Universidad Miguel Hernández de Elche

MÁSTER UNIVERSITARIO EN ROBÓTICA

"Estudio de Redes Neuronales Siamesas para la creación de modelos y localización de robots móviles"

Trabajo de Fin de Máster

Curso académico 2021-2022

Autor: Juan José Cabrera Mora

Tutores: Luis Payá Castelló
Arturo Gil Aparicio

Contenido

Lista de Figuras

Lista de tablas

1 Introducción

En la actualidad, los robots son una realidad con la que convivimos en el día a día aunque no seamos conscientes de ello. Por ejemplo, los podemos encontrar en el hogar desempeñando tareas de limpieza, en restaurantes sirviendo las mesas o en industrias que producen bienes consumidos por el público general.

La robótica está conformada por diferentes campos del conocimiento como lo son la ingeniería mecánica, eléctrica, electrónica, telecomunicaciones e informática. Algunas ramas de la robótica, como la robótica de manipuladores, pueden ser considerados una tecnología clásica en el sector industrial, donde las primeras soluciones robotizadas datan de los años 60. Otras ramas de la robótica, como los robots móviles, son una tecnología más emergente cuyas aplicaciones se encuentran principalmente en los sectores servicios y social.

En el sector industrial, es frecuente el uso de brazos robóticos industriales para conformar las diferentes cadenas de producción (Figura **1-1**). El inconveniente de dichos robots es que requieren estar aislados del personal que trabaja en la planta ya que pueden provocar graves daños. Es por ello que los robots colaborativos están cobrando una mayor fuerza ya que trabajan a velocidades menores y disponen de sensores que permiten evitar colisiones con las personas (Figura **1-2**). Estos robots están diseñados para la realización de tareas repetitivas, peligrosas y por tanto, poco recomendadas para las personas. En el sector servicios, se está comenzando a introducir robots móviles para desempeñar tareas en hostelería (Figura **1-3**), logística y transporte debido a la falta de personal (Figura **1-4**). En cuanto al sector social, existen empresas centradas en la creación de robots especialmente diseñados para la atención, cuidado y entretenimiento de las personas (Figura **1-5**).

Los robots autónomos móviles son capaces de realizar la tarea para la cual están diseñados al mismo tiempo que navegan de forma segura por el entorno. Este tipo de robots han demostrado ser aptos para la realización de una gran variedad de tareas altamente peligrosas, evitando así poner en riesgo a personas. Por ejemplo, se han empleado para llevar a cabo la exploración tanto de océanos (Figura **1-6**) como de otros planetas (Figura **1-7**). Más recientemente se han comenzado a extender a la resolución de tareas más rutinarias.

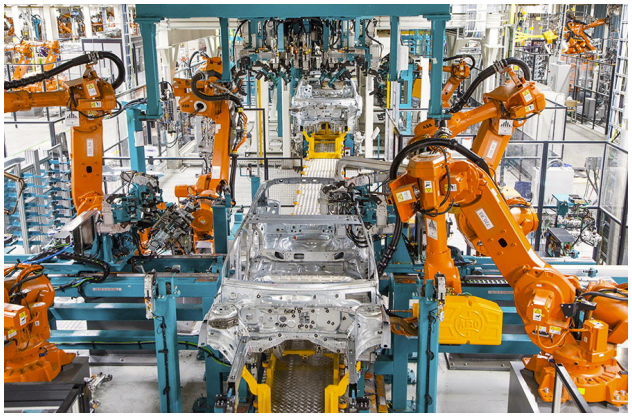


Figura 1-1: Robot industrial.

Figura 1-2: Robot colaborativo.

Figura 1-3: Robot camarero.

Figura 1-4: Robot Handle diseñado por Boston Dynamics para desempeñar tareas de transporte de cargas.

Figura 1-5: Robot social para hacer más entretenidas las terapias de rehabilitación.

Para que un robot sea capaz de navegar de forma segura por el entorno, debe crear un mapa del entorno que le permita navegar. En ocasiones, cuando no se dispone de una estimación sobre la posición/orientación del robot, se utilizará algoritmos de SLAM para realizar esta tarea. En este tipo de algoritmos, el robot debe crear un mapa y, al mismo tiempo, localizarse dentro de él. Cabe destacar que los entornos por los que navega el robot son dinámicos, y por tanto, el robot debe estar preparado para encontrarse ante situaciones cambiantes e imprevistas. Para llevar a cabo la creación del mapa, se puede optar por dos alternativas: mapas métricos o topológicos. Los mapas métricos contienen la información referente a la posición de los diferentes elementos que conforman el entorno respecto de un sistema de referencia global y con una determinada incertidumbre asociada. De este modo, el robot podrá estimar su pose respecto al sistema de referencia. En cuanto a los mapas topológicos, contienen información sobre ciertas zonas relevantes del entorno y las relaciones de conectividad entre ellas. En este caso, se trata de un tipo de representación adecuado para llevar a cabo la localización y la planificación de trayectorias en entornos extensos y tareas que no requieran conocer la localización con precisión métrica.

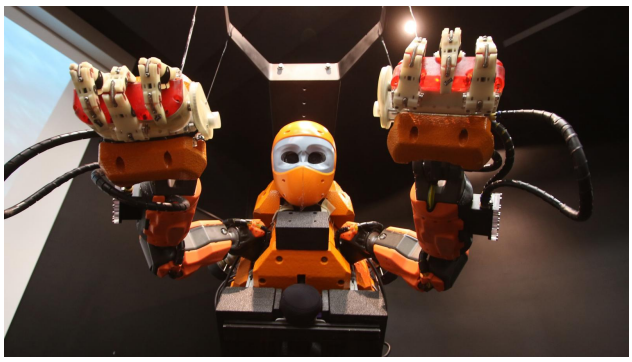


Figura 1-6: Robot submarino diseñado por la Universidad Stanford y denominado OceanOne.

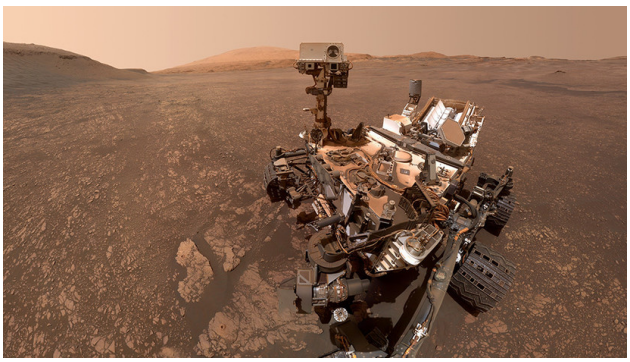


Figura 1-7: Robot Curiosity diseñado para la exploración de Marte.

Para poder realizar las tareas de mapping y localización, resulta necesario que el robot vaya equipado con sensores que permitan conocer el estado del entorno. A continuación, se describen los sensores más empleados en robótica móvil:

- **GPS**: La localización de cualquier receptor GPS se determina mediante una medición del tiempo de vuelo. Los satélites envían su ubicación orbital más el instante de tiempo en la que la enviaron, el receptor calcula su ubicación mediante una trilateración considerando los instantes temporales en los que los satélites enviaron su posición. Tiene como principal inconveniente el elevado error de posición (en torno a 4 metros para sistemas GPS comerciales y además no es uniforme). Además, en zonas cercanas a

edificios y/o árboles el error de localización aumenta, no pudiéndose realizar la localización en entornos de interior. Es por ello que se suele combinar con otro tipo de tecnologías como la odometría (Onho *et al.*, [28]).

- **GPS diferencial**: Este tipo de sistemas requiere de un receptor GPS situado en una estación base cuya posición sea conocida de forma precisa. El receptor de la estación base calcula su posición basándose en las señales de los satélites y compara esta ubicación con la conocida. La diferencia se aplica a los datos GPS registrados por el receptor GPS que incorpora el robot a localizar. De esta manera, se obtiene un error de localización del orden de los 10 centímetros.

