,i})^2} \tag{2}$$

where $\mathbf{d}_{test} = (d_{test,1}, d_{test,2}, \ldots, d_{test,m})$ and $\mathbf{d}_{c_i,j} = (d_{c_i,j,1}, d_{c_i,j,2}, \ldots, d_{c_i,j,m})$ are the descriptors of size m, and $d_{test,i}$ and $d_{c_i,j,i}$ are the i-th components of the vectors $\mathbf{d}_{test}$ and $\mathbf{d}_{c_i,j}$, respectively.

After that, a set $\mathbf{q}_t = \{q_{t1}, \ldots, q_{tN_i}\}$ is constructed with the calculated distances. The index k which minimizes the distance in the set $\mathbf{q}_t$ is found in Eq. 3. Subsequently, the estimated position (x_{est}, y_{est}) corresponds to the position $(x_{c_i,k}, y_{c_i,k})$ from which the image $\mathbf{im}_{c_i,k}$ of the visual map (i.e, the image whose descriptor is the retrieved one $\mathbf{d}_{c_i,k}$) was captured (Eq. 4). This hierarchical approach ensures both a broad understanding of the scene and precise localization within the identified room, contributing to an effective visual localization strategy. Figure 1 outlines the whole localization process.

$$k = \arg\min(\mathbf{q}_t) \tag{3}$$

$$x_{est} = x_{c_i,k}, \quad y_{est} = y_{c_i,k} \tag{4}$$

3.2 CNN selection and adaption

Designing a Convolutional Neural Network to a address a specific task supposes a big challenge. In the present work, the CNN must be able to predict the room in which an image was captured and embed the input image into a global descriptor to retrieve the exact position within the predicted room. Crafting a CNN from scratch demands both a profound understanding of the specificities involved and access to a sufficiently varied dataset for effective training. Furthermore, as previously demonstrated in Ballesta et al. (2021), in general terms, re-training networks that have been designed for a different objective yields more precise and reliable outcomes in the new task than training from scratch.

In light of this information, this research work incorporates several widely recognised and tested CNN models, each of which serves as the backbone for our hierarchical localization task. These models cover a diverse range, addressing different architectural complexities and capabilities. All of the architectures employed were originally designed for visual object recognition. In this work, the CNN is first used to address the room retrieval problem, which is a similar task:

- AlexNet (Krizhevsky et al. 2012): AlexNet is a pioneering CNN architecture known for its success in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2012. Comprising multiple convolutional and

fully connected layers, AlexNet laid the foundation for subsequent CNN designs. This network and the following ones were trained to classify the 1.2 million high-resolution images into 1000 different classes. The weights and biases obtained by training with this database have been taken as starting point for our own task.

- ResNet-152 (He et al. 2016): ResNet, or Residual Network, introduced the concept of residual learning. This approach is based on skip connections and allows the CNN to learn an identity function. ResNet-152 is a specific variant featuring 152 layers, enabling the model to effectively capture intricate hierarchical features. Although it is computationally costly due to its depth, its accuracy and robustness compensate this cost.
- ResNeXt-101 64x4d (Xie et al. 2017): ResNeXt is an extension of the ResNet architecture, emphasizing a cardinality parameter to enhance model capacity. The cardinality is just the number of parallel blocks, that allows to learn various input representations. In this sense, ResNeXt-101 64x4d has a cardinality of 64. By increasing the cardinality, the network can capture a greater diversity of features, enhancing its potential ability to image recognition.
- MobileNetV3 (Howard et al. 2019): MobileNetV3 is designed for efficient mobile and edge computing applications. It uses depth-wise separable convolutions to build light weight deep neural networks. This fact makes them specially suitable for scenarios with resource constraints, such as performing the localization in real time by the robot's on-board computer.
- EfficientNetV2 (Tan and Le 2021): EfficientNetV2 is based on the EfficientNet architecture, and uses a technique called compound coefficient to scale up models in a simple but effective manner. It prioritizes model efficiency, achieving remarkable accuracy with fewer parameters compared to traditional CNNs. This makes EfficientNetV2 an attractive choice for applications requiring high accuracy with limited computational resources.
- ConvNeXt Large (Liu et al. 2022): ConvNeXt Large represents a recent advancement in CNN architectures. It leverages a combination of depth-wise separable convolutions, an inverted bottleneck and spatial factorization ("patchify"), contributing to improved efficiency and effectiveness in capturing features. Thus, outperforming the previous models in terms of accuracy.

By evaluating these diverse CNN models, we aim to comprehensively understand their strengths and weaknesses in the context of scene recognition and localization task. Regarding the room recognition, the final layer of all the architectures needs to be adapted for classifying the images into N categories corresponding to N possible rooms in the target environment ($N = 9$ in the dataset used in the present work, as described in Sect. 4.1). As for the fine localization, the global descriptor has been extracted by flattening the output of the Average Pooling Layer of each CNN model. Finally, Table 1 shows a summary with the evaluated models and its corresponding number of Floating Point Operations (FLOPs) and the number of parameters.

3.3 Data augmentation

Training a model involves setting up its parameters to perform a specific task. When a model has many parameters, it requires a sufficiently large number of examples for effective training. However, in practice, the training dataset is often limited. In such cases, data augmentation is a useful solution as it is able to generate new instances by applying various visual effects. This not only helps the model avoid overfitting but also makes it more robust against challenging real-operation dynamic conditions.

In previous studies focused on training models for visual localization, various effects like changes in orientation, reflections, alterations in illumination, noise, and occlusions were applied (Cabrera et al. 2022). The use of data augmentation has shown to improve model performance. These effects are applied individually or together to each image in the original dataset, and all the generated images are combined into a new augmented training dataset. However, the specific impact of each type of effect on the resulting CNN's performance is not well understood. This study aims to apply different data augmentation effects individually to evaluate their influence on the resulting CNN.

The focus of this work is on two categories of visual effects: changes in illumination conditions and changes in orientation. For changes in illumination conditions, the following effects are considered:

- Spotlights and shadows: Circular light sources, like bulbs, are common indoors. The proposed approach involves increasing pixel values to simulate higher light intensity (spotlights) and decreasing pixel values to sim-

Table 1 FLOPs and parameters of the evaluated and adapted models when the size of the input image is $512 \times 128 \times 3$ pixels

Backbone model	FLOPs (G)	Number of parameters (M)
AlexNet	0.9	57.0
ResNet-152	15.2	58.2
ResNeXt-101 64X4d	20.4	81.4
MobileNetV3	0.3	4.2
EfficientNetV2	16.2	117.2
ConvNeXt Large	44.9	196.2

ulate shadows (shadow spots). Spotlights and shadow spots are applied separately for different data augmentation options. In our experiments, these bulbs are created with diameters ranging from 15 to 40 pixels. Five kinds of intensities variations are applied. In the first type the intensity is degraded ± 160 and in the fifth ± 100.

- General brightness and darkness: Low intensity values of the original images are increased to create brighter images, simulating higher overall illumination (e.g., a sunny day). Conversely, high intensity values are decreased to create darker images, simulating lower light supply (e.g., capturing images at night). Brightness and darkness are applied separately but used for the same data augmentation.
- Contrast: Image contrast plays a vital role in distinguishing objects in a scene. Images with low contrast tend to have a smoother appearance with fewer shadows and reflections. The contrast is modified following Eq. 5

$$I_s = 64 + c * (I - 64) \tag{5}$$

where I_s is the resulting image, I the original image and c is the contrast factor. For $c > 1$ the contrast increases and $c < 1$ decreases the contrast.

- Saturation: Color saturation, indicating the color intensity given by pixels, is considered. Lower saturation results in less colorful images, potentially resembling grayscale images for very low saturation. This phenomenon may occur in real environments and is incorporated into data augmentation. The color saturation can be adjusted by first converting the RGB image to HSV. Then, the saturation channel can be directly modified by multiplying it by a constant factor s. If the saturation is multiplied by $s > 1$, the colors become more saturated, whereas if multiplied by $s < 1$, the saturation decreases.

Regarding changes in orientation, these can occur during image capture when the robot captures images from the same position but with a different orientation. For this data augmentation option, new images are generated for each original image by applying rotations of n degrees, where $n = i \times 10°, i \in [1, 35]$. Thus, for each original image in the training set, 35 additional images are generated.

Figure 2 shows an example of the effects applied to a sample omnidirectional image converted to panoramic format. The first image corresponds to the original one and the rest of images include the different effects presented above (they have been separately applied).

4 Results

4.1 COLD Freiburg database

The current study utilizes images sourced from the Freiburg dataset, a subset of the COsy Localization Database (COLD) (Pronobis and Caputo 2009). This dataset contains omnidirectional images captured by a robot which follows various paths within a building at Freiburg University. The robot explores diverse spaces such as kitchens, corridors, printer areas, bathrooms, personal offices, and more. Image capture occurs under realistic operational conditions, including

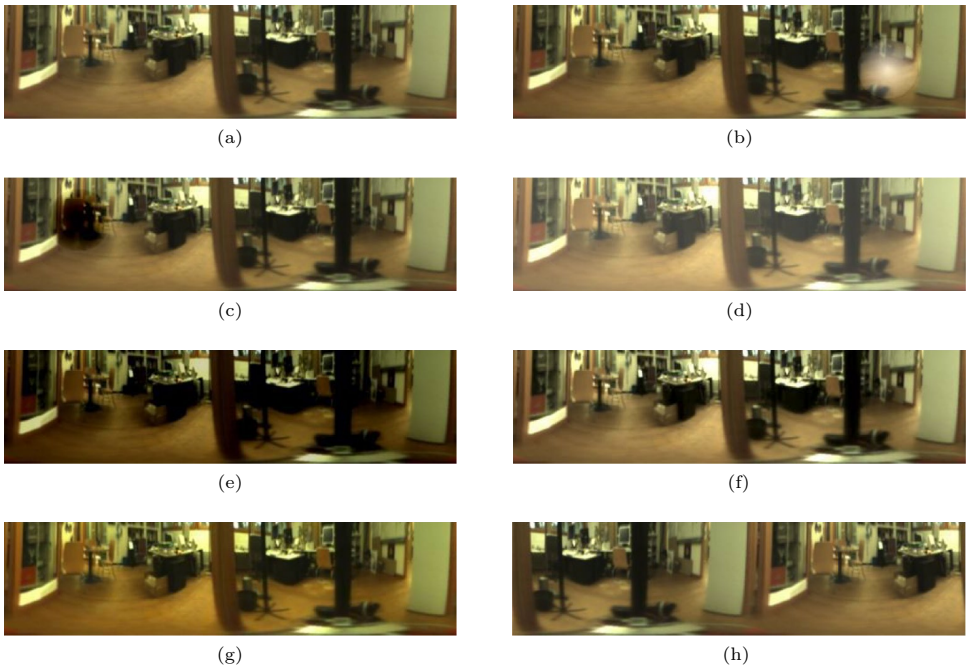


Fig. 2 Example of data augmentation where only one effect is applied over each image. **a** Original image, **b** spotlight effect, **c** shadow effect **d** general brightness, **e** general darkness, **f** contrast, **g** saturation and **h** rotation. The images contained in this dataset can be downloaded from the web site https://www.cas.kth.se/COLD/

changes in furniture arrangement, the dynamic presence of individuals in scenes, and fluctuations in illumination conditions, including cloudy days, sunny days, and nights.

To assess the impact of these variations on the localization task, we propose incorporating images taken exclusively on cloudy days as part of the training data. Additionally, a separate dataset comprising cloudy images (distinct from the aforementioned one) is employed as test set to evaluate localization performance without illumination changes. Furthermore, to appraise localization under varying illumination conditions, datasets captured on sunny days and at night are utilized as test sets. Beyond the images, the dataset offers ground truth data (obtained via a laser sensor), which is exclusively employed in this study to quantify localization errors. The ground truth over the path of the robot has been generated using the laser sensor in a grid-based SLAM technique, in particular, the one described in Grisetti et al. (2005, 2007). This solution, based on these two papers, can have an error up to 5 cm or 10 cm depending on the grid resolution.

Concerning the image capture process, the robot acquires images while it moves, introducing potential blur effects or dynamic alterations. Moreover, the chosen environment has the longest trajectory within the available database and is characterized by extensive windows and glass walls, making visual localization a particularly challenging problem. Consequently, this environment provides ideal conditions for evaluating the proposed localization methods under real operation conditions and real scenarios.

The selected dataset contains images from nine distinct rooms: a kitchen, a bathroom, a printer area, a stairwell, a long corridor and four offices. The cloudy dataset is downsampled to achieve an average distance of 20 cm between consecutive image capture points, resulting in the Baseline Training Dataset comprising 556 images. This dataset serves the dual purpose of training the CNNs and providing a visual map. In addition, a Validation Dataset is used during training and keeps the same proportion of images as the Baseline Training set. The Validation Dataset is also sampled at 20-cm intervals, but in this case in an interleaved manner with respect to the Baseline Training Dataset in such a way that the images in the baseline and validation datasets are different. In this regard, the validation covers uniformly the whole environment, which is expected to be a robust approach for validation, considering that the retrained CNN must be able to solve the localization problem considering the whole environment. Furthermore, the Baseline Training Dataset undergoes a data augmentation as described in Sect. 3.3, resulting in six additional training datasets. These datasets will be individually employed to train the CNNs, allowing an exploration of the impact of each visual effect on network performance. Table 2 shows a summary with the number of images per room of each training and validation dataset.

In terms of the test data, various datasets are considered: Cloudy Test Dataset, comprising images captured in cloudy conditions along a route distinct from training and validation sets (2595 images); Sunny Test Dataset, including all images captured in sunny conditions (2114 images); and Night Test Dataset, containing all images captured at night (2707 images). Table 3 shows a summary with the number of images per room of each test set. Consequently, network training and validation, in all instances, employs images captured exclusively in cloudy conditions, while testing occurs under three distinct lighting conditions: cloudy, sunny, or night. This methodology enables the assessment of the network's robustness against variations in lighting conditions.

Table 2 Number of images in each training dataset (number of images per room)

Training dataset	1P0-A	2P01-A	2P02-A	CR-A	KT-A	LO-A	PA-A	ST-A	TL-A
Baseline	44	46	31	238	46	26	57	30	38
Validation	43	47	32	236	46	26	57	31	38
Augmented 1	264	276	186	1428	276	156	342	180	228
Augmented 2	264	276	186	1428	276	156	342	180	228
Augmented 3	308	322	217	1666	322	182	399	210	266
Augmented 4	264	276	186	1428	276	156	342	180	228
Augmented 5	264	276	186	1428	276	156	342	180	228
Augmented 6	1364	1426	961	7378	1426	806	1767	930	1178

Table 3 Number of images in each test dataset (number of images per room)

Test dataset	1P0-A	2P01-A	2P02-A	CR-A	KT-A	LO-A	PA-A	ST-A	TL-A
Cloudy	155	230	135	1040	254	177	222	133	249
Night	168	215	168	1114	270	121	241	198	212
Sunny	123	187	109	793	213	102	191	180	216

4.2 Implementation details

In this work, the CNNs are trained to address the coarse localization or room retrieval stage. As this is a classification task, these networks have been retrained employing a Cross Entropy loss function (Eq. 6).

$$\mathcal{L}(y,\hat{y}) = -\frac{1}{B}\sum_{i=1}^{B}\sum_{j=1}^{R} y_{ij}\log(\hat{y}_{ij}) \tag{6}$$

where y is the matrix of actual labels and $\hat{y}$ is the matrix of model predictions, both matrices have size $B \times R$, in which B is the number of samples (batch size) and R is the number of classes (rooms), y_{ij} is 1 if sample i belongs to class j and 0 otherwise, and $\hat{y}_{ij}$ is the probability predicted by the model that sample i belongs to class j.