- **Sensores de rango**: Son los sensores más empleados en robótica móvil pues permiten la creación de mapas, la localización y la detección y elusión de obstáculos. Son capaces de medir distancias a los elementos de la escena mediante la medición del tiempo de vuelo. Existen diferentes tipos de sensores en función del fenómeno físico que empleen:

  - **Láser**: Este sensor permite medir la distancia a los objetos mediante la medición del tiempo de vuelo del láser pulsado, el cual se trata de una onda electromagnética. La tecnología láser se incorpora en los sensores LiDAR (Light Detection And Ranging) que permiten realizar mediciones con un rango angular muy amplio. Por contrapartida, lo sensores láser no permiten detectar objetos trasparentes y pueden verse afectados por ciertas condiciones meteorológicas que provocan la dispersión de los rayos. Esta problemática se ha estudiado por Wolcott *et al.*, [47] que emplean métodos probabilísticos para lidiar con estos problemas en la presencia de fuertes nevadas.

  - **Infrarrojos**: Este sensor permite medir la distancia a los objetos mediante la medición del tiempo de vuelo de la luz infrarroja generada por un fototransistor o fotodiodo. Para medir distancia, algunos autores proponen medir el desplazamiento de fase de la señal recibida respecto de la enviada (Martín *et al.*, [24]).

  - **Ultrasonidos**: Este sensor permite medir la distancia a los objetos mediante la medición del tiempo de vuelo de una señal acústica. Los sensores de ultrasonidos permiten detectar objetos transparentes. Además, la temperatura del aire (y, por ende, su densidad) afecta sensiblemente las medidas realizadas, que deben corregirse cuando las temperaturas son extremas. Moreno *et al.*, [25] emplearon este tipo de sensores junto con un filtro no lineal para implementar un sistema de navegación integral.

- **Sensores de visión**: Las cámaras permiten obtener una gran cantidad de información del entorno. Por este motivo, se trata de uno de los sensores más versátiles que se pueden emplear en robótica móvil, pues permiten reconocer el entorno e identificar objetos en las escenas. Entre otras, podemos destacar:

- **Cámara estándar**: Se trata de cámaras monoculares o de una sola cámara. Al disponer de un solo punto de vista, no es posible determinar la profundidad a la que se encuentran los objetos. Pero sí que pueden ser empleadas para la obtención de información relevante de la escena. De esta forma, a partir de las imágenes capturadas se puede llevar a cabo una clasificación de los objetos que aparecen en la escena, reconocer a las personas u otro tipo de tareas similares útiles para la localización y mapping. Además, puesto que solo permiten cubrir una parte del entorno, se suelen combinar con otros sensores del mismo o de diferente tipo que permitan obtener una mayor cantidad de información. Por ejemplo, los automóviles de la empresa Tesla disponen de 8 cámaras, 12 sensores de ultrasonidos y un sensor radar. En cuanto a la creación de mapas, el uso de cámaras monoculares como única fuente de información solo permitiría la creación de mapas topológicos, y no métricos.

- **Cámara estereoscópica**: Se trata de sistemas de visión formados por dos cámaras. Este tipo de sensores han demostrado ser una alternativa robusta, al igual que los sistemas láser o sonar (Murray *et al.*, [26]). El principio de funcionamiento se basa en la búsqueda de correspondencias entre las dos imágenes capturadas. La búsqueda de correspondencias se realiza en base a métodos basados en la correlación de píxeles, métodos basados en características visuales puntuales (SURF, SIFT) o bien métodos basados en la proyección de un patrón conocido (Intel Realsense, Microsoft Kinect, etc)

- **Cámara omnidireccional**: Este tipo de cámaras permiten capturar imágenes con un ángulo de visión de 360º. De esta manera, con una sola imagen posibilitan la descripción completa del entorno tal y como es percibido desde un determinado punto de captura. En cambio, las cámaras estereoscópicas y las estándar requieren de múltiples imágenes que cubran diferentes puntos de vista y mediante reconstrucción 3D, podrían obtener una imagen panorámica que cubra los 360º. Otra de las ventajas de las cámaras omnidireccionales es que las imágenes que capturan contienen la misma información independientemente de la orientación del robot (según el robot gira sobre un determinado punto de captura, únicamente se produce un giro de la imagen omnidireccional o un desplazamiento circular de las columnas en caso de la imagen panorámica).

Por tanto, resulta una decisión acertada el empleo de sensores de visión para llevar a cabo tareas de navegación. Las imágenes requerirán ser procesadas por un ordenador, de manera que se obtenga la información relevante de la escena para desempeñar la tarea deseada. Este procesamiento conlleva un alto coste computacional que en muchas ocasiones se puede realizar en tiempo real gracias al aumento en la capacidad de procesamiento de los computadores actuales.

En el presente trabajo, se va a hacer uso de cámaras omnidireccionales para llevar a cabo la tarea de localización de un robot móvil. Entre las diferentes configuraciones de cámara omnidireccional, podemos destacar dos: sistema multicámara como el de la Figura **1-8** y sistema catadióptrico omnidireccional compuesto por una cámara estándar y un espejo hiperbólico (Figura **1-9**). Ésta última es la configuración empleada en el presente trabajo por tratarse de un modelo más sencillo.

Figura 1-8: Insta360 Titan.

Figura 1-9: Sistema catadióptrico omnidireccional.

Las cámaras omnidireccionales tiene un coste relativamente bajo en relación con la cantidad de información que ofrecen. Es por ello, que su uso resulta de especial interés ya que a diferencia de otro tipo de sensores como los LiDAR, permiten capturar la reflectividad, las diferentes texturas y colores de los elementos del entorno. Además, las imágenes capturadas por los sistemas omnidireccionales tienen un campo de visión de 360^{o} alrededor del sistema de visión, lo que las hace especialmente adecuadas para crear el modelo visual del entorno al requerir de una menor cantidad de imágenes.

El avance en la capacidad de cómputo de los ordenadores modernos ha propiciado el desarrollo de algoritmos basados en Inteligencia Artificial (IA). En la actualidad, esta tecnología es popular debido a su fácil adaptación a problemas de las diferentes áreas del conocimiento. De esta manera, se emplean sus algoritmos y métodos para llevar a cabo predicciones en bolsa, detección temprana de enfermedades en personas, decodificación del genoma humano, conducción autónoma de vehículos, controles de calidad en industrias, procesamiento del lenguaje natural y creación de imágenes a partir de texto, entre muchas otras aplicaciones.

En el presente trabajo se hace uso de Redes Neuronales Convolucionales, las cuales se encuentran enmarcadas dentro del aprendizaje profundo (subcampo de la Inteligencia Artificial). Las Redes Neuronales Convolucionales están inspiradas en el funcionamiento de las células nerviosas del cerebro. Cuando las personas adquieren nuevos conocimientos o destrezas, se modifican las conexiones existentes entre las neuronas del cerebro. Lo mismo ocurre

en las Redes Neuronales Convolucionales, cuando se adquieren nuevos conocimientos se modifican los pesos y umbrales que hacen de enlace entre las neuronas. En la Figura **1-10** se puede apreciar el esquema de una neurona artificial. La neurona tiene una serie de señales de entrada $x_1, x_2, ..., x_n$ las cuales llevan asociadas una serie de pesos que refuerzan o debilitan los enlaces $w_1, w_2, ..., w_n$. Además, al igual que las neuronas del cerebro no transmiten las señales de entrada hacia la siguiente neurona hasta que éstas sobrepasan un determinado potencial eléctrico, las neuronas artificiales no transmiten la información a la neurona siguiente hasta que la suma de los valores de entrada multiplicados por sus pesos no superan un cierto umbral denominado bias w_0. De esta manera, las señales con un mayor peso asociado tienen una mayor contribución y si la suma ponderada de las entradas supera el bias, ésta se introduce en la función de activación obteniendo así la salida de la neurona y.