In addition, Stochastic Gradient Descent (SGD) with Momentum 0.9 and Learning Rate of 0.001 has been used as optimization algorithm. Furthermore, the training batch size (B) was 16 and the total number of epochs was 30. For every architecture, the network that presents the best validation accuracy for room retrieval during the training is preserved for testing. Table 4 summarizes all the values of the parameters that have been described above.

All experiments are carried out with a NVIDIA GeForce RTX 3090 GPU with 24 GB. Our code is publicly available on the project website https://github.com/juanjo-cabrera/IndoorLocalizationSingleCNN.git.

4.3 CNN backbone ablation study

In this section, we asses an experimental evaluation of the different CNN models used as backbone presented in Sect. 3.2 for both rough and fine localization. As previously stated, the hierarchical localization proposed in this study comprises two distinct steps. The initial stage, rough localization step, involves retraining a model to execute the room retrieval task. Subsequently, the fine localization step utilizes the previously trained CNN to generate holistic descriptors, employing a nearest neighbor search method to estimate the precise position where an image was captured.

Table 4 Training parameters for room retrieval

Parameter	Value
Batch size (B)	16
Number of epochs	30
Learning rate	1×10^{-3}
Momentum	0.9
Number of rooms (R)	9

4.3.1 Coarse localization: room retrieval

This section presents the results derived from the use of different CNNs for the execution of the coarse localization or room retrieval stage. As described in Sect. 3.2, the CNN models evaluated in this article are AlexNet (Krizhevsky et al. 2012), ResNet-152 (He et al. 2016), ResNeXt-101 64x4d (Xie et al. 2017), MobileNetV3 (Howard et al. 2019), EfficientNetV2 (Tan and Le 2021) and ConvNeXt Large (Liu et al. 2022). The reason why we have selected these models is to cover a wide range of architectures proposed for image classification in the last 10 years.

The results in Table 5 showcase the performance of six different models used as backbone in the context of room retrieval across varied environmental conditions. In fact, each model was subjected to evaluation under cloudy, night, and sunny conditions, providing a comprehensive understanding of their robustness and adaptability to changes in environment illumination.

AlexNet exhibits an excellent overall performance, particularly in Cloudy conditions with an accuracy of 97.61%. In contrast, ResNet demonstrates robust performance but slightly lower accuracy compared to AlexNet. Notably, its accuracy decreases in sunny conditions which is the most demanding illumination environment. The ResNext model excels in cloudy conditions. However, it shows a comparatively lower accuracy in night scenarios. On the one hand, MobileNet stands out for its consistency, achieving high accuracy across all conditions. Its notable performance in sunny conditions, with an accuracy of 77.29%, highlights its generalisation capability. On the other hand, EfficientNet emerges as a top-performing model, outperforming others in terms of accuracy in cloudy and night scenarios, which are the most similar to training conditions. Finally, the most striking result comes from ConvNext, which consistently achieves the highest accuracy in all scenarios, making it the top-performing model. Particularly noteworthy is its

Table 5 Room retrieval ablation study for different top-level classification architectures tested under three different illumination conditions: cloudy, night, sunny and all together

Backbone model	Room retrieval accuracy (%)			
	Cloudy	Night	Sunny	Global
AlexNet	97.61	97.60	70.67	89.93
ResNet-152	96.76	96.64	64.95	87.63
ResNeXt-101 64X4d	98.11	95.16	72.47	89.71
MobileNetV3	98.50	96.93	77.29	91.88
EfficientNetV2	**98.81**	97.16	75.73	91.63
ConvNeXt Large	98.77	**97.64**	**86.28**	**94.80**

Bold values represent the best accuracy for every lighting condition

exceptional accuracy of 86.28% in sunny conditions, indicating its robustness and generalization capabilities.

4.3.2 Fine localization

Once the CNN model is trained for the room retrieval step, it can be used to embed the input image into a global descripor. This facilitates the resolution of the fine localization step through an image retrieval process, in which the descriptor of the test image is compared with the descriptors of the visual map of the previously retrieved room. As in previous subsection, we are going to evaluate the performance of different CNN backbones to address the fine localization step. Fig. 3 shows the hierarchical localization error for different backbone models (AlexNet, ResNet-152, ResNeXt-101, MobileNetV3, EfficientNetV2 and ConvNeXt Large) under various lighting conditions (cloudy, night, sunny) and considering jointly the three conditions (global). The errors are measured in meters and are represented by box plots with whiskers, indicating the distribution of the errors. Furthermore, the Mean Absolute Error (Eq. 7) is represented by the black dot and the text displaying the error value. In addition, Table 6 shows the computation time required to execute the whole hierarchical localization process for all the evaluated models.

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}\left|(x_i, y_i) - (\hat{x}_i, \hat{y}_i)\right| \tag{7}$$

where (x_i, y_i) is the actual position, $(\hat{x}_i, \hat{y}_i)$ is the position of the visual map retrieved after the complete localization process, and N is the number of images in the test dataset.

Each backbone model exhibited similar characteristics in hierarchical localization comparing to room retrieval, since both tasks are correlated. As Fig. 3 shows, AlexNet demonstrated a consistent localization error and low dispersion for cloudy and night conditions. However, its performance degraded in sunny conditions. ResNet-152 displayed higher errors across all conditions compared to AlexNet, with a notable increase of both the mean absolute error and dispersion in sunny conditions. ResNeXt-101 demonstrated a better performance than ResNet-152 for cloudy and sunny conditions, but the error slightly increases for night scenarios. MobileNet consistently maintained low errors across all conditions, signifying its adaptability to diverse lighting environments. EfficientNet showcased a worse performance than MobileNet in each scenario. Finally, ConvNeXt emerged as the top-performing model, consistently outperforming others with the lowest errors across all conditions. Its remarkable accuracy in sunny conditions implies a robust capability to handle scenarios with substantial changes of the lighting conditions. In terms of computation time, Table 6 illustrates that the hierarchical localization process with the shortest average computation time occurs when employing AlexNet, which requires only 3.4 ms. In contrast, the hierarchical localization process employing ConvNeXt Large requires the longest computation time, with a mean of 12.5 ms. However, despite the need for more time to estimate the

Table 6 Computation time required to execute the whole hierarchical localization process for all the evaluated models

Backbone model	Mean time (ms)
AlexNet	3.4
ResNet-152	6.9
ResNeXt-101 64X4d	9.5
MobileNetV3	4.6
EfficientNetV2	10.7
ConvNeXt Large	12.5

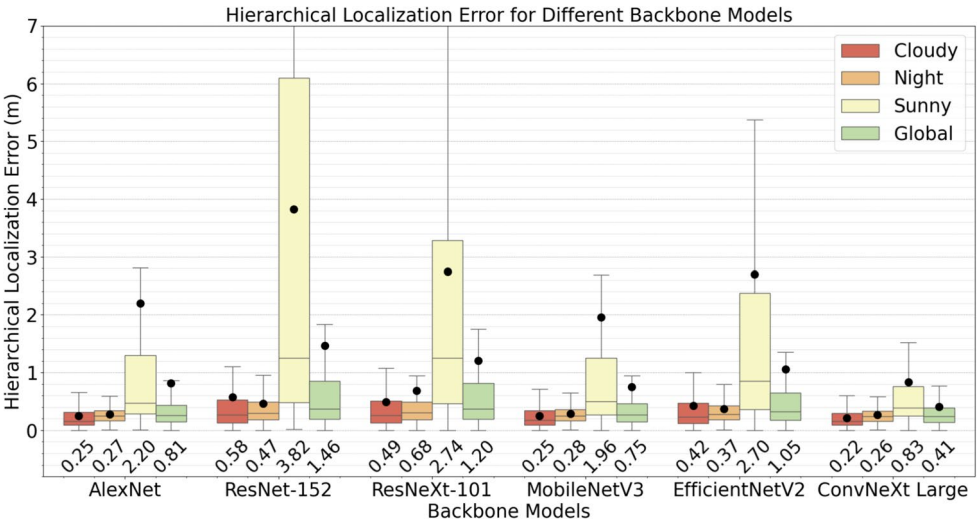


Fig. 3 Hierarchical localization errors in meters for different CNN architectures. The box plots represent the distribution of errors, with whiskers indicating variability. The Mean Absolute Error for each model and condition is marked by a black dot and annotated with the specific error value. Results are obtained under different lighting conditions: cloudy (red), night (orange), sunny (yellow) and considering jointly the three conditions (green)

position, this time is sufficiently short to enable real-time localization.

4.4 Data augmentation ablation study

In this comprehensive experiment, the investigation is extended to evaluate the influence of both data augmentation effects (illumination and orientation changes) on the performance of the CNN. Due to the existence of a high probability of variations in robot orientation during operation under real operation conditions with respect to the images captured in the visual map, a model should demonstrate robustness to orientation changes. To this end, a data augmentation technique is employed that consists in applying 35 different orientation changes to each training image as described in Sect. 3.3. This augmentation is essential to improve the adaptability of the model to the various orientations encountered in practice.

Simultaneously, the illumination effects that occur under real operating conditions, a critical aspect for robust visual perception, have been explored in detail. Five specific lighting effects are considered (Sect. 3.3): spotlights, shadow spots, general brightness/darkness, contrast, and saturation. Each effect is systematically applied individually on the training data set, leading to the creation of distinct augmented training datasets. Using the different effects separately allows a detailed understanding of their individual contributions, which sheds light on the importance of each effect in performance.

In particular, for each image, the experiment incorporates a detailed approach by applying different levels of spotlights, contrast and saturation (five levels for each), ensuring a thorough assessment of the impact of these factors on the ability of the CNN to adapt to various lighting conditions. In addition, the effect of brightness is meticulously explored, with three levels of brightness and three levels of darkness applied to each image. This dual investigation of orientation changes and illumination effects is intended to provide a comprehensive understanding of the robustness of the CNN to cope with real-world challenges, encompassing variations in both spatial orientation and illumination conditions. As a result of applying these effects, six additional training datasets have been obtained: Augmented Training Dataset 1 (spotlights), Augmented Training Dataset 2 (shadows), Augmented Training Dataset 3 (general brightness/darkness), Augmented Training Dataset 4 (contrast), Augmented Training Dataset 5 (saturation) and Augmented Training Dataset 6 (rotations). Augmented Training Datasets 1, 2, 4 and 5 consist of 3336 images each, whereas Augmented Training Datasets 3 and 6 includes 3892 and 17,236 images respectively.

In conclusion, in this ablation study the model is retrained using separately each of the Augmented Training Datasets 1, 2, 3, 4, 5 and 6. As in previous experiments, the Baseline Training Dataset serves as a visual map and the Validation Dataset is employed to validate the performance of the CNN. Furthermore, for the model evaluation, three different test datasets are considered: the Cloudy Test Dataset, the Night Test Dataset and the Sunny Test Dataset.

4.4.1 Coarse localization: room retrieval

In this subsection we use the best CNN architecture obtained in Sect. 4.3.1, which is ConvNeXt Large. In a similar approach, we have departed from the pre-trained weights for ImageNet Large Scale Visual Recognition Challenge and re-trained the model for the different datasets obtained by the proposed data augmentation.

Table 7 presents the room retrieval accuracy when the model has been trained with each of the augmented training datasets previously described. The performance of the CNN is evaluated under the three different lighting conditions: cloudy, night, sunny and all together.

Training with the baseline dataset shows a remarkable accuracy, especially in cloudy and night conditions. However, a significant decrease is observed in sunny conditions, which differ more from the training set. This evaluation provides a reference to analyse the impact of the different effects that have been applied to the training data.

The spotlight augmentation (Augmentation 1) shows insignificant improvements or even small decreases under night and sunny conditions. In contrast, data augmentation with shadows (Augmentation 2) produces slight improvements, especially in sunny conditions.

Alterations to the overall brightness and darkness of the image (Augmentation 3) are effective and show substantial improvements, especially in sunny conditions. In addition, contrast-based effects (Augmentation 4) are very effective, with substantial improvements in all lighting conditions and

Table 7 Room retrieval accuracy for ConvNeXt Large architecture with different augmented training datasets

Training dataset	Room retrieval accuracy (%)			
	Cloudy	Night	Sunny	Global
Baseline	98.77	**97.64**	86.28	94.80
Augmented 1 (spotlights)	98.84	97.45	86.14	94.71
Augmented 2 (shadows)	98.96	97.56	86.52	94.90
Augmented 3 (brightness/darkness)	98.81	97.41	91.11	96.10
Augmented 4 (contrast)	99.08	97.27	**93.57**	**96.84**
Augmented 5 (saturation)	98.88	97.60	83.07	93.91
Augmented 6 (rotations)	**99.15**	97.52	91.39	96.34

Bold values represent the best accuracy for every lighting condition

especially in sunny circumstances, thus achieving improved results in this challenging environment.

Surprisingly, augmentation with changes in saturation (Augmented 5) shows a negative impact on accuracy, especially in sunny conditions. Finally, augmenting the data set with rotations (Augmented 6) shows substantial improvements, especially in cloudy conditions.

4.4.2 Fine localization

Once the ConvNeXt Large model is trained for the room retrieval step, it can be used to embed the input image into a global descriptor. This facilitates the resolution of the fine localization step through an image retrieval process, wherein the descriptor of the test image is compared with the descriptors of the visual map. As in previous subsection, we are going to evaluate the performance of different data augmentation effects to address the fine localization step.

As shown in Fig. 4, training with every augmented dataset result in similar network performance under cloudy illumination conditions for the fine localization task, achieving a mean absolute error around 0.22 ms. The same happens under the night condition, in which the mean absolute error is around 0.27 ms. In this case, the minimum error is obtained by training the network without data augmentation.

In contrast, under sunny lighting conditions the mean localization error has a higher variability, similarly to the coarse localization (Table 7). This demonstrates the correlation between the two tasks. Under this condition, the best fine localization result is obtained by training the model with the contrast effect (DA 4) and the worst with saturation (DA 5).