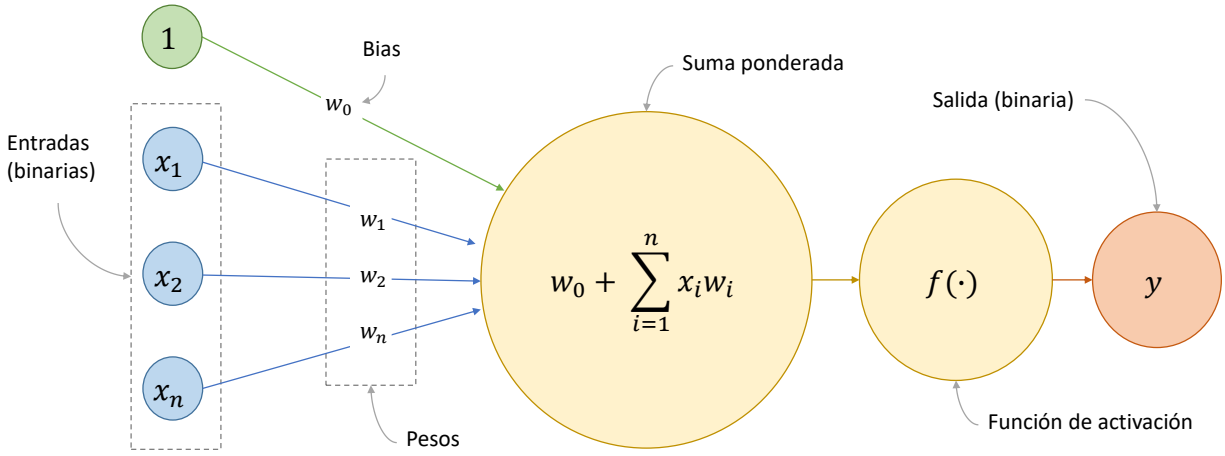


Figura 1-10: Esquema básico de una neurona artificial.

Todos los elementos mencionados anteriormente son aprendidos o ajustados por la red neuronal a medida que va evolucionando el entrenamiento de las mismas salvo la función de activación, cuya función es establecida por el diseñador. Entre las funciones de activación más comunes, podemos destacar las siguientes:

- **Función sigmoide:** La función sigmoide tiene forma característica de "S", representa la progresión temporal de sistemas complejos pasando de unos niveles bajos al inicio a un valor estable al final mediante una transición con una fuerte aceleración en forma

de “S”. De esta manera, lo valores de salida tras aplicar la función sigmoide están comprendidos entre 0 y 1.

- **Tanh (Tangente hiperbólica):** Al igual que la función sigmoide, tiene forma de “S”, pero los valores de salida están comprendidos entre -1 y 1. Esta función es el resultado del cociente del seno hiperbólico y el coseno hiperbólico del valor de entrada.

- **ReLu (Rectified Linear Unit):** Es una función a trozos de manera que tiene como salida el valor 0 si el valor de entrada es negativo y tiene como salida el propio valor de entrada si éste es positivo. En la Figura **1-11**, se muestran las funciones de activación sigmoide, tangente hiperbólica y ReLU.

- **SoftMax:** Se emplea en la última capa de la red neuronal cuando se quiere llevar a cabo una clasificación. Calcula la probabilidad de cada clase y toma como salida la clase que toma el valor máximo. Se trata de una función exponencial normalizada.

Las neuronas se organizan en capas y en función de cómo están conectadas, del número de neuronas por capa y del orden de las mismas existen infinidad de arquitecturas diseñadas para tareas totalmente distintas. De forma general, toda red posee una capa de entrada, una serie de capas ocultas y una capa de salida. Las neuronas de la capa de entrada no poseen pesos asociados a ellas, simplemente se encargan de transmitir la información de entrada a las capas ocultas. Por el contrario, las capas ocultas y de salida si que llevan asociado una serie de pesos, los cuales son modificados durante el entrenamiento con el fin de que la red sea capaz de desempeñar la tarea objetivo.

En los inicios de la Inteligencia Artificial, las redes neuronales disponían de una sola capa (Figura **1-13**). Posteriormente, se empezaron a introducir más capas dando lugar a las redes neuronales poco profundas (Figura **1-14**) con un escaso número de capas. En la actualidad, existen redes neuronales con decenas de capas ocultas (*hidden layers*) dando lugar a las redes neuronales profundas (Figura **1-15**).

Recientemente, se ha propuesto el uso de Redes Neuronales Siamesas para llevar a cabo tareas de reconocimiento facial (Wu *et al.*, [50]). Estas redes están compuestas a su vez por dos Redes Neuronales Convolucionales con la misma arquitectura y con los pesos y umbrales compartidos (Figura **1-12**). Esto permite tomar dos conjuntos de datos como entrada y generar una función de diferencia entre los mismos. En el presente trabajo, se propone emplear Redes Neuronales Siamesas para llevar a cabo la localización a partir de las imágenes omnidireccionales capturadas por un robot autónomo móvil que navega a lo largo de un entorno de interior bajo diferentes condiciones de iluminación.

Además, se plantea abordar el problema de localización bajo dos puntos de vista diferentes:

- **Localización global**: Consiste en estimar las coordenadas desde las que el robot capturó la imagen en un único paso. De esta manera se aborda el problema de localización de forma bruta sin dividirlo en subtareas, lo que repercute en el tiempo de cómputo.
- **Localización jerárquica**: Consiste en estimar las coordenadas desde las que el robot capturó la imagen en dos pasos. El primer paso se denomina localización gruesa y tiene como objetivo identificar la estancia en la que se encuentra el robot. El segundo paso se denomina localización fina y se lleva a cabo una búsqueda restringida a la estancia predicha anteriormente, de manera que se obtienen las coordenadas en las que se encuentra el robot.

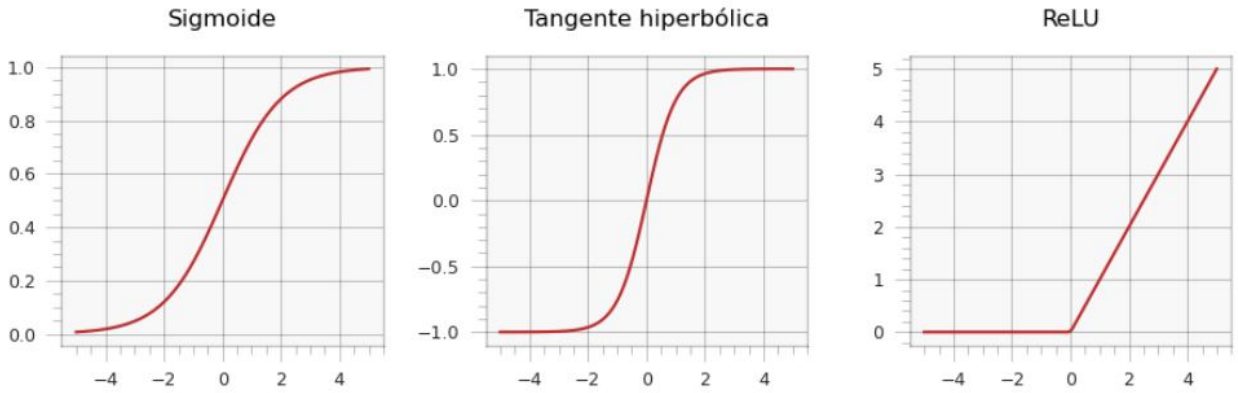


Figura 1-11: Representación de las funciones de activación más comunes.