4.4.3 General comparison with other methods

Finally, the proposed method is compared with other previous global appearance techniques, including the use of single CNN structures (Cabrera et al. 2022; Rostkowska and Skrzypczynski 2023), triplet structures (Alfaro et al. 2024) and two classical analytical descriptors: HOG and gist, as described in Cebollada et al. (2022). Both HOG and gist are only taken into consideration when testing with night and sunny conditions, since the conditions of the cloudy test experiment in Cebollada et al. (2022) are different to the conditions in the present work. Table 8 compares all the methods in a global localization task, using in all cases the COLD-Freiburg dataset, which is the same dataset used in

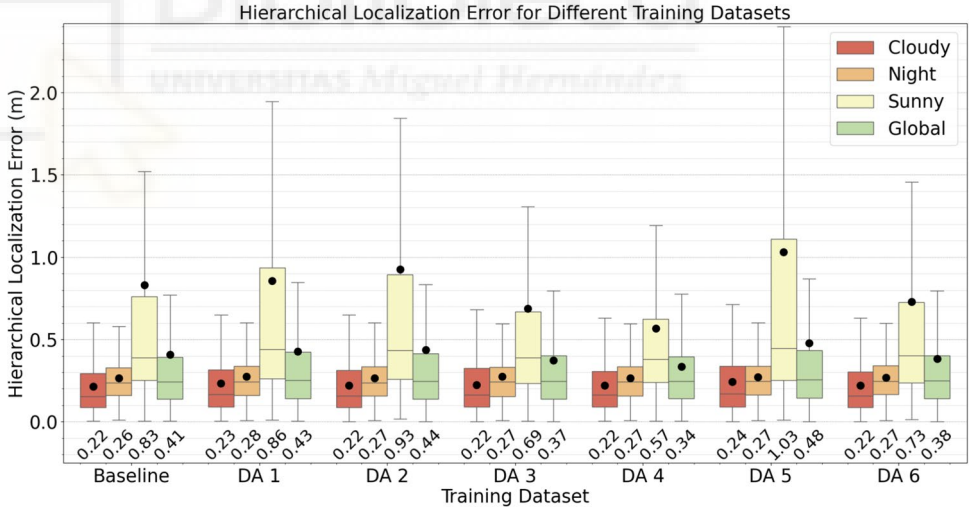


Fig. 4 Hierarchical localization errors in meters when training the ConvNeXt Large architecture with different data augmentation effects. The box plots represent the distribution of errors, with whiskers indicating variability. The Mean Absolute Error for each model and condition is marked by a black dot and annotated with the specific error value. Results are obtained under different lighting conditions: cloudy (red), night (orange), sunny (yellow) and considering jointly the three conditions (green)

Table 8 Comparison with other methods

Global-appearance descriptor technique	Cloudy error (m)	Night error (m)	Sunny error (m)
Alexnet + DA (Cabrera et al. 2022)	0.29	0.29	0.69
EfficientNet (Rostkowska and Skrzypczynski 2023)	0.24	0.33	0.44
Triplet VGG16 (Alfaro et al. 2024)	0.25	0.28	**0.40**
ConvNeXt Large (ours)	**0.22**	**0.26**	0.83
ConvNeXt Large + DA (ours)	**0.22**	0.27	0.57
HOG (Cebollada et al. 2022)	–	0.45	0.82
gist (Cebollada et al. 2022)	–	1.07	0.88

Bold values represent the minimum error for every lighting condition

the previous subsections. This table shows that ConvNeXt Large without data augmentation provides the best results in terms of localization error for cloudy and night conditions. Training with data augmentation does not improve the performance in cloudy conditions. However, it favours the results under sunny conditions. In this illumination condition, the best result is obtained with a triplet VGG16 proposed in Alfaro et al. (2024).

5 Conclusion

This study assesses the application of a deep learning technique in addressing hierarchical localization using omnidirectional imaging. The technique involves training a CNN to perform room retrieval, addressed as an image classification problem. Additionally, the CNN is employed to embed the input image into a holistic descriptor from intermediate layers, aggregating relevant information that characterizes the input image. Additionally, we evaluate the influence of two main components on the localization performance: CNN architecture and effects applied in the data augmentation.

As for the CNN backbone, AlexNet shows excellent overall performance, especially when tested under the same lighting conditions than the training images. In contrast, ResNet performance decreases in sunny conditions which are the most challenging test conditions. This fact shows its low capability of generalization. The ResNext model surpass both in cloudy and sunny conditions, showcasing versatility across different lighting environments. However, EfficientNet exhibits a slight advantage over the ResNext model in terms of accuracy, although it requires more computational time. Furthermore, MobileNet consistently produces accurate results with a competitive computational time, demonstrating high performance across all conditions. Finally, the most striking result comes from ConvNext, which consistently achieves the highest accuracy in all scenarios, making it the top-performing model. Particularly noteworthy is its exceptional accuracy in sunny conditions, indicating its robustness and generalization capabilities.

Regarding the proposed data augmentation, training with the baseline dataset yields a remarkable accuracy, especially in cloudy and night conditions. However, a significant decrease is observed in sunny conditions, which diverge more from the training dataset. The spotlight effect shows marginal improvements, indicating that spotlight-based enhancement does not contribute to improve the generalization ability of the network. In contrast, data augmentation with shadows produces moderate improvements, especially in sunny conditions. Changing the overall brightness and darkness of the image produces substantial improvements, especially in sunny conditions. In addition, contrast-based effects are very effective, with significant improvements in all lighting conditions and especially in sunny conditions, improving results in this tough environment. Surprisingly, augmenting the dataset with changes in saturation shows a negative impact, especially in sunny conditions. Finally, increasing the dataset with rotations results in significant improvements in cloudy conditions. Finally, as for sunny conditions, the contrast effect yields the most optimal results, thereby enhancing the model's generalization capabilities and preventing overfitting.

In future works, studying more advanced techniques for generating more realistic visual effects with Generative Adversarial Networks (GANs) is a priority. Furthermore, we will evaluate other deep learning schemas such as Siamese, Triplet Neural Networks and Feature Pyramid Networks (FPNs). Finally, we will approach the localization problem in outdoor environments by using CNNs, considering the specificities of such scenarios.

Acknowledgements The Ministry of Science, Innovation and Universities (Spain) has supported this work through "Ayudas para la Formación de Profesorado Universitario" (FPU21/04969). This work is also part of the project TED2021-130901B-I00, funded by MCIN/AEI/10.13039501100011033 and the European Union "NextGenerationEU"/PRTR, and of the project PROMETEO/2021/075 funded by Generalitat Valenciana.

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature.

Data availability Data is available in the github repo provided in Code availability.

Code availability Our code is publicly available on the project website https://github.com/juanjo-cabrera/IndoorLocalizationSingleCNN.git.

References

Aguilar WG, Luna MA, Moya JF, Abad V, Parra H, Ruiz H (2017) Pedestrian detection for UAVs using cascade classifiers with meanshift. In: 2017 IEEE 11th international conference on semantic computing (ICSC). IEEE, pp 509–514

Alfaro M, Cabrera JJ, Jiménez LM, Reinoso Payá L (2024) Hierarchical localization with panoramic views and triplet loss functions. arXiv preprint. arXiv:2404.14117

Bai D, Wang C, Zhang B, Yi X, Yang X (2018) CNN feature boosted SeqSLAM for real-time loop closure detection. Chin J Electron 27(3):488–499

Ballesta M, Payá L, Cebollada S, Reinoso O, Murcia F (2021) A cnn regression approach to mobile robot localization using omnidirectional images. Appl Sci 11(16):7521

Cabrera JJ, Cebollada S, Flores M, Reinoso Ó, Payá L (2022) Training, optimization and validation of a CNN for room retrieval and description of omnidirectional images. SN Comput Sci 3(4):1–13

Cebollada S, Payá L, Jiang X, Reinoso O (2022) Development and use of a convolutional neural network for hierarchical appearance-based localization. Artif Intell Rev 55(4):2847–2874

Céspedes OJ, Cebollada S, Cabrera JJ, Reinoso O, Payá L (2023) Analysis of data augmentation techniques for mobile robots localization by means of convolutional neural networks. In: IFIP international conference on artificial intelligence applications and innovations. Springer, pp 503–514

Chen Z, Lam O, Jacobson A, Milford M (2014) Convolutional neural network-based place recognition. arXiv preprint. arXiv:1411.1509

Ding J, Chen B, Liu H, Huang M (2016) Convolutional neural network with data augmentation for SAR target recognition. IEEE Geosci Remote Sens Lett 13(3):364–368

Grisetti G, Stachniss C, Burgard W (2005) Improving grid-based slam with rao-blackwellized particle filters by adaptive proposals and selective resampling. In: Proceedings of the 2005 IEEE international conference on robotics and automation. IEEE, pp 2432–2437

Grisetti G, Stachniss C, Burgard W (2007) Improved techniques for grid mapping with rao-blackwellized particle filters. IEEE Trans Robot 23(1):34–46

He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778

Howard A, Sandler M, Chu G, Chen L-C, Chen B, Tan M, Wang W, Zhu Y, Pang R, Vasudevan V (2019) Searching for mobilenetv3. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 1314–1324

Kim Y (2014) Convolutional neural networks for sentence classification. arXiv preprint. arXiv:1408.5882

Komorowski J, Wysoczańska M, Trzcinski T (2021) Minkloc++: lidar and monocular image fusion for place recognition. In: 2021 international joint conference on neural networks (IJCNN). IEEE, pp 1–8

Kopitkov D, Indelman V (2018) Bayesian Information Recovery from CNN for probabilistic inference. In: 2018 IEEE/RSJ international conference on intelligent robots and systems (IROS), pp 7795–7802 . https://doi.org/10.1109/IROS.2018.8594506

Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems, p 25

LeCun Y, Bengio Y (1995) Convolutional networks for images, speech, and time series. The handbook of brain theory and neural networks. MIT Press, Cambridge

Liu Z, Mao H, Wu C-Y, Feichtenhofer C, Darrell T, Xie S (2022) A convnet for the 2020s. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 11976–11986

Milford MJ, Wyeth GF (2012) Seqslam: visual route-based navigation for sunny summer days and stormy winter nights. In: 2012 IEEE international conference on robotics and automation. IEEE, pp 1643–1649

Naseer T, Ruhnke M, Stachniss C, Spinello L, Burgard W (2015) Robust visual SLAM across seasons. In: 2015 IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE, pp 2529–2535

Perez L, Wang J (2017) The effectiveness of data augmentation in image classification using deep learning. arXiv preprint. arXiv:1712.04621

Pronobis A, Caputo B (2009) COLD: COsy localization database. Int J Robot Res 28(5):588–594. https://doi.org/10.1177/0278364909103912

Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 779–788

Ren S, He K, Girshick R, Sun J (2015) Faster r-cnn: towards real-time object detection with region proposal networks. In: Advances in neural information processing systems, p 28

Ronneberger O, Fischer P, Brox T (2015) U-net: convolutional networks for biomedical image segmentation. In: Medical image computing and computer-assisted intervention—MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18. Springer, pp 234–241

Rostkowska M, Skrzypczynski P (2023) Optimizing appearance-based localization with catadioptric cameras: small-footprint models for real-time inference on edge devices. Sensors 23(14):6485. https://doi.org/10.3390/s23146485

Salamon J, Bello JP (2017) Deep convolutional neural networks and data augmentation for environmental sound classification. IEEE Signal Process Lett 24(3):279–283. https://doi.org/10.1109/LSP.2017.2657381

Sarlin P, Cadena C, Siegwart R, Dymczyk M (2019) From coarse to fine: robust hierarchical localization at large scale. In: 2019 IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 12708–12717 . https://doi.org/10.1109/CVPR.2019.01300

Sermanet P, Eigen D, Zhang X, Mathieu M, Fergus R, LeCun Y (2013) Overfeat: integrated recognition, localization and detection using convolutional networks. arXiv preprint. arXiv:1312.6229

Shorten C, Khoshgoftaar TM (2019) A survey on image data augmentation for deep learning. J Big Data 6(1):60

Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv preprint. arXiv:1409.1556

Sünderhauf N, Shirazi S, Dayoub F, Upcroft B, Milford M (2015) On the performance of convnet features for place recognition. In: 2015 IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE, pp 4297–4304

Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1–9

Tan M, Le Q (2021) Efficientnetv2: smaller models and faster training. In: International conference on machine learning. PMLR, pp 10096–10106

Uy MA, Lee GH (2018) Pointnetvlad: deep point cloud based retrieval for large-scale place recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4470–4479

Wang H, Yang W, Huang W, Lin Z, Tang Y (2018) Multi-feature fusion for deep reinforcement learning: sequential control of mobile robots. In: International conference on neural information processing. Springer, pp 303–315

Xie S, Girshick R, Dollár P, Tu Z, He K (2017) Aggregated residual transformations for deep neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1492–1500

Zhou B, Lapedriza A, Xiao J, Torralba A, Oliva A (2014) Learning deep features for scene recognition using places database. In: Advances in neural information processing systems, pp 487–495

Artificial Intelligence Review (2024) 57:198
https://doi.org/10.1007/s10462-024-10840-0

An experimental evaluation of Siamese Neural Networks for robot localization using omnidirectional imaging in indoor environments

Juan José Cabrera[1] · **Vicente Román**[1] · **Arturo Gil**[1] · **Oscar Reinoso**[1,2] · **Luis Payá**[1]

Published online: 8 July 2024

Abstract
The objective of this paper is to address the localization problem using omnidirectional images captured by a catadioptric vision system mounted on the robot. For this purpose, we explore the potential of Siamese Neural Networks for modeling indoor environments using panoramic images as the unique source of information. Siamese Neural Networks are characterized by their ability to generate a similarity function between two input data, in this case, between two panoramic images. In this study, Siamese Neural Networks composed of two Convolutional Neural Networks (CNNs) are used. The output of each CNN is a descriptor which is used to characterize each image. The dissimilarity of the images is computed by measuring the distance between these descriptors. This fact makes Siamese Neural Networks particularly suitable to perform image retrieval tasks. First, we evaluate an initial task strongly related to localization that consists in detecting whether two images have been captured in the same or in different rooms. Next, we assess Siamese Neural Networks in the context of a global localization problem. The results outperform previous techniques for solving the localization task using the COLD-Freiburg dataset, in a variety of lighting conditions, specially when using images captured in cloudy and night conditions.

Keywords Localization · Omnidirectional imaging · Holistic description · Mobile robots · Siamese Neural Network

1 Introduction

During the past few years, vision sensors have been used extensively in the field of map building and localization with mobile robots (Hu et al. 2020; Zhong et al. 2018). In particular, the ability to localize in the map is of paramount importance in order to develop

✉ Juan José Cabrera
juan.cabreram@umh.es

[1] Institute for Engineering Research (I3E), Miguel Hernández University, Elche, Spain

[2] Valencian Graduate School and Research Network for Artificial Intelligence (valgrAI), Valencia, Spain

autonomous robots that can navigate in real operating conditions. The interest in using vision sensors to capture information from the environment is still high. Cameras can capture a big amount of information from the environment with a relatively low cost and they can be used in both, indoor and outdoor areas. Additionally, the images permit carrying out other highly specialized tasks such as object recognition and people detection.

Among the available configurations to capture visual information, the use of omnidirectional vision sensors in mobile robotics has become common. Omnidirectional cameras obtain images that cover a field of view of 360° around the robot (Junior et al. 2016). As a result, they are commonly used to address navigation tasks (Rituerto et al. 2010).

The large amount of information provided by cameras requires robust techniques to extract and describe the relevant visual information. Different paradigms have been considered to extract this relevant information. A first group of techniques concentrate on detecting, describing and tracking some landmarks or local features along the scenes (Cao et al. 2020; Lin et al. 2020). Different local features have been used in mapping and localization tasks, including SIFT, SURF and ORB descriptors (E. Rublee and Bradski 2011). A global description of each image can then be obtained, for example, by means of the Bag of Words model (Raúl Mur-Artal and Tardós 2015). A second group of techniques work with each scene as a whole, and build a unique descriptor per image that contains information on its global appearance (Korrapati and Mezouar 2017; Khaliq et al. 2019). Finally, hardware developments have led many authors to use Artificial Intelligence (AI) techniques to extract relevant information from images. Specifically, Convolutional Neural Networks (CNNs) have been proposed to address different computer vision and robotics tasks. For example, Xu et al. (2019) and Leyva-Vallina et al. (2019) proposed global appearance descriptors based on a CNN to obtain the most probable pose of the robot.

In general terms, holistic description methods lead to maps in which a set of robot poses and their associated descriptors are stored. In this way, each pose of the robot is represented by a holistic descriptor and this representation leads to straightforward localization algorithms, based on the pairwise comparison between descriptors.

In this manuscript we assess the usage of Siamese Neural Networks in the context of image description and robot localization. Siamese Neural Networks permit evaluating two images at the same time in such a way that they provide a similarity measurement at the output. Therefore, they have the potential to address visual recognition of places and estimate the position of a mobile robot. In the present paper, we evaluate this potential. The main contributions of this paper can be summarized as follows.

1. We explore the capability of Siamese Neural Networks for modeling indoor environments, using panoramic images as the unique source of information.
2. We train and evaluate Siamese Neural Networks with the purpose of detecting whether two images have been captured in the same or in different rooms.
3. We train Siamese Neural Networks capable of estimating robot position as a global image retrieval problem.
4. We conduct an exhaustive study on the influence of the Siamese Neural Networks' architecture and the most relevant parameters. Moreover, we analyze the robustness against some common visual phenomena that may occur in real operating conditions, such as changes of the lighting conditions or image blur.

The following sections are structured as follows. First, in Sect. 2 we present a review of the state of the art on visual localization and mapping using Artificial Intelligence techniques.

Second, in Sect. 3 we introduce Siamese Neural Networks for both room discrimination and global localization. After that, Sect. 4 presents the CNN architectures, the dataset and the proposed data augmentation. Furthermore, in this section we also describe the proposed method for room discrimination and global localization by means of Siamese Neural Networks. Then, Sect. 5 describes the experiments carried out to test and validate the proposed method. Finally, conclusions and future works are outlined in Sect. 6.

2 State of the art

As stated before, Siamese Neural Networks are able to generate a similarity function from pairs of input data. They can be regarded as a superstructure that includes two Neural Networks. These architectures accept two different inputs and offer a single output. The underlying networks share the same weights and different functions can be used to conform a single output. They were first proposed in 1993 in order to distinguish correct signatures from forgeries (Bromley et al. 1993). Since then, these architectures have been proposed in different areas of knowledge. For example, Thiolliere et al. (2015) proposed a Siamese Neural Network for audio and speech signal processing, Zheng et al. (2019) used this architecture for the comparison of DNA sequences or Jeon et al. (2019) used it for drug discovery purposes. Furthermore, Parajuli et al. (2017) developed a Siamese Neural Network to track cardiac motion and Sandouk and Chen (2017) proposed a Siamese architecture in order to recognize music tags. Recently, Suljagic et al. (2022) use this kind of architecture for multi-object tracking (MOT) and person re-identification.

During the past few years, AI in general and CNNs in particular have been used in the field of mobile robotics for a variety of purposes. For instance, for *mapping* (Sinha et al. 2018; Moolan-Feroze et al. 2019), *localization* (Weinzaepfel et al. 2019; Cattaneo et al. 2019), *navigation* (Zhao et al. 2018; Ma et al. 2019) and *simultaneous localization and mapping* (Lu and Lu 2019; Liu et al. 2019). A complete state-of-the-art review on mobile robotics tasks based on the use of AI can be found in (Cebollada et al. 2020). Other applications of AI in the context of mobile robotics include: *self-driving navigation* (Polvara et al. 2018; Organisciak et al. 2020), *face detection and recognition* (Wang et al. 2017; Hu et al. 2021), *object recognition and categorization* (Zaki et al. 2019; Feng et al. 2020) and *mapping and localization* (Holliday and Dudek 2018; Ruan et al. 2019).

Convolutional Neural Networks (CNNs) are the most popular techniques among AI tools. Currently, they are used in many mapping and localization tasks due to their successful performance in many practical applications. They are designed to receive images as input and their structures are specially created to obtain descriptors that synthesize the information in them (Chollet et al. 2018). Therefore, they can be used to describe the global appearance of an image. In this sense, Cebollada et al. (2019) proposed holistic descriptors obtained with a CNN to perform localization within topological models, studying their strength against illumination variations. Also, Xu et al. (2019) and Leyva-Vallina et al. (2019) proposed these techniques to obtain the most probable robot position. Additionally, Ballesta et al. (2021) studied localization tasks using CNNs and regression layers as global appearance descriptors. Recently, Rostkowska and Skrzypczyński (2023) employed the EfficientNet model (Tan and Le 2019) to embed an omnidirectional image into a single descriptor followed by a K-Nearest Neighbours (KNN) algorithm to robustly predict the topological position in a given database (map). In this regard, this work implements the Facebook AI Similarity Search

(FAISS) library (Johnson et al. 2019) to efficiently perform the nearest neighbour search using a KD-Tree.

Some well known architectures have been used as basic structures to develop new modified networks for robotic navigation purposes. AlexNet (Krizhevsky et al. 2012), VGG16 (Simonyan and Zisserman 2014), GoogleNet (Szegedy et al. 2015) or NetVLAD (Arandjelovic et al. 2016) are some of them.

The Convolutional Neural Networks presented above can be used to form a Siamese Neural Network. In the field of robotics, they have been rarely used and some studies that proposed this structure in this field are mentioned below. For example, Utkin et al. (2017) use a Siamese Neural Network to support the security control of a robot by detecting anomalies in its behaviour and Zeng et al. (2018) present a robotic pick-and-place system capable of identifying and grasping both known and novel objects in cluttered environments using a Siamese Neural Network. Moreover, Li and Zhang (2019) use the VGG16 network to conform a Siamese structure for object detection and tracking. Additionally, Zhang and Peng (2019) presented a study in which Siamese Networks are followed by Fully Connected layers or Region Proposal Network structures in the context of real-time visual tracking.

Regarding robot localization tasks, Leyva-Vallina et al. have proposed the use of Siamese Neural Networks to address the place recognition problem in garden environments (Leyva-Vallina et al. 2019, 2021). Moreover, this architecture has been proposed for localization using LiDAR scans (Yin et al. 2018; Chen et al. 2022).

In the present paper, we address the localization of a mobile robot using panoramic images in such a way that we study in detail different architectures and training configurations of Siamese Neural Networks. For this purpose, we propose as an initial approach to train and test the capability of the network to distinguish between images captured in the same and different rooms. In addition, in this study we also tackle the global localization problem using Siamese Neural Networks.

3 Visual localization using Siamese Neural Networks

Siamese Neural Networks can be described as a superstructure that includes, at least, two different Neural Networks beneath. Weights are shared between the networks and a single output is generated by combining the outputs of both networks. Figure 1 shows a general representation of a Siamese Neural Network architecture. In the present work, we use Convolutional Neural Networks to conform the two branches of the Siamese Neural Network. The output of each CNN is a descriptor which is used to characterize each input image. The dissimilarity of the input images is computed by measuring the distance between these descriptors. In this way, Siamese Neural Networks can be trained to generate similar descriptors when the training images belong to the same category. This fact makes Siamese Neural Networks particularly suitable to perform image retrieval tasks. Additionally, it is worth noting that the outputs, training, and performance of the network depend directly on:

- The architectures used in subnetworks W1 and W2 to extract the main features of the images.
- The conversion of the feature maps from the convolutional layers to a descriptor vector.

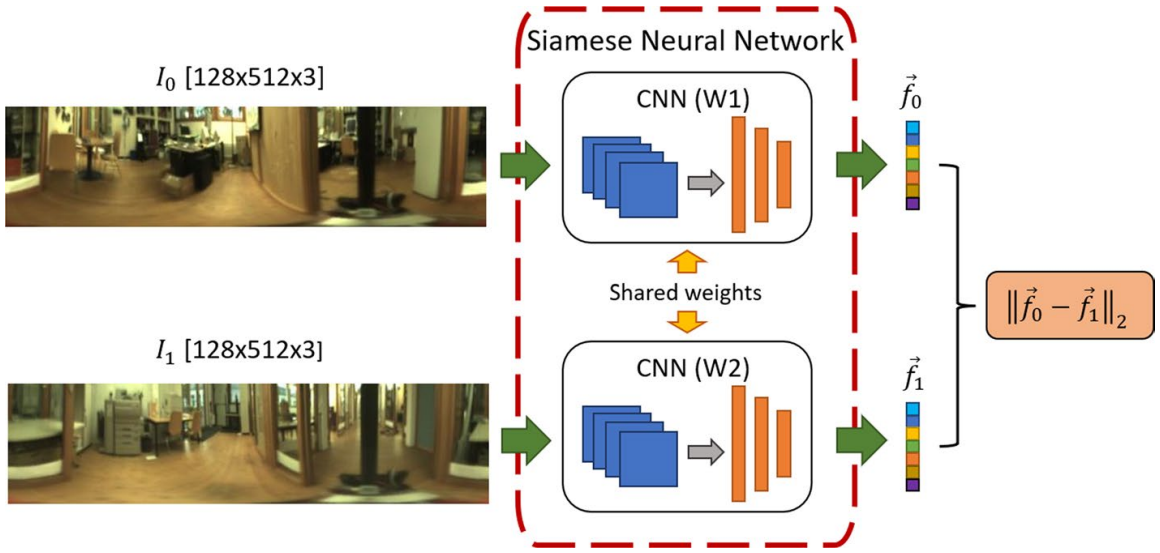


Fig. 1 Representation of the architecture of a general Siamese Neural Network

- The dimension of the output descriptors that embed the pair of input images.
- The training carried out with the available images. In particular, the labelling and the ratio of images of each category.

In this manuscript, we analyze the influence of these items on the visual localization of the robot. In this sense, we assume that a visual map of the environment is initially available. To obtain this map, the robot has moved throughout the area capturing omnidirectional images along the trajectory. Firstly, the images are transformed to a panoramic format (with size 128x512 in the present work), resulting in the set $\{I_1, I_2, \ldots, I_N\}$. These images are captured from N points of view, whose poses are known and stored $\vec{P}_i = (x_i, y_i, \theta_i), i = 1, \ldots, N$. Additionally the room where the picture has been captured is known too, so a set of labels is available: $\vec{R}_i = (r_i), i = 1, \ldots, N$. Each image will be embedded into a single descriptor during the localization, using the proposed architecture, yielding $\{\vec{f}_1, \vec{f}_2, \ldots, \vec{f}_N\}$. The trajectory followed by the robot includes different rooms with different visual information. In this work, these rooms include a corridor, some offices, a library and a bathroom.

Taking these facts into account, the initial map is composed by the set of images, their poses and the room in which the images are captured $\{(I_1, \vec{P}_1, r_1), (I_2, \vec{P}_2, r_2), \ldots, (I_N, \vec{P}_N, r_N)\}$. Using this information, some Siamese Neural Networks are trained to address localization.

3.1 Room discrimination

In this subsection an initial task related to localization is evaluated to study whether a Siamese Neural Network is able to distinguish between images captured from the same or from different rooms. For this purpose, the model will be trained and tested with pairs of random images captured from the same and/or different room.

3.2 Global localization

In this study we consider that a map of the environment is available, as described before. The absolute localization problem is solved by comparing the test image directly with all the images in the map. This comparison is performed using the descriptors $\vec{f}_i$ associated to each image in the map. The pose of the robot is found as the most similar descriptor contained in that map. The problem is approached with pure visual information and assuming that no information about the previous pose of the robot is available.

4 Architecture and training of the deep learning tools

The structure of a classical CNN used for classification tasks can be split into two different stages (Cebollada et al. 2019): the feature learning and the classification stages. Features are extracted using several convolutional layers whereas the classification task can be constructed using fully connected layers and a final Softmax function. In our approach, the classification stage is replaced by a feature aggregation phase. In this sense, the feature extraction phase outputs multiple feature maps which are flattened to a vector and dimensionally reduced by fully connected layers. This phase permits generating a single description vector per input image. As a result, the model provides two vectors $\vec{f}_0$ and $\vec{f}_1$ (one per input image). These descriptors are compared using the Euclidean distance in the comparison phase ($d(\vec{f}_0,\vec{f}_1) = \|\vec{f}_0 - \vec{f}_1\|_2$). This architecture is shown in Fig. 2. Therefore, during training, the weights of the networks are updated in order to obtain the optimal global descriptors. After the comparison, the distance between them and the similarity label ($1 : dissimilar, 0 : similar$) are used as data for the loss function. In our case the loss function used is the Constrastive Loss Function.

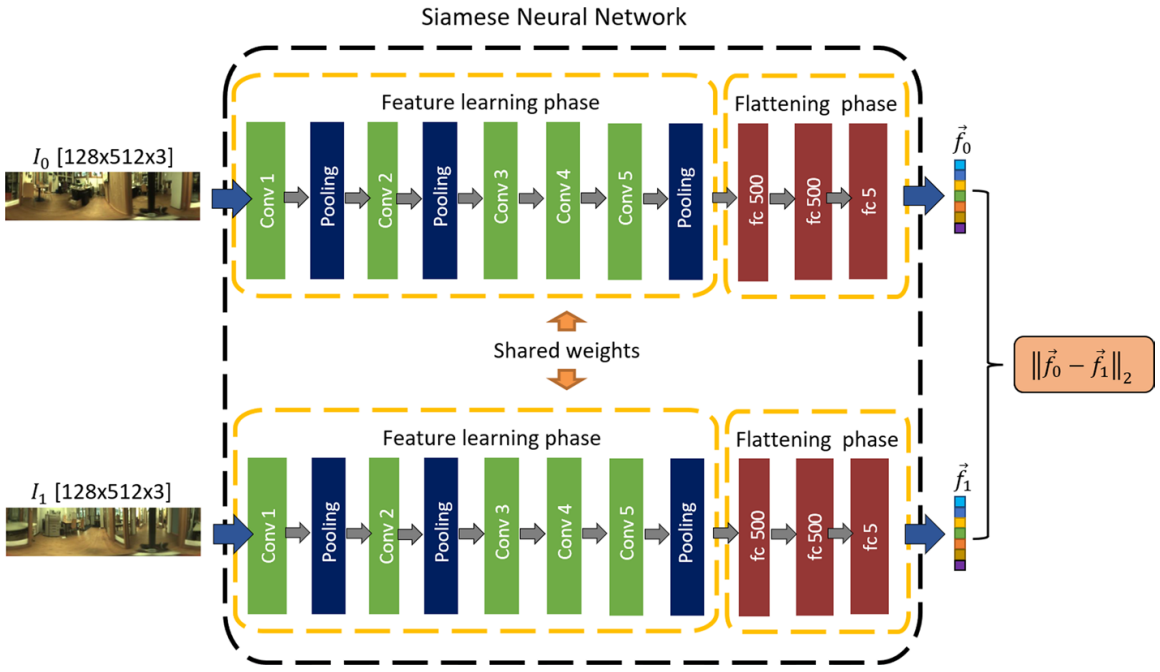


Fig. 2 Detailed representation of a Siamese Neural Network with AlexNet in the feature extraction and feature aggregation phase

$$L(\vec{f}_0,\vec{f}_1) = \frac{1}{2}(1-y)d(\vec{f}_0,\vec{f}_1)^2 + y\frac{1}{2}max(\alpha - d(\vec{f}_0,\vec{f}_1),0)^2 \quad (1)$$

Where y is the similarity label and $\alpha > 0$ is a margin. The margin defines a radius around the descriptor so that dissimilar pairs of images contribute to the loss function only if their distance is within this radius (Hadsell et al. 2006).