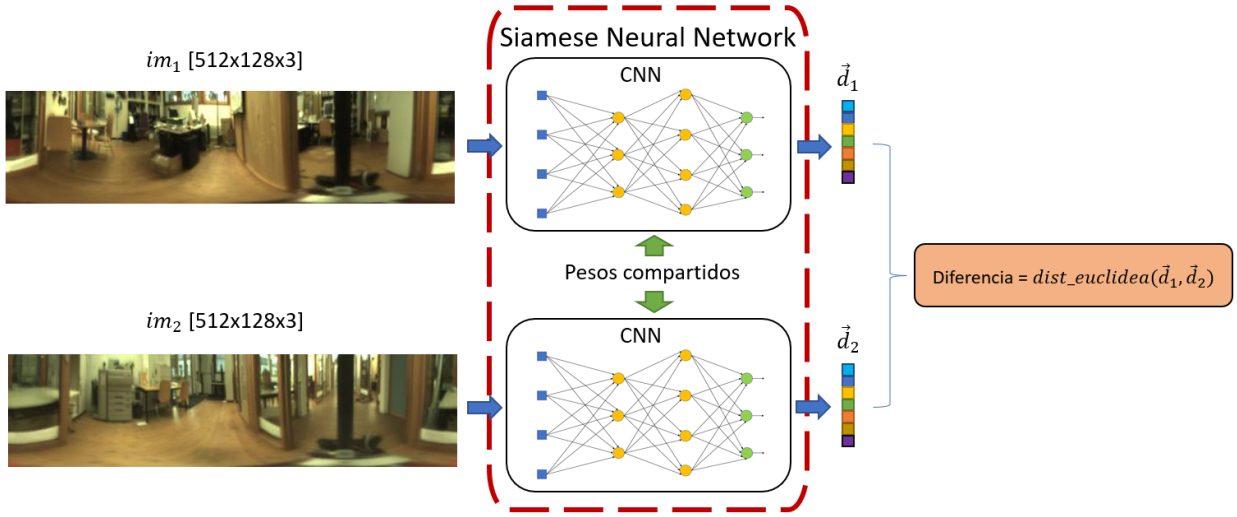


Figura 1-12: Arquitectura general de una Red Neuronal Siamesa.

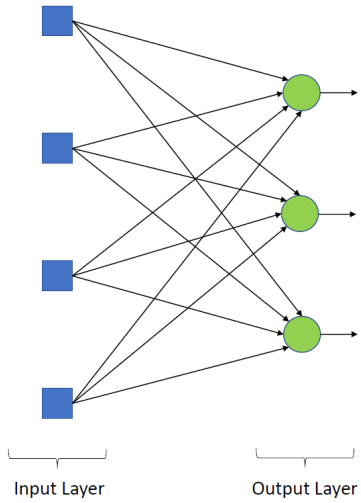


Figura 1-13: Red Neuronal de una sola capa.

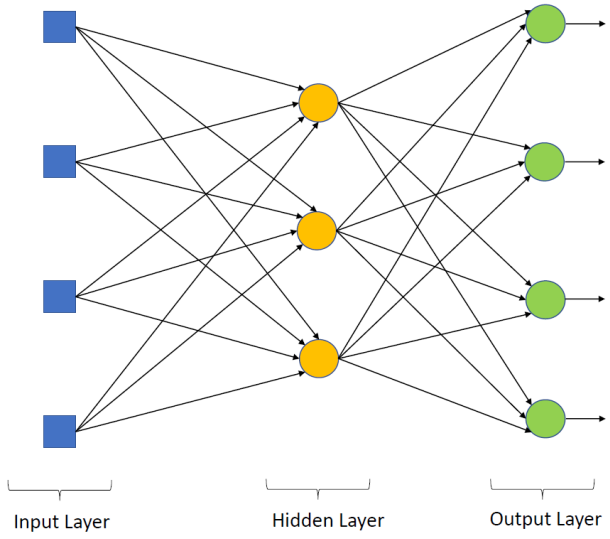


Figura 1-14: Red Neuronal Multicapa (Poco profunda).

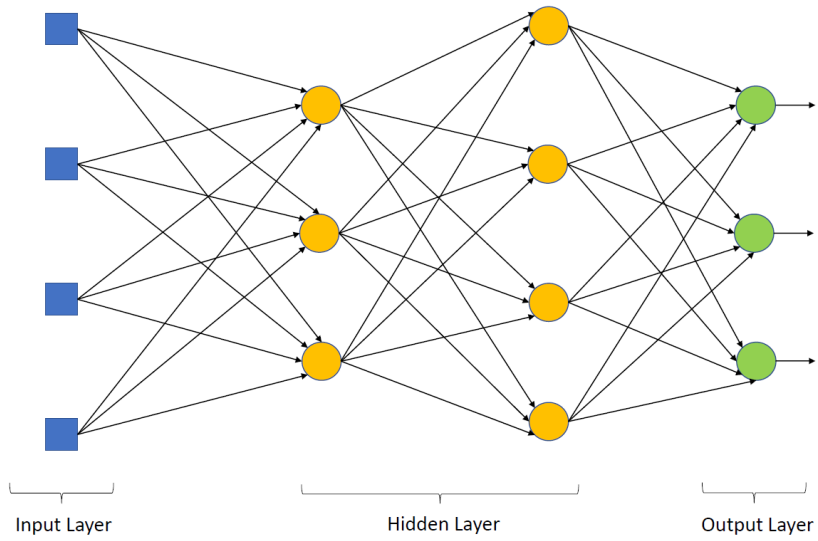


Figura 1-15: Red Neuronal Profunda.

El resto de la memoria tiene la siguiente estructura, la sección 2 contiene el estado del arte en el que se realizan referencias a investigaciones previas relacionadas con la localización de robots autónomos móviles. La sección 3 muestra las diferentes herramientas que se han empleado en el presente trabajo. En concreto, se detallan en profundidad conceptos introducidos anteriormente como la visión omnidireccional, las capas que componen una CNN, las diferentes arquitecturas empleadas, los descriptores de apariencia global, el aumento de datos y la base de datos empleada. En la sección 4, se describirá el método propuesto para

abordar la localización jerárquica y/o global mediante el empleo de Redes Neuronales Siamesas. En la sección 5, se expondrán los resultados obtenidos para los diferentes experimentos realizados y en la última sección, se comentarán las diferentes conclusiones extraídas y se propondrán nuevas líneas de investigación para abordar en un futuro.

2 Estado del arte

En los últimos años, se ha incrementado sustancialmente el despliegue de robots autónomos móviles debido a su capacidad de automatizar procesos y/o realizar tareas sin poner en riesgo a personas. Para que un robot móvil pueda realizar una tarea autónomamente, en un entorno a priori desconocido, debería de ser capaz de crear un modelo de este entorno (mapa) que le permita estimar su pose (posición y orientación) de forma suficientemente precisa. Además, para poder navegar, deberá ser capaz de detectar aquellos obstáculos u objetos que puedan provocar colisiones.

2.1. Creación de mapas y localización de robots móviles

La creación de mapas y la localización son dos cuestiones de estudio en el ámbito de la robótica móvil y una solución robusta a estas tareas permitiría abordar el desarrollo de otras misiones de alto nivel, como podrían ser la exploración, planificación de trayectorias y navegación integrada. Tanto el mapping como la localización han sido y continúan siendo, dos áreas de trabajo muy activas, sobre las que se han planteado multitud de soluciones, permitiendo así la construcción de mapas con varios tipos de sensores (Kim *et al.*, [18]), múltiples grados de libertad (Rebecq *et al.*, [33]), organización jerárquica de la información en el modelo (Ruiz-Sarmiento *et al.*, [35]) e inclusión de información semántica (Sualeh *et al.*, [38]). Además, los problemas de mapeo y de localización se pueden resolver de manera conjunta mediante la técnica de SLAM (*Simultaneous Mapping And Localization*). Esta técnica considera el caso en el que no existe ninguna información previa sobre el entorno por el que se mueve el robot y éste debe construir el mapa, mientras, al mismo tiempo, se localiza dentro de él (Aulinas *et al.*, [3]). Son muchos los autores que han realizado contribuciones en el estado del arte de dicha técnica mediante aproximaciones probabilísticas (Thrun *et al.*, [45]). Uno de los aspectos que se debe tener en cuenta al crear modelos mediante esta técnica son las inconsistencias que se pueden producir cuando el tamaño del entorno es grande o se produce el fenómeno de aliasing.