4.1 Parameters and networks

In this manuscript we compare different networks in the feature learning stage. As inputs to the feature aggregation stage we consider the representation computed in the last convolutional layer of Alexnet (Krizhevsky et al. 2012), DenseNet (He et al. 2016), VGG11, VGG13, VGG16 and VGG19 (Simonyan and Zisserman 2014). AlexNet is a pioneering CNN architecture known for its success in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2012. Visual Geometry Group (VGG) networks further contributed to the advancement of image classification problem, outperforming benchmarks on a variety of tasks and datasets outside of ImageNet (Bayraktar et al. 2019, 2020). The main difference between VGG networks is the number of convolutional layers: 11, 13, 16 and 19 layers respectively. In Table 1 the feature extraction layers of those CNNs are presented. Additionally two simple networks created with three conv2d layers are also evaluated (Table 2). The ReLU activation layers are not shown for brevity, but they have been used after each conv2d layer. The feature extraction layers are shown with black color in Tables 1 and 2. The different feature learning structures are evaluated in the Sect. 5.

In all the cases, the feature extraction stage outputs a high dimensional vector obtained by flattening the feature maps from the last maxpool or averagepool layer. Therefore, if the descriptor was extracted from this layer, comparing descriptors through nearest neighbour search would be computationally expensive. To alleviate this problem, we use fully connected layers to compress the flattened vector into a compact global vector descriptor, which can be used for efficient retrieval as demonstrated in (Schaupp et al. 2019). These layers are shown with blue color in Tables 1 and 2. As a global baseline three fully connected layers are used, but different versions are considered, with different number of neurons. The different layers used during the evaluation are presented in Table 3.

Other parameters are also tested during the training phase with the aim of obtaining the best Siamese Neural Network for our application. The hyperparameters considered during the evaluation are the following: the batch size (number of samples processed before the model is updated), the epochs (number of complete passes through the training dataset) and the percentage of images (percentage of training pairs of images from the same or different rooms, so that the network can learn adequately similarities and dissimilarities between rooms). In the experiments, the learning rate is kept constant at 0.001 (rate of change of the model in response to the estimated error) and the momentum is 0.9 (contribution of the parameter update step of the previous iteration upon the current iteration).

Table 1 Configuration of the feature extraction neural networks. (Color table online)

AlexNet	DenseNet	VGG11	VGG13	VGG16	VGG19
input (128 x 512 RGB image)					
conv2d-64	conv2d-112	conv2d-64	conv2d-64 conv2d-64	conv2d-64 conv2d-64	conv2d-64 conv2d-64
maxpool	maxpool	maxpool			
conv2d-192	conv2d-56 x 6	conv2d-128	conv2d-128 conv2d-128	conv2d-128 conv2d-128	conv2d-128 conv2d-128
maxpool	averagepool	maxpool			
conv2d-384 conv2d-256 conv2d-256	conv2d-28 x 12	conv2d-256 conv2d-256	conv2d-256 conv2d-256	conv2d-256 conv2d-256 conv2d-256	conv2d-256 conv2d-256 conv2d-256 conv2d-256
	averagepool	maxpool			
maxpool	conv2d-14 x 24	conv2d-512 conv2d-512	conv2d-512 conv2d-512	conv2d-512 conv2d-512 conv2d-512	conv2d-512 conv2d-512 conv2d-512 conv2d-512
	averagepool	maxpool			
	conv2d-7 x 16	conv2d-512 conv2d-512	conv2d-512 conv2d-512	conv2d-512 conv2d-512 conv2d-512	conv2d-512 conv2d-512 conv2d-512 conv2d-512
	averagepool	maxpool			
fc-500					
fc-500					
fc-5					

The ReLU activation layers have been omitted for brevity

* Blue color layers correspond to the feature aggregation layers

**VGG networks have their Batch Normalize (bn) version where after each conv2d layer a BatchNorm2d layer normalizes the results

Table 2 Simple convolutional neural networks without pretraining. (Color table online)

Simple 1	Simple 2
input (128 x 512 RGB image)	
conv2d-3 conv2d-8 conv2d-16	conv2d-3 conv2d-16 conv2d-32
maxpool	
fc-500	
fc-500	
fc-5	

Blue color layers correspond to the feature aggregation layers

Table 3 Configuration of the feature aggregation phase in our approach

Version 1	Version 2	Version 3
fc-500	fc-500	fc-1000
fc-500	fc-100	fc-1000
fc-5	fc-10	fc-10

4.2 Datasets and data augmentation

4.2.1 Training and test datasets

The images used in the experiments are obtained from an indoor dataset (Pronobis and Caputo 2009). This database was captured by an omnidirectional vision sensor mounted on a mobile robot which followed different trajectories that visited 9 different rooms. A variety of lighting conditions was considered to capture the sets of images.

Table 4 shows the number of images per room for each of the datasets used in this research. Two training sets are considered: training set 1 consists of 8486 images captured under cloudy, sunny and night illumination conditions (*COLD-Freiburg Part A Path 2 Cloudy 3, Freiburg Part A Path 2 Night 1, Freiburg Part A Path 2 Sunny 3*). Training set 2 has been obtained by applying a data augmentation to the cloudy sequence of training set 1, thus generating 977,856 images. With respect to the test sets, four different sets are considered: test set 1 consists of 2595 images under cloudy lighting condition (*COLD-Freiburg Part A Path 2 Cloudy 2*), test set 2 contains images captured under night lighting condition and consists of 2707 images (*COLD-Freiburg Part A Path 2 Night 2*), test set 3 consists of 2114 images under sunny lighting condition (*COLD-Freiburg Part A Path 2 Sunny 2*) and test set 4 is composed of all the images in the previous test sets. It should be noted that the images in the test sets are different, in all cases, from the images that constitute the training sets. Finally, the visual map has been obtained after sampling the path under the cloudy lighting condition of the test set 1, obtaining a total of 556 images.

Table 4 Summary of the training and test datasets

Room	Training dataset 1	Training dataset 2	Test dataset 1	Test dataset 2	Test dataset 3	Visual map
1P0-A	518	76,736	218	168	123	44
2P01-A	694	82,016	233	215	187	46
2P02-A	428	55,616	158	168	109	31
CR-A	3258	416,416	1183	1114	793	238
KT-A	674	80,608	229	270	213	46
LO-A	395	46,464	132	121	102	26
PA-A	804	99,968	284	241	191	57
ST-A	495	53,152	151	198	180	30
TL-A	619	66,880	190	212	216	38
Total	8486	977,856	2595	2707	2114	556

This table shows the number of images per room and the total of images of each dataset

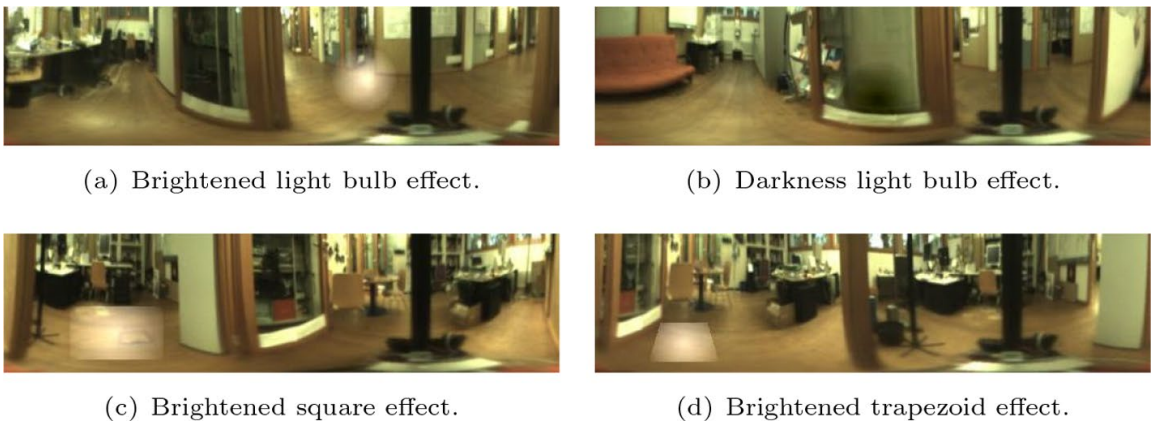
In this way, the training sets will be used to carry out the training of the Siamese Neural Networks, and the test sets will evaluate the performance of the networks under the three lighting conditions. The visual model is the map available for the robot to carry out the localization, so it will be used in the testing phase of the global localization.

4.2.2 Data augmentation

Additionally, a data augmentation technique is proposed as a method to improve the performance of the network. It increases the number of images in the training dataset. Having a larger number of training images reduces the overfitting of the model and boosts its robustness against real operating conditions. Cabrera et al. (2021) and Sakkos et al. (2019) demonstrated the use of data augmentation in CNNs to improve their effectiveness under changing lighting conditions.

Our proposed data augmentation is focused mainly on such lighting conditions and concentrates on editing local regions by simulating lights, reflections and shadow effects caused by light sources from different angles. Moreover global illumination changes are also taken into account. Other effects not related with the illumination but that can appear when images are captured in real operating conditions are also used.

Local effects: Light sources that fall on a specific area or the surface of an object are reproduced. We call this local illumination changes since only a small patch of the image is being affected. The shape of different light sources can vary meaningfully. Circular shapes from light bulbs or square and trapezoid shapes from reflections or windows are common. We edit the intensity of different regions following these shapes to simulate the light source; the pixel intensity is increased to reproduce more bright or it is decreased to simulate a shadow effect. In order to replicate a realistic fading effect, the intensity of brightening/darkening is gradually decreased from the center to the edge as an attenuation of the light. The size of the shapes and the position is selected randomly to simulate the effect in different ways and so does the maximum value to consider different intensities. In our experiments these figures are built with sizes between 15 and 40 pixels, different intensities are applied



(a) Brightened light bulb effect. (b) Darkness light bulb effect.

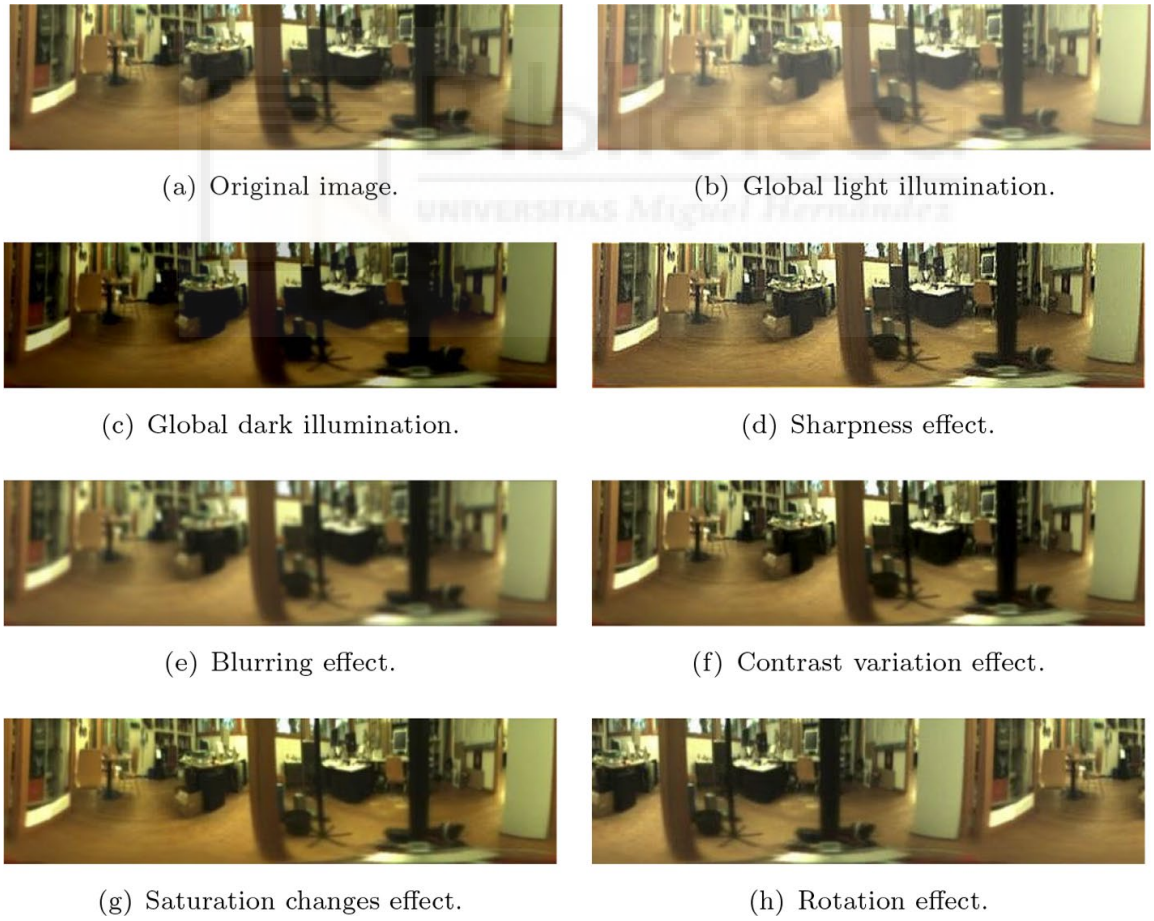
(c) Brightened square effect. (d) Brightened trapezoid effect.

Fig. 3 Individual local effects for data augmentation based on illumination

and the patch is degraded from intensity values ± 160 or 100 to 5. The effects and shapes are shown in the Fig. 3.

Global illumination: Global illumination variations can occur in some cases. To model such illumination changes, we need to alter pixels across the whole image, rather than in a small region. A constant value c is added to all the pixels to model a global brightness effect on the image or it is subtracted to simulate a global darkness. The value of c varies from 35 to 75 in this work. Figure 4b and c shows the effect.

Sharpness/Blurring: Finding sharper borders among diverse objects will contribute to provide a better separation among them and between foreground and background. In contrast, blurring effects are caused by low illumination and movements of the camera, which are common in mobile robotics. Both effects are incorporated in the data augmentation. They can be observed in Fig. 4d and e. Both can be achieved by a convolution operation using the following masks.



(a) Original image. (b) Global light illumination.

(c) Global dark illumination. (d) Sharpness effect.

(e) Blurring effect. (f) Contrast variation effect.

(g) Saturation changes effect. (h) Rotation effect.

Fig. 4 Global effects for data augmentation

Sharpness effect	Blurring effect
$m_{sh} = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	$m_{bl} = \frac{1}{25}\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$

Contrast variation: The contrast of the image plays an important role in highlighting different objects in the scene. Low contrast images usually look softer and with less shadows and reflections. The effect is proposed for this data augmentation to improve the robustness of the framework. The contrast is modified following the next equation:

$$I_s = 64 + c * (I - 64)$$

where I_s is the resulting image, I the original image and c is the contrast factor. For $c > 1$ the contrast increases and $c < 1$ decreases the contrast. Additionally, an equalization of the image is also added to the data augmentation set. It evenly distributes the histogram values, which permits obtaining a new contrast augmentation effect. Figure 4f shows this effect.

Saturation changes: The colour saturation of the image deals with the intensity of the colour. The less saturation, the less colourful the image is, even it can resemble a grey-scale image if the saturation is very low. In contrast, more vivid colours are obtained when the colour saturation is high. It can simulate situations when illumination changes significantly. The colour saturation can be edited by converting the RGB image to HSV, after that, it is possible to directly change the saturation channel by multiplying it by a constant factor c. If the saturation attribute is multiplied by $c > 1$ the colours become more saturated and by $c < 1$ the colour saturation decreases. The effect can be seen in Fig. 4g.

Rotation: The original image covers 360° around the robot. For that reason the image can be rotated without losing any piece of information. This effect will simulate the situation in which the robot is in the same position but the orientation is different. Moreover, having a training dataset containing this type of effect is expected to provide the Neural Network with rotation invariance. Figure 4h shows a rotation effect of 115°. Random rotations between 10 and 350° are applied to the training images.

Combined changes: Additionally some effects are combined to obtain a larger data augmentation, but not all the effects are combined together. Global illumination and a single local effect are combined in all the possible variations, e.g. global darkness is combined with a brightening circle shape effect, global brightness is combined with a brightening trapezoidal effect, etc. Additionally, the local effects are also combined. The circle shape effect is combined with the square effect, the trapezoidal effect or another circle shape effect, the combinations can be brightened+brightened, brightened+darkness and darkness+darkness; the circle shape effect is also combined with other two circle shape effects,

obtaining an image with three light bulb effects. Finally, the rotation effect is individually combined with all the effects and the combinations described above.

4.3 Training and testing the Siamese Neural Network

As presented in Sect. 4.1, different CNNs architectures can be used as the base of Siamese Neural Networks. Initially, we start from pretrained networks with known weights and biases. Then, we retrain the network to fit it to our application. This transfer learning technique is well-known and has previously been used in mobile robotics (Cabrera et al. 2021).

Ssection 4.3.1 will address an initial task which consists in training and evaluating the capability of a Siamese Neural Network to identify whether two images were captured from the same or different rooms. Finally, in Sect. 4.3.2 we will detail the characteristics of the training and test to address the absolute localization problem with siamese architectures. Emphasis will be placed on the labelling required to perform the desired task.

4.3.1 Room Discrimination

The main goal of this task is to evaluate whether a Siamese Neural Network is capable of determining if two images belong to the same or different room. It is an important capability to perform localization tasks.

The *training phase* is performed by feeding the network with pairs of images. These pairs are labelled with 0 if they have been captured from the same room and 1 if not. The ratio same/different room pairs is varied in the *training phase* to study its influence.

During the *test phase*, pairs of images are fed into the network. At the output, the network labels them with a number between 0 and 1; if the result is under 0.5 we interpret that the images have been captured from the same room. On the contrary, the images belong to different rooms. The images used to test the network are different from the training images, they are captured in the same building but in different times, in a variety of lighting conditions. Also the trajectory followed by the robot to capture the test images is similar to the one used to capture the training images, but the images are captured from different robot poses (Fig. 5).

4.3.2 Global localization

The global localization problem considers the estimation of the robot pose within the whole floor of the building. For this purpose, a Siamese Neural Network is trained. The *training* is carried out with image pairs labelled with the following equation:

$$Label(I_i, I_j) = \begin{cases} \frac{\|\vec{p}_i - \vec{p}_j\|_2}{K_b} & \text{if } I_i \text{ and } I_j \text{ belong to the same room} \\ 1 & \text{otherwise} \end{cases} \quad (2)$$

where I_i and I_j are two images and $\vec{p}_i$ and $\vec{p}_j$ are their corresponding positions (coordinates of the capture points). This constitutes a normalized Euclidean distance between the

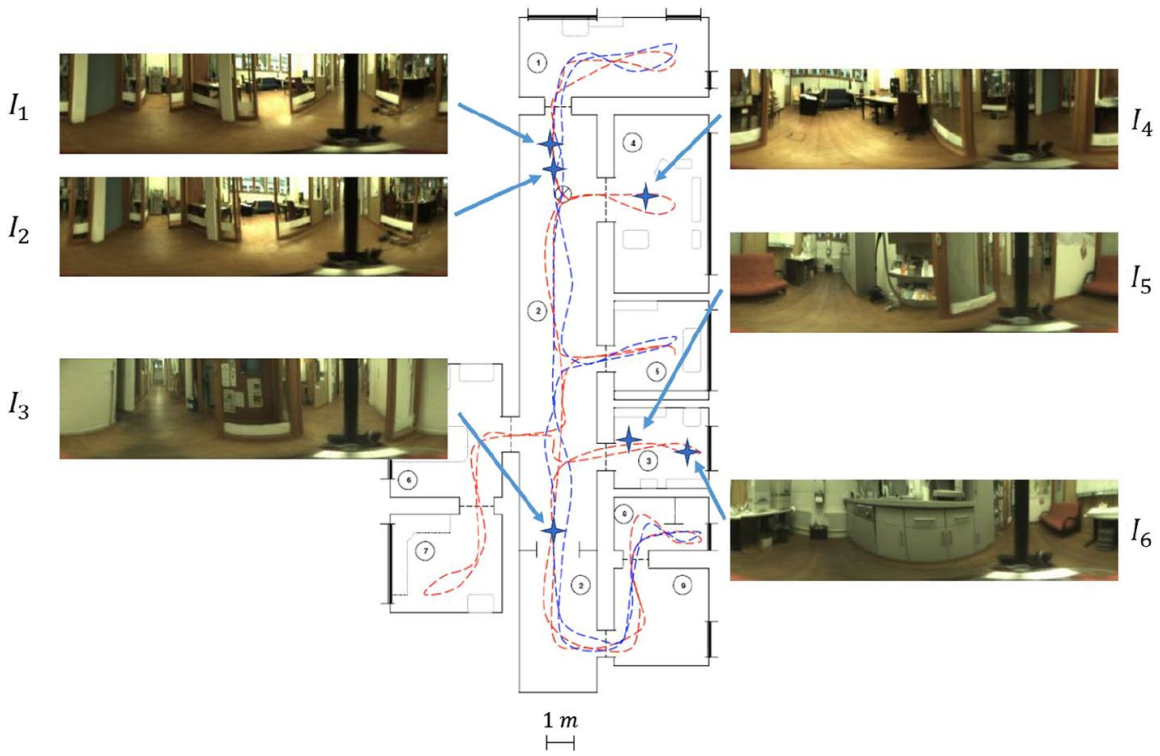


Fig. 5 Example of different trajectories of the robot

Table 5 Example pairs and its label value for the absolute localization task

Pair	Euclidean distance (m)	Label value
$I_1 - I_2$	0.33	$\frac{0.33}{18.99} = 0.017$
$I_1 - I_3$	12.82	$\frac{12.82}{18.99} = 0.675$
$I_1 - I_4$	–	1
$I_1 - I_5$	–	1
$I_4 - I_5$	–	1
$I_5 - I_6$	2.48	$\frac{2.48}{18.99} = 0.131$

The labels of the images are shown in Fig. 5

18.99 m is the maximum distance between two images in the target environment

capture points. K_b corresponds to the maximum distance between two images in the building. Table 5 shows different examples according to Fig. 5.

Once the network has been trained, the test is performed by using the map which is composed by the set of image descriptors and their positions $\{(\vec{f}_1, \vec{p}_1), (\vec{f}_2, \vec{p}_2), \ldots, (\vec{f}_N, \vec{p}_N)\}$. Each descriptor has been calculated by the trained Siamese Neural Network. The absolute localization is performed as a pairwise comparison between image descriptors. Given a test image I_t, the Siamese Neural Network outputs its corresponding descriptor $\vec{f}_t$. Finally, the position of the robot is estimated by selecting the pose associated to the descriptor in the map that minimizes the distance $\|\vec{f}_t - \vec{f}_i\|_2$, with $i = 1, \ldots, N$.

Table 6 Accuracy using different feature extraction neural networks. (Color table online)

Network	Global Test Accuracy	Same Room Accuracy	Different Room Accuracy
Simple 1	84.59%	98.16%	**71.03%**
Simple 2	86.45%	98.87%	74.06%
Alexnet	86.10%	98.78%	73.41%
Densenet	86.06%	97.61%	74.52%
VGG11	87.43%	99.08%	75.78%
VGG11bn	87.51%	97.49%	77.53%
VGG13	**89.65%**	99.44%	**79.86%**
VGG13bn	88.52%	98.26%	78.77%
VGG16	89.19%	**99.47%**	78.91%
VGG16bn	**82.04%**	**92.68%**	73.39%
VGG19	89.17%	99.30%	79.04%
VGG19bn	86.58%	95.52%	77.64%

5 Experiments

The set of experiments is designed to test the performance of the Siamese Neural Network as global descriptor generator to tackle the room discrimination and global localization task as explained in Sects. 4.3.1 and 4.3.2.

5.1 Room Discrimination

In this subsection we assess the ability of the network to predict whether two images are taken from the same room. The effectiveness of the Siamese Neural Network is calculated by comparing pairs of images and checking their label. The results are expressed in percentage of accuracy. Several experiments have been conducted while varying different parameters: the feature extraction architecture, the feature aggregation layers and the percentage of similar/dissimilar images. As common parameters, we train the network using 8486 pairs of images per epoch from the training dataset 1 and we use the Stochastic Gradient Descent (SGD) optimiser, with a learning rate of 0.001 and momentum of 0.9. Moreover, we test the network with 7000 pairs of images extracted from the test dataset 4.

5.1.1 Influence of the architecture on the feature extraction process

In this subsection we compare different models in the feature extraction stage of a Siamese Neural Network. The different models used can be observed in Table 1. The training has been performed using a batch size of 256 and 5 epochs. During training, the dataloader presents a 50% of images from the same room and a 50% of images from the different rooms. During these experiments, the feature aggregation is performed with 3 fully connected layers composed by 500–500-5 neurons in each.

Results are presented in Table 6 in terms of global accuracy. Additionally, the test accuracy for the same and different room predictions is also presented. The table shows that the best networks are VGG13 and VGG16. They obtain the best accuracy for predicting pairs of images in the same room (99.44% and 99.47% respectively). In addition, VGG13 and VGG16 present the best accuracy predicting if two images are taken from different rooms (79.86% and 78.91%). Moreover, the ‘Simple 1’ and ‘Simple 2’ networks obtain considerably good results using only three convolutional layers. Finally, in general terms, it can be

Table 7 Accuracy of **VGG13**. (Color table online)

Epoch	Percentage of Training Images (same-different)	Number of Training Images (same-different)	Global Accuracy	Same Room Accuracy	Different Room Accuracy
7	5%-95%	3,046-57,882	89.88%	92.03%	87.73%
9	5%-95%	3,917-74,419	91.89%	92.51%	91.27%
11	5%-95%	4,787-90,957	92.20%	92,71%	91,70%
7	10%-90%	6,093-54,835	92.72%	98.13%	87.30%
9	10%-90%	7,834-70,502	94.76%	98.69%	90.82%
11	10%-90%	9,574-86,170	95.08%	98.90%	91.25%
7	25%-75%	15,232-45,696	93.10%	99.09%	87,12%
9	25%-75%	19,584-58,752	93.46%	99.06%	87.86%
11	25%-75%	23,936-71,808	93.53%	99.21%	87.85%

The table presents a variation in the total number of images and in the same-different ratios of training images

Table 8 Accuracy of **VGG16**. (Color table online)

Epoch	Percentage of Training Images (same-different)	Number of Training Images (same-different)	Global Accuracy	Same Room Accuracy	Different Room Accuracy
7	5%-95%	3,046-57,882	94.35%	96.47%	92.23%
9	5%-95%	3,917-74,419	94.94%	96.48%	93.39%
11	5%-95%	4,787-90,957	94.24%	97.77%	90.72%
7	10%-90%	6,093-54,835	93.04%	97.16%	88.92%
9	10%-90%	7,834-70,502	94.26%	97.18%	91.35%
11	10%-90%	9,574-86,170	93.59%	97.96%	89.22%
7	25%-75%	15,232-45,696	92.46%	99.21%	85.71%
9	25%-75%	19,584-58,752	92.28%	99.30%	85.25%
11	25%-75%	23,936-71,808	91.78%	98.81%	84.74%
7	40%-60%	24,371-36,557	92.95%	99.38%	86.52%
9	40%-60%	31,334-47,002	92.72%	99.48%	85.95%
11	40%-60%	38,298-57,446	93.28%	99.50%	87.05%

The table presents a variation in the total number of images and in the same-different ratios of training images

observed that all the architectures perform better in predicting whether two images belong to the same room. For this reason, we consider below the possibility of varying the percentage of images of each category in the training phase.