La creación de mapas se ha abordado tanto en entornos de interior como de exterior, incorporando tanto información 2D como 3D y contemplando condiciones de operación reales como es el caso de modificaciones en la apariencia del entorno. Cuando esto ocurre, resulta de interés la identificación de aquellas personas y/u objetos que aparecen de forma esporádica en la escena y que no deberían de conformar el modelo o mapa del entorno (Hahnel *et*

al., [16]). En este sentido, Wolf *et al.*, [48] desarrollaron un modelo del entorno compuesto por tres capas, la primera de ellas contiene la información estática, la segunda la dinámica y la tercera, puntos característicos estáticos que permitan llevar a cabo la localización del robot. Mediante este método, consiguen una descripción del entorno robusta ante cambios dinámicos.

En cuanto a la navegación, cuando un robot móvil ha de moverse de forma autónoma en entornos no estructurados, debe hacer frente a situaciones imprevistas y de gran dinamismo (Kemp *et al.*, [17]), especialmente cuando se encuentra en exteriores donde las condiciones son más desafiantes debido a áreas más amplias, cambios meteorológicos y cambios de estación. Para resolver el problema de navegación, se ha de conocer de forma precisa la ubicación del robot. Para ello, existen sistemas de localización basados en Global Navigation Satellite System (GNSS), pero tienen como inconveniente que el error de localización que llevan asociado es mayor al habitualmente requerido por los algoritmos de control de la navegación (Tao *et al.*, [43]). Una alternativa son los sistemas GPS/GNSS con corrección del error, denominados RTK-GPS, pero no siempre son empleables, pues precisan de la existencia de un canal de comunicaciones directo con una estación base.

Por otro lado, otros investigadores han optado por tecnología láser para hacer frente al problema de localización y navegación (Zhang *et al.*, [52]). En este sentido, los sensores LiDAR (Light Detection and Ranging) son ampliamente empleados para desempeñar tareas propias de robots móviles. Por ejemplo, Sprunk *et al.*, [37] realizaron una comparativa entre los sistemas LiDAR y sistemas de visión estereoscópica para desarrollar tareas en aplicaciones de robótica agrícola. Continuando con los sensores de rango, la tecnología SONAR (Sound Navigation And Ranging) también se suele emplear con frecuencia para llevar a cabo la localización y creación de mapas. En concreto, Feder *et al.*, [11] emplearon esta tecnología para desarrollar un mapeo estocástico haciendo uso de un filtro de Kalman.

En cuanto a los sensores de visión, han sido ampliamente empleados para tareas de mapeo, localización y navegación. Además, es posible combinar este tipo de sensores con otros, por ejemplo, Ohya *et al.*, [29] hicieron uso de una cámara estándar combinada con un sensor de ultrasonidos para desempeñar la tarea de navegación de un robot móvil. Por otro lado, Gallegos *et al.*, [13] emplearon una cámara omnidireccional y un sensor láser 2D para abordar la tarea de SLAM. Otros autores han combinado sistemas LiDAR junto con cámaras RGB-D para desempeñar la tarea de *sense and avoid* (Gatesichapakorn *et al.*, [15]).

No obstante, el rendimiento de la mayoría de las soluciones decrece sustancialmente cuando el robot navega por entornos más complejos y en condiciones de trabajo reales y desafiantes (Garcia-Fidalgo *et al.*, [14]). Por este motivo, resulta de especial interés realizar un estudio sobre la localización de robots móviles en entornos de interior, donde no siempre se dispone

de información GPS y además, la presencia de personas y otros elementos dinámicos puede provocar situaciones inesperadas con las que el robot debe ser capaz de lidiar.

2.2. Descripción de escenas

Tal y como se ha mencionado anteriormente, para que un robot pueda construir un modelo del entorno y estimar su posición, éste debe ir equipado con sensores que permitan conocer el estado del entorno y obtener información relevante del mismo. En este sentido, resulta especialmente adecuado el uso de sensores visuales para resolver estas tareas debido al gran volumen de información que capturan. En concreto, el uso de cámaras omnidireccionales junto con técnicas de visión por computador ha demostrado ser una alternativa sólida para abordar la tarea de localización en robótica móvil (Payá *et al.*, [32]). Este tipo de cámaras tienen un campo de visión de 360 grados y un coste relativamente bajo en comparación con otros tipos de sensores. En investigaciones previas, se han instalado en vehículos que no solo desempeñan las tareas mencionadas anteriormente, sino que también estiman con alta precisión la trayectoria del automóvil (Tardif *et al.*, [44]). Por otro lado, Kuutti *et al.*, [20] evaluaron técnicas basadas en sistemas de visión para la localización de vehículos autónomos.

Cuando trabajamos con información visual para resolver las tareas de mapping y localización, resulta necesaria la extracción de información relevante de las imágenes capturadas, de manera que permitan al robot estimar su pose (posición y orientación). Esta extracción de características se emplea para describir las escenas tanto de forma local como global. Por un lado, las características locales se basan en la extracción de regiones características de la escena, como es el caso de los puntos, bordes y esquinas. Cada característica de la escena está descrita mediante un vector descriptor, por lo que la escena está representada por un conjunto de puntos o regiones características junto con sus respectivos descriptores. Por ejemplo, Andreasson *et al.*, [2] modificaron el algoritmo SIFT (*Scale-Invariant Feature Transform*) de Lowe *et al.*, [23] para llevar a cabo la localización de un robot móvil mediante correspondencias de puntos característicos. Por otro lado, los descriptores de apariencia global u holísticos permiten describir la información global de la escena mediante un único vector descriptor. Payá *et al.*, [30] demostraron que los descriptores holísticos pueden ser empleados para llevar a cabo la localización y creación de mapas. En la literatura relacionada se pueden encontrar numerosos trabajos que proponen métodos analíticos para la obtención de ambos tipos de descriptores.

Por otro lado, la información relevante de una escena también se puede extraer mediante algoritmos basados en Inteligencia Artificial (IA). Este campo se ha visto beneficiado por la mejora del rendimiento de los computadores lo que ha propiciado el desarrollo de nuevos algoritmos y modelos, los cuales se han aplicado a diferentes áreas del conocimiento y entre otras, a la robótica móvil (Cebollada *et al.*, [7]). Recientemente, se ha extendido el uso

de sensores visuales junto con CNNs (Convolutional Neural Networks) para abordar tareas propias de robots móviles.

2.3. Aprendizaje profundo

Las CNNs ocupan un lugar importante dentro de las diferentes técnicas pertenecientes a la Inteligencia Artificial y aunque parecen una tecnología reciente, fueron introducidas por primera vez en 1943 por McCulloch y Pitts en un artículo que explicaba el modelo computacional de la actividad neuronal del cerebro. En 1949, Hebb continuó con las líneas de McCulloch y Pitts mediante el trabajo "*The organization of behavior: A neuropsychological theory*". Más tarde, en 1958, Rosenblatt [34] crea el primer algoritmo de aprendizaje denominado "Perceptrón". Posteriormente, en 1969 Minsky y Papert estudiaron ciertos aspectos clave de las redes neuronales simples mediante el libro "*Perceptrons: An introduction to computational geometry*". En 1975, Paul Werbos introdujo la retropropagación del error para desempeñar el entrenamiento de las redes neuronales y en 1990 consiguió un desarrollo avanzado de este método (Werbos, [46]).