5.1.2 Influence of the training parameters

In the light of the previous results, next, different training parameters are evaluated. As we explain in the previous subsection, the ratio of training pairs of images in each category is expected to have a substantial influence upon the results. In consequence, we propose to change the percentage of pairs of images at the training phase. The percentage of images taken from the same and different rooms varies from 5% to 40% and from 95% to 60% respectively. For brevity, we only show the results obtained with VGG13, VGG16 and AlexNet networks. The rest of the training parameters is tuned as before, using 256 as

Table 9 Accuracy of **AlexNet**. (Color table online)

Epoch	Percentage of Training Images (same-different)	Number of Training Images (same-different)	Global Accuracy	Same Room Accuracy	Different Room Accuracy
7	5%-95%	3,046-57,882	92.36%	90.11%	94.60%
11	5%-95%	4,787-90,957	93.58%	94.08%	93.07%
14	5%-95%	6,093-115,763	93.68%	94.14%	93.22%
7	10%-90%	6,093-54,835	92.05%	94.65%	89.44%
11	10%-90%	9,574-86,170	93.41%	96.84%	89.98%
14	10%-90%	12,186-109,670	93.01%	97.19%	88.82%
7	25%-75%	15,232-45,696	90.91	97.54%	84.28%
11	25%-75%	23,936-71,808	91.16%	98.92%	83.39%
14	25%-75%	30,464-91,392	90.59%	98.28%	82.19%
7	40%-60%	24,371-36,557	88.33%	98.80%	77.85%
11	40%-60%	38,298-57,446	88.65%	99.07%	78.23%
14	40%-60%	48,742-73,114	88.54%	99.25%	77.82%

The table presents a variation in the total number of images and in the same-different ratios of training images

Table 10 Accuracy using **VGG16** and different batch sizes. (Color table online)

Batch Size	Epoch	Percentage of Training Images (same-different)	Global Accuracy	Same Room Accuracy	Different Room Accuracy
256	7	5%-95%	94.35%	96.47%	92.23%
256	11	5%-95%	94.24%	97.77%	90.72%
256	7	10%-90%	93.04%	97.16%	88.92%
256	11	10%-90%	93.59%	97.96%	89.22%
256	7	25%-75%	92.46%	99.21%	85.71%
256	11	25%-75%	91.78%	98.81%	84.74%
16	7	5%-95%	95.50%	98.26%	92.74%
16	11	5%-95%	93.84%	98.83%	88.85%
16	7	10%-90%	93.77%	98.13%	89.41%
16	11	10%-90%	94.42%	98.80%	90.05%
16	7	25%-75%	94.77%	99.15%	90.39%
16	11	25%-75%	94.08%	99.15%	89.00%

batch size and a feature aggregation phase with three fully connected layers composed by 500, 500 and 5 neurons. The results are presented in Tables 7, 8 and 9. They show a correlation between the percentage of images of same/different room and its respective accuracy, i. e., when the percentage of pairs of images in the same room increases, its associated accuracy also does and a similar phenomenon occurs with the different room category.

Until this moment, all the experiments have been performed using 256 as batch size, but other values have been tested in order to check the best configuration. Tables 10 and 11 show the accuracy using different batch sizes. They show that the global accuracy increases when the batch size is lower.

These tables show that relatively good performances can be achieved with some configurations. Notwithstanding that, we observe that in general terms, the same-room accuracy tends to decrease when the different-room accuracy increases and vice versa. This will be

Table 11 Accuracy using **AlexNet** and different batch sizes. (Color table online)

Batch Size	Epoch	Percentage of Training Images (same-different)	Global Accuracy	Same Room Accuracy	Different Room Accuracy
256	7	5%-95%	89.76%	90.11%	94.60%
256	11	5%-95%	93.58%	94.08%	93.07%
256	7	10%-90%	93.77%	98.13%	89.41%
256	11	10%-90%	94.42%	98.80%	90.05%
256	7	25%-75%	90.91%	97.54%	84.28%
256	11	25%-75%	91.16%	98.92%	83.39%
16	7	5%-95%	94.64%	96.25%	93.02%
16	11	5%-95%	95.24%	98.25%	92.24%
16	7	10%-90%	95.06%	98.87%	91.25%
16	11	10%-90%	95.07%	98.92%	91.22%
16	7	25%-75%	94.76%	99.10%	90.42%
16	11	25%-75%	94.60%	99.26%	89.94%

Table 12 Accuracy using **VGG16** and different feature aggregation layers. (Color table online)

Fully Connected Layers	Batch Size	Epoch	Global Accuracy	Same Room Accuracy	Different Room Accuracy
500-500-5	16	7	93.77%	98.13%	89.41%
500-500-5	16	11	94.42%	98.80%	90.05%
500-500-5	16	14	94.75%	99.10%	90.39%
500-100-10	16	7	95.76%	98.92%	92.60%
500-100-10	16	11	95.98%	99.11%	92.86%
500-100-10	16	14	95.44%	99.18%	91.70%
1000-1000-10	16	7	96.16%	98.90%	93.41%
1000-1000-10	16	11	95.63%	99.10%	92.16%
1000-1000-10	16	14	95.27%	99.10%	91.44%

analyzed deeply in future works, but it may be due to the use of the Contrastive Loss function (Sun et al. 2020a).

5.1.3 Influence of the architecture of the feature aggregation layers

As explained in Sect. 4.1, the feature extraction layers output a matrix that is flattened and compressed in the feature aggregation phase. Different combinations of fully connected layers are also evaluated. All these experiments have been performed training the network with a 10 of pairs of images taken from the same room and a 90 of pairs of images from different rooms.

Tables 12 and 13 show the results using 3 different combinations of fully connected layers. Each variation is described in Table 3. Similar results are obtained with the 3 different variations. The best result is obtained with 3 fully connected layers with 1000-1000-10 neurons each. Finally, if we analyse jointly all the results of the room discrimination experiment, the best result is obtained using VGG16 as the feature extraction network, 3 fully connected layers (1000-1000-10), 7 epoch and a batch size of 16; with this configuration 96.16% global accuracy is obtained: 98.90% same room accuracy and 93.41% different room accuracy.

Table 13 Accuracy using **AlexNet** and feature aggregation layers. (Color table online)

Fully Connected Layers	Batch Size	Epoch	Global Accuracy	Same Room Accuracy	Different Room Accuracy
500-500-5	16	7	93.77%	98.13%	89.41%
500-500-5	16	11	94.42%	98.80%	90.05%
500-500-5	16	14	93.84%	98.68%	88.99%
500-100-10	16	7	95.31%	98.20%	92.42%
500-100-10	16	11	95.41%	98.98%	91.83%
500-100-10	16	14	95.10%	99.06%	91.15%
1000-1000-10	16	7	95.36%	98.72%	91.99%
1000-1000-10	16	11	94.66%	98.59%	90.74%
1000-1000-10	16	14	95.28%	99.12%	91.43%

Table 14 Localization error in terms of mean absolute error (MAE), mean square error (MSE) and average recall (%) at top 1% (Recall@1%) with **VGG16**. (Color table online)

Percentage of Training Images (same-different)	Global MAE	Global MSE	Global Recall@1%	Cloudy MAE	Night MAE	Sunny MAE
80%-20%	0.628 m	0.612 m^2	45.01%	0.183 m	0.560 m	0.802 m
70%-30%	0.604 m	0.540 m^2	46.37%	0.175 m	0.538 m	0.771 m
60%-40%	0.601 m	0.417 m^2	48.45%	0.180 m	0.499 m	0.865 m
50%-50%	0.582 m	0.424 m^2	48.34%	0.175 m	0.484 m	0.838 m
40%-60%	0.509 m	0.335 m^2	49.82%	0.148 m	0.455 m	0.651 m
30%-70%	0.519 m	0.359 m^2	52.52%	0.152 m	0.491 m	0.663 m
20%-80%	0.520 m	0.366 m^2	53.03%	0.152 m	0.492 m	0.664 m

The table presents the global localization results with variations in the same-different ratio of training image pairs

5.2 Global localization

The global localization is performed as explained in Sect. 4.3.2. The VGG16 network is employed in this task since it led to the best results in the room discrimination task. Different experiments have been performed in order to choose the best configuration. We will mainly analyze the ratio of same/different room pairs, which is the parameter that has shown the greatest influence on the results. Moreover, in this subsection we will assess the influence of the data augmentation on the results. Each pair of images is labelled according Eq. 2.

First, concerning the experiment to evaluate the influence of the ratio same/different room pairs, we train the network using 8486 pairs of images per epoch from the training dataset 1. Second, with respect to the experiment to assess the effect of the data augmentation, 977,856 pairs of images per epoch from the training dataset 2 are used. These two experiments are described in Sect. 5.2.1. In both cases, the fully connected layers are configured with 500-500-5 neurons. Moreover, Sect. 5.2.2 evaluates the influence of the feature aggregation layers. In this case, the training dataset 1 is used. As common parameters, we use 16 as batch size, the Stochastic Gradient Descent (SGD) optimizer, with a learning rate of 0.001 and a momentum of 0.9 and 30 epochs.

Table 15 Localization error in terms of mean absolute error (MAE), mean square error (MSE) and average recall (%) at top 1% (Recall@1%) with **VGG16** and data augmentation. (Color table online)

Epochs	Percentage of Training Images (same-different)	Global MAE	Global MSE	Global Recall@1%	Cloudy MAE	Night MAE	Sunny MAE
1	50%-50%	0.608 m	0.529 m^2	37.35%	0.063 m	0.468 m	1.188 m
1	40%-60%	0.609 m	0.596 m^2	51.26%	0.061 m	0.468 m	1.270 m
4	50%-50%	0.904 m	0.914 m^2	46.74%	0.038 m	0.396 m	1.826 m
4	40%-60%	0.497 m	0.440 m^2	53.18%	0.041 m	0.403 m	1.205 m
11	50%-50%	0.422 m	0.428 m^2	50.43%	0.044 m	0.359 m	1.226 m
11	40%-60%	0.331 m	0.222 m^2	57.99%	0.042 m	0.318 m	1.005 m
21	50%-50%	0.281 m	0.207 m^2	57.51%	0.038 m	0.268 m	1.051 m
21	40%-60%	0.253 m	0.162 m^2	55.99%	0.033 m	0.257 m	0.991 m

The table presents the global localization results with variations in the same-different ratio of training image pairs

5.2.1 Influence of the training parameters

Ratio of same/different room pairs: Table 14 shows the results using VGG16 in the feature extraction part and three fully connected layers with 500-500-5 neurons in the feature aggregation part. The training of the model has been performed with different percentages of pairs of images belonging to the same and different rooms. The results show that the lowest localization error is obtained when the training is performed using 40% of images from the same room and 60% of images from different rooms. In general, The CNN shows excellent overall performance, especially when tested under the same lighting conditions as the training images (cloudy). However, the performance decreases in sunny conditions which are the most challenging test conditions. Studying the results, as a general rule, training with a large percentage of image pairs from the same room deteriorates the localization error.

Data Augmentation: Next, we evaluate the influence of the data augmentation on the localization task. Table 15 presents the results using the training dataset 2 (augmented) and test datasets 1, 2 and 3. For this purpose, we will start from the best configurations obtained so far and show the results according to the percentage of training image pairs. When the training is performed with the augmented dataset, remarkable results in terms of average error are obtained, especially in cloudy and night conditions. In this sense, the Mean Average Error decreases by 10 cm in cloudy conditions and by 20 cm in night conditions comparing to Table 14 (no data augmentation). However, training with this dataset shows a decrease in the performance of the

Table 16 Localization error in terms of mean absolute error (MAE), mean square error (MSE) and average recall (%) at top 1% (Recall@1%) with **VGG16** and different configurations of the fully connected layers when training 30 epochs and 50% of images from the same room and 50% of images from different rooms. (Color table online)

Fully Connected Layers	Global MAE	Global MSE	Global Recall@1%	Cloudy MAE	Night MAE	Sunny MAE
500-500-5	0.582 m	0.424 m^2	48.34%	0.187 m	0.510 m	0.751 m
1000-1000-10	0.590 m	0.572 m^2	49.11%	0.094 m	0.589 m	0.888 m
4096-4096-1000	0.831 m	0.859 m^2	51.15%	0.156 m	0.611 m	0.887 m

Table 17 Comparison with other methods. (Color table online)

Global-Appearance Descriptor Technique	Cloudy Error	Night Error	Sunny Error
Alexnet (Cebollada et al., 2022)	0.051 m	0.288 m	0.389 m
EfficientNet (Rostkowska and Skrzypczyński, 2023)	0.240 m	0.330 m	0.337 m
Siamese Network (ours)	0.148 m	0.455 m	0.651 m
Siamese Network + DA (ours)	0.033 m	0.257 m	0.991 m
HOG (Cebollada et al., 2022)	0.163 m	0.451 m	0.820 m
gist (Cebollada et al., 2022)	0.052 m	1.065 m	0.884 m

network in sunny circumstances. Therefore, the data augmentation proves to be beneficial, unless the test images experience substantial changes.

5.2.2 Influence of the architecture of the feature aggregation layers

To conclude the experimental section, Table 16 shows the results after evaluating different fully connected layers. Using 4096-4096-1000 neurons in these three layers demonstrated a consistent localization error for cloudy and night conditions. However, its performance degraded in sunny conditions. When the size of the fully connected layers is 1000-1000-10 the best result in cloudy conditions is achieved, but also the worst result for sunny scenarios. In contrast, the configuration 500-500-5 neurons consistently maintained low errors across all conditions, showing its adaptability to diverse lighting environments and generalization capabilities. The Siamese Neural Network is able to perform the localization with an average error of 0.5821 m when using as feature aggregation method three different fully connected layers with 500, 500 and 5 neurons.

5.2.3 General comparison with other methods

Finally, the Siamese Neural Networks are compared with other previous global-appearance techniques which include the use of a single AlexNet structure and two classic analytic descriptors: HOG and gist, as described in the work by Cebollada et al. (2022). Table 17 compares all the methods in a global localization task using, in all cases, the COLD-Freiburg Dataset. This table shows that the siamese structures with the VGG architecture

and the data augmentation proposed in the present work provide the best results in terms of localization error for cloudy and night conditions. Also, the approach proposed by Rostkowska and Skrzypczyński (2023) achieves good results in the case of sunny conditions. Apart from using a different architecture, the main difference between their approach and the one presented here is that they use a cross-entropy loss (single input) during training, while in the present paper we employ the contrastive loss (double input). Furthermore, in the present paper, the model is fed with an omnidirectional image transformed to a panoramic view, whereas in Rostkowska and Skrzypczyński (2023) directly use the omnidirectional image without conversion. In addition, they embed the image with an EfficientNet model (Tan and Le 2019) architecture which is followed by the Facebook AI Similarity Search (FAISS) KD-Tree, while in the approach proposed in the present paper the pairwise euclidean distance between descriptors is computed and employed to retrieve the closest descriptor in the database.

6 Conclusions

In this paper, a global localization method using Siamese Neural Networks has been proposed and evaluated. Localization, along with mapping, is one of the main tasks to be addressed by an autonomous mobile robot. First, an initial task of discriminating same and different rooms has been proposed in order to assess the ability of Siamese Neural Networs and know the influence of the most relevant parameters. After that, the global localization problem is addressed.

In the experiments, several well known architectures have been tested to conform the Siamese Neural Network, some of which are AlexNet, VGG11, VGG13, VGG16, VGG19, VGG11bn, VGG13bn, VGG16bn and VGG19bn. The best performance in the initial task has been achieved by VGG13 and VGG16. In general terms, the VGG architectures have provided the best results.

Apart from these feature extraction architectures, a group of Fully Connected layers have been added to carry out the conversion of the activation maps resulting from the convolutional layers to a description vector. In the present work, different sizes of the Fully Connected layers have been studied, as well as the size of the final descriptor. For the initial task, the performance of the network is slightly higher when the Fully Connected layers sizes are 1000-1000-10. In contrast, in the global localization, the localization error decreases drastically in those networks that have a set of Fully Connected layers of size 500-500-5 neurons.