A lo largo de los últimos años se han desarrollado multitud de redes neuronales con diferentes propósitos. En 1998, Yann LeCun *et al.*, [21] consiguieron el reconocimiento de números manuscritos mediante una pionera red neuronal compuesta por 8 capas y denominada LeNet-5. En la actualidad, existen numerosas redes las cuales han sido entrenadas con un elevado número de imágenes y cuyos modelos preentrenados son de acceso libre. Algunas de estas redes son AlexNet la cual fue entrenada para llevar a cabo la clasificación entre 1000 objetos diferentes (Krizhevsky *et al.*, [41]), GoogLeNet que fue diseñada para clasificar entre 1000 objetos al igual que AlexNet. En cambio, con GoogLeNet se aumenta el número de capas de 8 a 22, pero se reduce el número de parámetros de 60 millones a 4 millones (Szegedy *et al.*, [40]).

Los modelos preentrenados de la mayoría de redes son de acceso libre y sus conocimientos (pesos y umbrales) se pueden emplear como punto de partida con la finalidad de conseguir un objetivo distinto al del modelo preentrenado. A esta técnica se le conoce como *Transfer Learning* pues permite conseguir resultados mucho más robustos, en un menor tiempo de computación. Por ejemplo, Wozniak *et al.*, [49] tomó como punto de partida los pesos y la arquitectura de la red neuronal VGG-F, para reentrenarla con el propósito de clasificar entre 16 habitaciones. De esta manera, el *Transfer Learning* permite resolver dos problemas dentro del *Deep Learning*. El primero, es la escasez de datos de entrenamiento y el segundo problema, son los altos tiempos de computación necesarios para entrenar un modelo.

En cuanto al uso de Redes Neuronales para llevar a cabo la tarea de localización, Tai *et al.*, [42] entrenaron una CNN para abordar el problema de la navegación evitando el

impacto con obstáculos, Amer *et al.*, [1] llevan a cabo una localización de UAVs en zonas urbanas, Sandino *et al.*, [36] también hicieron uso de UAVs y de IA para llevar a cabo el mapping aéreo de bosques afectados por diferentes patógenos. Xu *et al.*, [51] proponen un sistema de localización global en interiores basado en el entrenamiento de una CNN. Por otro lado, Chen *et al.*, [8] emplean una CNN para la extracción de vectores descriptores. La descripción del entorno mediante la extracción de descriptores de apariencia global también ha sido propuesta por otros autores. Payá *et al.*, [31] hicieron uso de vectores descriptores holísticos con el fin de desempeñar una localización jerárquica. Cebollada *et al.*, [6] y Cabrera *et al.*, [4] propusieron entrenar una CNN con un doble propósito. El primero de ellos era la identificación de la estancia en la que el robot capturó la imagen por medio de la CNN de clasificación. El segundo propósito consistía en emplear esa misma red, para extraer un descriptor de apariencia global a partir de una de sus capas con el objetivo de determinar la pose del robot dentro de la estancia previamente seleccionada como una búsqueda del vecino más cercano.

En cuanto al proceso de entrenamiento de las CNNs, las herramientas de aprendizaje profundo requieren de un gran conjunto de datos para obtener modelos lo suficientemente robustos. Sin embargo, en algunos casos, el conjunto de datos disponible para el entrenamiento es pequeño y, entonces, el modelo no puede ser entrenado correctamente. Entre las técnicas propuestas para abordar este problema, el presente trabajo se centra en el aumento de datos la cual permite mejorar el rendimiento del entrenamiento del modelo aumentando el número de instancias de entrenamiento y evitando el sobreajuste. El aumento de datos consiste básicamente en crear nuevos datos (en este caso, imágenes) aplicando diferentes efectos sobre los originales (Ding *et al.*, [10]).

Por último, recientemente se han propuesto arquitecturas novedosas como las Redes Neuronales Siamesas. Mientras que las CNNs convencionales toman una imagen como dato de entrada, las Redes Neuronales Siamesas ofrecen la posibilidad de recibir más de una imagen de entrada, lo que les permite aprender relaciones de similitud o diferencia entre pares de imágenes. Si bien surgieron con el propósito de abordar el reconocimiento de caras (Wu *et al.*, [50]), su arquitectura las hace especialmente interesantes para abordar la tarea de localización (Leyva-Vallina *et al.*, [22]) o el tracking de objetos (Zhang *et al.*, [53]).

3 Herramientas usadas

En la presente investigación se han empleado diferentes herramientas y métodos como son la Visión omnidireccional, redes neuronales convolucionales como AlexNet o VGG16, el *Transfer Learning* y el *Data Augmentation*.

3.1. Visión omnidireccional

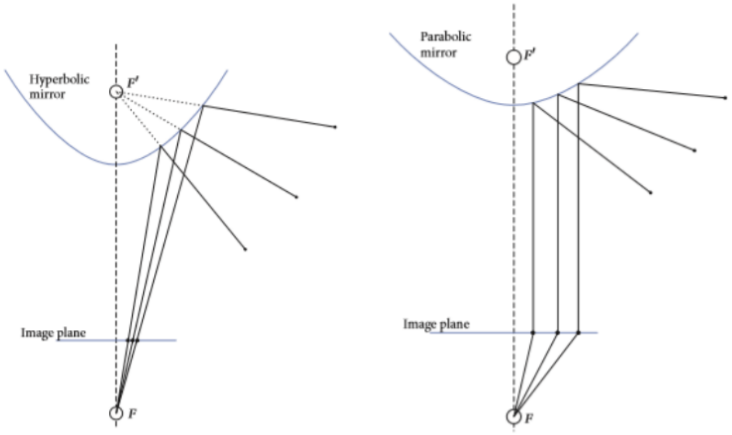
Los sensores de visión permiten extraer una gran cantidad de información del entorno por lo que son ampliamente empleados para resolver tareas propias de la robótica móvil. Además, el coste económico de este tipo de sensores es sustancialmente menor comparado con otro tipo de sensores empleados en robótica

En concreto, las cámaras omnidireccionales presentan además otra ventaja como la obtención de información de los 360^{o} que envuelven al robot en una sola imagen. Esta característica las hace especialmente idóneas para desempeñar la tarea de localización ya que permiten disminuir el número de imágenes que conforman el modelo visual y por tanto, reducir el coste computacional. Además, este tipo de cámaras capturan la misma información independientemente de la orientación de robot.

Se pueden encontrar diferentes configuraciones de sensores de visión omnidireccionales, entre los más destacados se encuentra el sistema catadióptrico debido a su gran simplicidad. La reflexión de la luz que incide sobre el espejo cóncavo permite la creación de imágenes omnidireccionales empleando una única cámara. La curvatura del espejo puede ser hiperbólica (Figura **3-1** (a)) o parabólica (Figura **3-1** (b)) en función de si la lente de la cámara es convencional o ortográfica.

Existen una gran cantidad de trabajos propuestos en la literatura en los que se hace uso de cámaras omnidireccionales. Por ejemplo, Cebollada *et al.*, [5] entrenaron una CNN con un doble propósito: 1) Determinar la estancia en la que se encuentra el robot por medio de una tarea de clasificación y, 2) La obtención de vectores descriptores de apariencia global que caractericen a las imágenes de entrada. Al igual que en el presente trabajo, Cebollada *et al.*, [5] abordaron la tarea de localización mediante dos enfoques, globalmente y jerárquicamente. Para llevar a cabo la localización global hicieron uso del vecino más cercano comparando el vector descriptor de la imagen capturada por el robot con los vectores descriptores de las

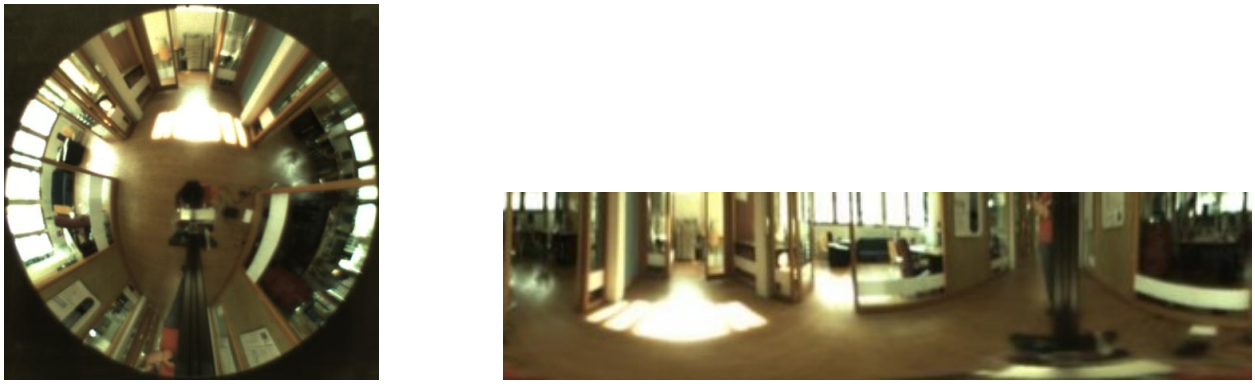
imágenes que conforman el modelo visual. En cuanto a la localización jerárquica, la imagen capturada por el robot es clasificada por la CNN y posteriormente, se compara su vector descriptor con los correspondientes al modelo visual de la habitación predicha, empleando también la técnica del vecino más cercano.



(a) (b)

Figura 3-1: Tipos de espejo para el sistema catadióptrico: (a) Espejo hiperbólico, (b) Espejo parabólico.

Es una práctica común la transformación de las imágenes omnidireccionales a panorámicas (Figura **3-2**). En el presente trabajo, considerando los resultados que hemos obtenidos en investigaciones anteriores ([5], [4]) se ha decidido trabajar con imágenes omnidireccionales convertidas a panorámicas por el mejor desempeño de localización que presentan.



(a) (b)

Figura 3-2: (a) Imagen omnidireccional sin conversión a panorámica, (b) Imagen omnidireccional convertida a panorámica.

3.2. Base de datos COLD

Las imágenes utilizadas en el presente trabajo se obtuvieron del conjunto de datos de Friburgo, incluido en la base de datos COLD (COsy Localization Database) la cual está constituida por imágenes omnidireccionales que se caracterizan por tener 3 capas (RGB) de 470x470 píxeles. Como se ha mencionado anteriormente, dichas imágenes se han convertido a panorámicas pasando a tener un tamaño de 512x128x3 píxeles. Dicha base de datos presenta imágenes capturadas en diversas trayectorias y bajo diferentes condiciones de iluminación (Figura **3-3**).

Las imágenes que componen la base de datos han sido capturadas mediante un sistema catadióptrico omnidireccional el cual va incorporado sobre un robot móvil (Figura **3-4**). Dicho robot circula por las diferentes estancias del edificio cuya planta se muestra en la Figura **3-5**. En el presente trabajo se ha empleado la secuencia 2 correspondiente a la ruta roja mostrada en la Figura **3-5**, con el objetivo de llevar a cabo la localización en un entorno con el máximo número de estancias. Por otro lado, el conjunto de datos escogido para llevar a cabo los entrenamientos cuenta con 8486 imágenes y está compuesto por imágenes capturadas bajo las tres condiciones de iluminación (seq2_cloudy3, seq2_night1, seq2_sunny3) que poseen 2778, 2876 y 2832 imágenes respectivamente. En cuanto a los conjuntos de datos para llevar a cabo el testeo se han escogido otras secuencias correspondientes a la misma trayectoria pero en instantes temporales diferentes. De esta forma, conformamos 3 conjuntos de testeo, cada uno correspondiente a una condición de iluminación diferente a partir de las secuencias seq2_cloudy2, seq2_night2, seq2_sunny2 que poseen 2595, 2707, 2114 imágenes respectivamente. Por otro lado, se escoge el dataset de nublado perteneciente al entrenamiento (seq2_cloudy3) y se muestrea para conformar el modelo visual con respecto al cual se llevará a cabo la localización y consta de 556 imágenes. El motivo por el cual se ha seleccionado la condición de iluminación nublada para el modelo visual del entorno es que se trata de una condición de iluminación neutral que permite visualizar un mayor nivel de detalle en las imágenes debido a un menor contraste entre el interior y el exterior del edificio. La separación media entre imágenes consecutivas seleccionada para el modelo visual es aproximadamente 40 centímetros con el fin de ser comparable con otros trabajos realizados previamente que siguen una distribución en forma de rejilla con una distancia entre imágenes similar. Este mapeado de rejilla fue realizado por el grupo de investigación ARVC en el edificio Innova de la Universidad Miguel Hernández de Elche.

El edificio de la base de datos empleada en el presente trabajo consta de nueve estancias. La estancia de mayor longitud y por tanto, con mayor número de imágenes, es el pasillo el cual hace de vinculo de unión con el resto de estancias. En la Tabla **3-1** se muestra el número de imágenes por estancia que conforma el modelo visual o mapa.

(a) (b) (c)

Figura 3-3: Ejemplo de imágenes pertenecientes a la base de datos COLD para las tres condiciones de iluminación: (a) nublado, (b) soleado y (c) noche. Las imágenes contenidas en este dataset pueden ser descargadas a través de su página web https://www.cas.kth.se/COLD/.

Figura 3-4: Ejemplo de robot móvil autónomo. Robot Pioneer P3-AT equipado con sensores SONAR, Láser 2D y sistema de visión catadióptrico.

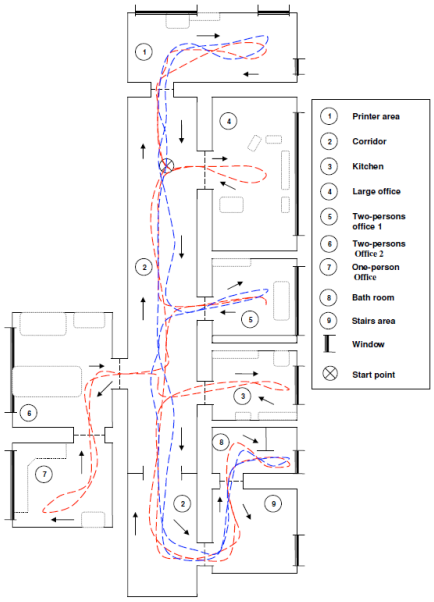


Figura 3-5: Planta edificio Friburgo de la base de datos COLD.

Estancia	1P0-A	2P01-A	2P02-A	CR-A	KT-A	LO-A	PA-A	ST-A	TL-A
Nº imágenes	44	46	31	238	46	26	57	30	38

Tabla 3-1: Número de imágenes por habitación que conforman el modelo visual.

3.3. Capas de una Red Neuronal Convolucional

El presente trabajo propone llevar a cabo la tarea de localización visual de un robot móvil por medio de Redes Neuronales Siamesas. Este tipo de redes están compuestas a su vez por dos Redes Neuronales Convolucionales idénticas, por lo que pueden ser consideradas como una variante de las Redes Neuronales Convolucionales tradicionales. Las redes neuronales se encuentran enmarcadas dentro de la rama del Deep Learning ya que toman de partida datos en bruto y extraen las características de los mismos para después decodificar la información y llevar a cabo una predicción. Existen diferentes capas necesarias para extraer dichas características y procesar la información de forma correcta. A continuación, se indican aquellas capas que suelen tener todas la redes neuronales convolucionales:

- **Convolutional Layer (Capa Convolucional):** Se trata de la capa más importante que conforma una Red Neuronal Convolucional puesto que se encarga de extraer las diferentes características presentes en las imágenes de entrada de la red. Para ello, se realiza la operación de convolución entre la imagen y una máscara de un tamaño determinado MxM (kernel size) y sobre unos determinados píxeles de la imagen (stride). Cuando se pasa la máscara sobre la imagen, se obtiene un mapa de características que resalta algún tipo de información presente en la imagen como los bordes y las esquinas.