The training parameter that contributes most to the performance of the network is the percentage of image pairs belonging to the same and different rooms. In this sense, there is a correlation between the percentage of images of same/different room and its respective accuracy, i.e., when the percentage of pairs of images in the same room increases, its associated accuracy also does and a similar effect occurs with the different room category. Furthermore, when the same room accuracy increases, the different room accuracy decreases, and vice versa. This situation may be caused by the Contrastive Loss function which has an associated lack of flexibility in the optimization. Other loss functions used in other applications could improve localization results, such as Circle Loss (Sun et al. 2020b) and will be considered in future studies.

In addition, a data augmentation technique has been proposed in order to improve the performance of the network. The proposed effects try to simulate real operating conditions. In addition, a set of effects specially designed to increase the robustness against changes of the lighting conditions in the scene have been generated. As for the results obtained, the performance of the network is especially benefited when working in cloudy and night lighting conditions. In the case of the cloudy lighting condition, when the training is performed with data augmentation, the average localization error is reduced around 12 cm. As for the night illumination condition, the average error is reduced around 20 cm. On the contrary, in sunny illumination condition the average localization error increases 34 cm when data augmentation is used. Thus, the siamese architecture is very efficient at solving the localization problem in real operating conditions, if the changes in the lighting conditions are not considerable, i.e., when working in cloudy and night scenarios. However, it is less effective at describing images in the presence of significant changes in lighting conditions, such as in the sunny scenarios. Other methods (such as HOG or gist) describe the image globally and give equal importance to all its regions, thus providing better resilience to large illumination changes. The reduced performance on sunny conditions when using siamese architectures can be explained by the lack of flexibility associated to the fact of having two networks with identical weights. In addition, the training process may have introduced an imbalance that causes the network to be more capable of detecting similarities than dissimilarities or vice versa. Additionally, the training dataset 1 (without data augmentation) comprises images from all illumination conditions, whereas the training dataset 2 (with data augmentation) is limited to cloudy images and attempts to replicate other illumination conditions by applying global and local effects. In this context, the proposed effects for data augmentation are beneficial in cloudy and night conditions, thus enhancing the performance of the model in these scenarios. However, the illumination effects that simulate different sunny conditions have been proven to be less effective than using real images captured at this particular illumination condition.

As future works, the proposed localization techniques will be extended to outdoor environments, which are more challenging because of their unstructured and changing conditions. In addition, other types of sensors will be considered to carry out the localization robustly, such as LiDAR.

Acknowledgements The Ministry of Science, Innovation and Universities (Spain) has supported this work through "Ayudas para la Formación de Profesorado Universitario" (FPU21/04969). This work is also part of the project TED2021-130901B-I00, funded by MCIN/AEI/10.13039/501100011033 and by the European Union "NextGenerationEU"/PRTR, and of the project PID2020-116418RB-I00 funded by MCIN/AEI/10.13039/501100011033.

Author contributions Conceptualization, J.J.C., V.R. and L.P.; methodology, J.J.C., V.R. and A.G.; software, J.J.C. and V.R.; validation, J.J.C. and V.R. ; formal analysis, A.G. and L.P.; writing (original draft preparation), V.R. and J.J.C.; writing (review and editing), A.G., L.P. and O.R.; supervision, A.G., L.P. and O.R.; Project administration L.P. and O.R.

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature.

Code availability Our code is publicly available on the project website https://github.com/juanjo-cabrera/IndoorLocalizationSNN.git.

Declarations

Competing interests The authors declare no competing interests.

References

Arandjelovic R, Gronat P, Torii A, Pajdla T, Sivic J (2016) NetVLAD: CNN architecture for weakly supervised place recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 5297–5307

Ballesta M, Payá L, Cebollada S, Reinoso O, Murcia F (2021) A CNN regression approach to mobile robot localization using omnidirectional images. Appl Sci 11(16):7521

Bayraktar E, Yigit CB, Boyraz P (2019) A hybrid image dataset toward bridging the gap between real and simulation environments for robotics: annotated desktop objects real and synthetic images dataset: ADORESet. Mach Vis Appl 30(1):23–40

Bayraktar E, Yigit CB, Boyraz P (2020) Object manipulation with a variable-stiffness robotic mechanism using deep neural networks for visual semantics and load estimation. Neural Comput Appl 32(13):9029–9045

Bromley J, Guyon I, LeCun Y, Säckinger E, Shah R (1993) Signature verification using a "Siamese" time delay neural network. In: Advances in neural information processing systems (NIPS 1993), vol 6. Morgan Kaufmann, San Mateo

Cabrera JJ, Cebollada S, Payá L, Flores M, Reinoso Ó (2021) A robust CNN training approach to address hierarchical localization with omnidirectional images. In: ICINCO, pp 302–310

Cao L, Ling J, Xiao X (2020) Study on the influence of image noise on monocular feature-based visual SLAM based on FFDNet. Sensors 20(17):4922

Cattaneo D, Vaghi M, Ballardini AL, Fontana S, Sorrenti DG, Burgard W (2019) CMRNET: camera to lidar-map registration. In 2019 IEEE intelligent transportation systems conference (ITSC). IEEE, pp 1283–1289

Cebollada S, Payá L, Román V, Reinoso O (2019) Hierarchical localization in topological models under varying illumination using holistic visual descriptors. IEEE Access 7:49580–49595

Cebollada S, Payá L, Flores M, Peidró A, Reinoso O (2020) A state-of-the-art review on mobile robotics tasks using artificial intelligence and visual data. Expert Syst Appl 167:114195

Cebollada S, Payá L, Jiang X, Reinoso O (2022) Development and use of a convolutional neural network for hierarchical appearance-based localization. Artif Intell Rev 55(4):2847–2874

Chen X, Läbe T, Milioto A, Röhling T, Behley J, Stachniss C (2022) OverlapNet: a siamese network for computing lidar scan similarity with applications to loop closing and localization. Auton Robot 46(1):61–81

Chollet F et al (2018) Deep learning with Python, vol 361. Manning, New York

Rublee E, Rabaud V, Konolige K, Bradski G(2011) ORB: an efficient alternative to SIFT or SURF. In: IEEE International conference on computer vision, ICCV 2011, pp 2564–2571

Feng Q, Shum HP, Morishima S (2020) Resolving hand-object occlusion for mixed reality with joint deep learning and model optimization. Comput Anim Virtual Worlds 31(4–5):e1956

Hadsell R, Chopra S, LeCun Y (2006) Dimensionality reduction by learning an invariant mapping. In: 2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06), vol 2. IEEE, pp 1735–1742

He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778

Holliday A, Dudek G (2018) Scale-robust localization using general object landmarks. In: 2018 IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE, pp 1688–1694

Hu S, Shum HP, Liang X, Li FW, Aslam N (2021) Facial reshaping operator for controllable face beautification. Expert Syst Appl 167:114067

Hu Y, Shum HP, Ho ES (2020) Multi-task deep learning with optical flow features for self-driving cars. IET Intell Transp Syst 14(13):1845–1854

Jeon M, Park D, Lee J, Jeon H, Ko M, Kim S, Choi Y, Tan AC, Kang J (2019) ReSimNet: drug response similarity prediction using Siamese neural networks. Bioinformatics 35(24):5249–5256
Johnson J, Douze M, Jégou H (2019) Billion-scale similarity search with GPUs. IEEE Trans Big Data 7(3):535–547
Junior JM, Tommaselli A, Moraes M (2016) Calibration of a catadioptric omnidirectional vision system with conic mirror. ISPRS J Photogramm Remote Sens 113:97–105
Khaliq A, Ehsan S, Chen Z, Milford M, McDonald-Maier K (2019) A holistic visual place recognition approach using lightweight CNNs for significant viewpoint and appearance changes. IEEE Trans Robot 36(2):561–569
Korrapati H, Mezouar Y (2017) Multi-resolution map building and loop closure with omnidirectional images. Auton Robot 41(4):967–987
Krizhevsky A, Sutskever I, Hinton GE (2012) ImageNet classification with deep convolutional neural networks. Adv Neural Inf Process Syst 25:1097–1105
Leyva-Vallina M, Strisciuglio N, Lopez-Antequera M, Tylecek R, Blaich M, Petkov N (2019) Tb-places: A data set for visual place recognition in garden environments. IEEE Access 7:52277–52287
Leyva-Vallina M, Strisciuglio N, Petkov N (2019) Place recognition in gardens by learning visual representations: data set and benchmark analysis. In: International conference on computer analysis of images and patterns. Springer, pp 324–335
Leyva-Vallina M, Strisciuglio N, Petkov N (2021) Generalized contrastive optimization of siamese networks for place recognition. arXiv preprint. arXiv:2103.06638
Li Y, Zhang X (2019) SiamVGG: visual tracking using deeper siamese networks. arXiv preprint. arXiv:1902.02804
Lin J, Peng J, Hu Z, Xie X, Peng R et al (2020) ORB-SLAM, IMU and wheel odometry fusion for indoor mobile robot localization and navigation. Acad J Comput Inf Sci 3(1):131–141
Liu W, Mo Y, Jiao J (2019) An efficient edge-feature constraint visual SLAM. In: Proceedings of the international conference on artificial intelligence, information processing and cloud computing, pp 1–7
Lu Y, Lu G (2019) Deep unsupervised learning for simultaneous visual odometry and depth estimation. In: 2019 IEEE international conference on image processing (ICIP). IEEE, pp 2571–2575
Ma L, Chen J et al (2019) Using RGB image as visual input for mapless robot navigation. arXiv preprint. arXiv:1903.09927
Moolan-Feroze O, Karachalios K, Nikolaidis DN, Calway A (2019) Improving drone localisation around wind turbines using monocular model-based tracking. In: 2019 International conference on robotics and automation (ICRA). IEEE, pp 7713–7719
Organisciak D, Sakkos D, Ho ES, Aslam N, Shum HP (2020) Unifying person and vehicle re-identification. IEEE Access 8:115673–115684
Parajuli N, Lu A, Stendahl JC, Zontak M, Boutagy N, Alkhalil I, Eberle M, Lin BA, O'Donnell M, Sinusas AJ et al (2017) Flow network based cardiac motion tracking leveraging learned feature matching. In: International conference on medical image computing and computer-assisted intervention. Springer, pp 279–286
Polvara R, Sharma S, Wan J, Manning A, Sutton R (2018) Obstacle avoidance approaches for autonomous navigation of unmanned surface vehicles. J Navig 71(1):241–256
Pronobis A, Caputo B (2009) COsy localization database. Int J Robot Res (IJRR) 28(5):588–594. https://doi.org/10.1177/0278364909103912
Mur-Artal R, Montiel JMM, Tardós JD (2015) ORB-SLAM: a versatile and accurate monocular SLAM system. IEEE Trans Robot. https://doi.org/10.1109/TRO.2015.2463671
Rituerto A, Puig L, Guerrero JJ (2010) Visual SLAM with an omnidirectional camera. In: 2010 20th International conference on pattern recognition. IEEE, pp 348–351
Rostkowska M, Skrzypczyński P (2023) Optimizing appearance-based localization with catadioptric cameras: small-footprint models for real-time inference on edge devices. Sensors 23(14):6485
Ruan X, Ren D, Zhu X, Huang J (2019) Mobile robot navigation based on deep reinforcement learning. In: 2019 Chinese control and decision conference (CCDC). IEEE, pp 6174–6178
Sakkos D, Shum HP, Ho ES (2019) Illumination-based data augmentation for robust background subtraction. In: 2019 13th International conference on software, knowledge, information management and applications (SKIMA). IEEE, pp 1–8
Sandouk U, Chen K (2017) Learning contextualized music semantics from tags via a siamese neural network. ACM Trans Intell Syst Technol 8(2):24
Schaupp L, Bürki M, Dubé R, Siegwart R, Cadena C (2019). OREOS: oriented recognition of 3d point clouds in outdoor scenarios. In 2019 IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE, pp 3255–3261

Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv preprint. arXiv:1409.1556

Sinha H, Patrikar J, Dhekane EG, Pandey G, Kothari M (2018) Convolutional neural network based sensors for mobile robot relocalization. In: 2018 23rd International conference on methods & models in automation & robotics (MMAR). IEEE, pp 774–779

Suljagic H, Bayraktar E, Celebi N (2022) Similarity based person re-identification for multi-object tracking using deep siamese network. Neural Comput Appl 34(20):18171–18182

Sun Y, Cheng C, Zhang Y, Zhang C, Zheng L, Wang Z, Wei Y (2020) Circle loss: a unified perspective of pair similarity optimization. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 6398–6407

Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov S, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1–9

Tan M, Le Q (2019) EfficientNet: rethinking model scaling for convolutional neural networks. In International conference on machine learning. PMLR, pp 6105–6114

Thiolliere R, Dunbar E, Synnaeve G, Versteegh M, Dupoux E (2015) A hybrid dynamic time warping-deep neural network architecture for unsupervised acoustic modeling. In: 16th annual conference of the international speech communication association

Utkin LV, Zaborovsky VS, Popov SG (2017) Siamese neural network for intelligent information security control in multi-robot systems. Autom Control Comput Sci 51(8):881–887

Wang Y, Bao T, Ding C, Zhu M (2017) Face recognition in real-world surveillance videos with deep learning method. In: 2017 2nd international conference on image, vision and computing (ICIVC). IEEE, pp 239–243

Weinzaepfel P, Csurka G, Cabon Y, Humenberger M (2019) Visual localization by learning objects-of-interest dense match regression. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 5634–5643

Xu S, Chou W, Dong H (2019) A robust indoor localization system integrating visual localization aided by CNN-based image retrieval with Monte Carlo localization. Sensors 19(2):249

Yin H, Tang L, Ding X, Wang Y, Xiong R (2018) LocNet: global localization in 3d point clouds for mobile vehicles. In: 2018 IEEE intelligent vehicles symposium (IV). IEEE, pp 728–733

Zaki HF, Shafait F, Mian A (2019) Viewpoint invariant semantic object and scene categorization with RGB-D sensors. Auton Robot 43(4):1005–1022

Zeng A, Song S, Yu KT, Donlon E, Hogan FR, Bauza M, Ma D, Taylor O, Liu M, Romo E et al (2018) Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching. In: 2018 IEEE international conference on robotics and automation (ICRA). IEEE, pp 3750–3757

Zhang Z, Peng H (2019) Deeper and wider Siamese networks for real-time visual tracking. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 4591–4600

Zhao Q, Zhang B, Lyu S, Zhang H, Sun D, Li G, Feng W (2018) A CNN-SIFT hybrid pedestrian navigation method based on first-person vision. Remot Sens 10(8):1229

Zheng W, Yang L, Genco RJ, Wactawski-Wende J, Buck M, Sun Y (2019) Sense: Siamese neural network for sequence embedding and alignment-free comparison. Bioinformatics 35(11):1820–1828

Zhong F, Wang S, Zhang Z, Wang Y (2018) Detect-SLAM: Making object detection and SLAM mutually beneficial. In: 2018 IEEE winter conference on applications of computer vision (WACV). IEEE, pp 1001–1010