- **Fully Connected Layer (FC):** En castellano se conocen como Capas Totalmente Conectadas y su función principal es la de transformar un mapa de características en un vector. Además, tal y como su nombre indica, conectan todas la neuronas de una capa con las neuronas de la capa siguiente mediante alguna operación matemática y se suelen emplear en la fase de clasificación de la red.

- **Dropout:** La tarea principal de esta capa es la de ignorar una parte aleatoria de las neuronas que le llegan como entrada con el objetivo de ahorrar tiempo de entrenamiento simplificando el modelo de la red. Cada neurona lleva asociados una serie de pesos a optimizar, con lo que al eliminar neuronas de dicha optimización permite agilizar el proceso de entrenamiento. Además, otra ventaja de emplear este tipo de capas es que permiten reducir el sobreajuste o sobreentrenamiento del modelo.

- **Batch normalization:** También se conoce como Normalización por lotes. Se encarga de normalizar las entradas de la capa posterior a ésta de manera que permite estabilizar el proceso de aprendizaje o entrenamiento del modelo. Además, permite reducir el número de iteraciones necesarias para llevar a cabo el correcto entrenamiento de la red.

- **Pooling Layer:** El propósito de esta capa es reducir las dimensiones del mapa de características, generando de esta forma mapas de características de menor tamaño con el fin de reducir el coste computacional. Para reducir los mapas, se agrupan las

neuronas en bloques de tamaño NxN y se opera de manera independiente en cada agrupación con el fin de limitar las conexiones entre las neuronas de una capa y su posterior. En función del método de pooling empleado, existen diferentes tipos de operaciones (Figura **3-6**):

- Max Pooling: Para cada agrupación, se escoge el mayor valor como salida y se desprecia el resto.
- Average Pooling: Para cada agrupación, se escoge el valor medio de los elementos del bloque como salida.
- Sum Pooling: Para cada agrupación, se escoge la suma de los elementos del bloque como salida.

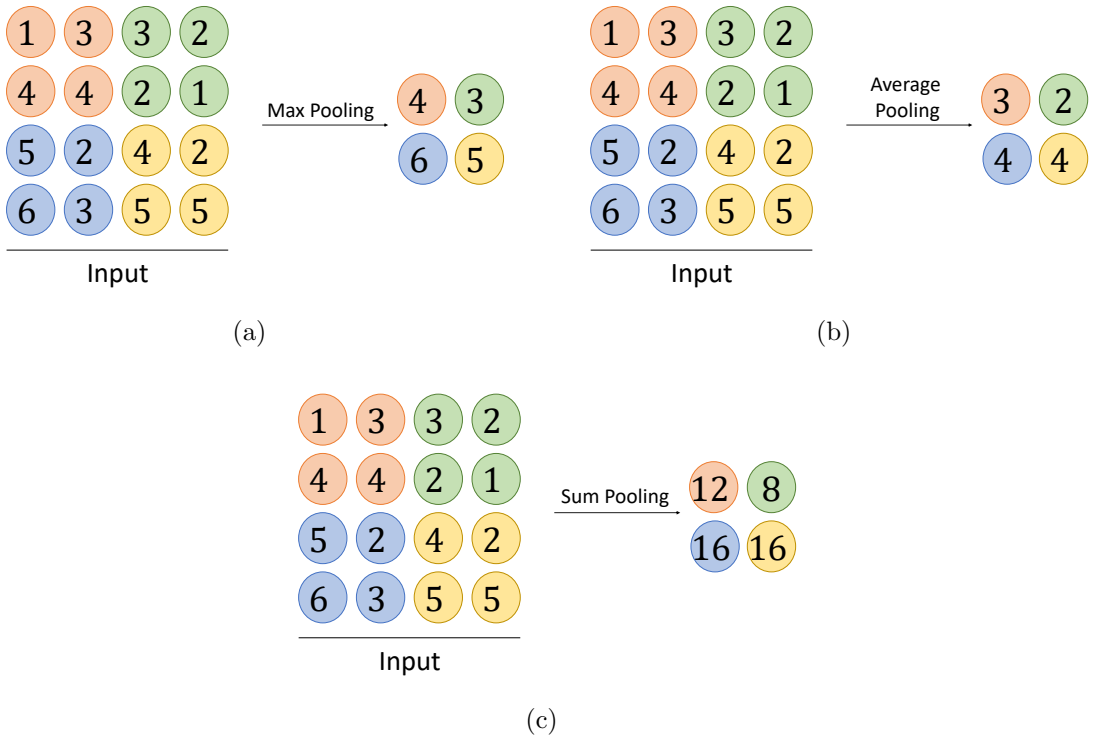


Figura 3-6: Ejemplo de operaciones de Pooling: (a) Max Pooling, (b) Average Pooling y (c) Sum Pooling.

Dependiendo del número, tipo y secuencia de capas y funciones de activación, existen infinidad de arquitecturas diseñadas y entrenadas para diversas tareas. A continuación, se detallarán las arquitecturas que se han empleado en el presente trabajo como punto de partida con el fin de llevar a cabo la localización visual con Redes Neuronales Siamesas.

3.4. Arquitecturas de CNNs

En el presente trabajo se han empleado diversas arquitecturas o modelos de CNNs para conformar la red neuronal siamesa. Entre otras, se han utilizado dos redes neuronales simples compuestas por tres capas convolucionales (Tabla **3-3**) y además, se ha propuesto la implementación de arquitecturas conocidas, como es el caso de AlexNet, DenseNet-121, VGG11, VGG13, VGG16 y VGG19 (Tabla **3-2**). Todas estas redes se caracterizan por tener una serie de capas convolucionales que son las encargadas de extraer las características de las imágenes de entrada. Las capas de activación ReLU no se han mostrado en las tablas por abreviarlas, pero se han empleado después de cada una de las capas convolucionales. Además, disponen de una serie de capas Fully Connected para orientar los datos hacia la clasificación final mediante una capa Softmax tras la capa fc-1000. El tamaño de la última capa Fully Connected representa el número de categorías que permite clasificar cada red.

AlexNet es una red neuronal convolucional de 8 capas de profundidad. Fue entrenada con más de un millón de imágenes de la base de datos ImageNet [9] con el objetivo de clasificar 1000 categorías de objetos. Por ejemplo, es capaz de clasificar objetos como teclados, ratones, lápices y animales. Como resultado de este entrenamiento, la red ha aprendido a obtener representaciones ricas en características para una amplia gama de imágenes. Por último, esta red tiene un tamaño de imagen de entrada 227 por 227 píxeles.

DenseNet se caracteriza por la introducción de convoluciones densas de manera que conecta cada capa convolucional con todas las anteriores. Esto permite una reducción sustancial del número de parámetros de la red. Además, refuerzan la propagación de características de una capa a otra. En cuanto al tamaño de la imagen de entrada, ha de tener un tamaño mínimo de 29 por 29 píxeles. De esta manera, ha sido entrenada con las bases de datos ImageNet [9] para la clasificación de objetos, The Street View House Numbers (SVHN) dataset [27] para la clasificación de dígitos y CIFAR [19] para llevar a cabo la clasificación de objetos en imágenes de baja resolución.

En el caso de las redes VGG, de forma general tienen la misma arquitectura pero disponen de un diferente número de capas convolucionales. Al igual que AlexNet, las redes VGG fueron entrenadas únicamente con la base de datos ImageNet [9] con el objetivo de llevar a cabo una clasificación entre 1000 objetos diferentes.

Tanto el diseño como el entrenamiento de CNNs son tareas relativamente sencillas, pero es posible ahorrar tiempo de entrenamiento y además, mejorar los resultados si empleamos una red neuronal previamente entrenada, conservando sus conocimientos y empleándolos como punto de partida de nuestros entrenamientos. Sin embargo, tendremos que modificar ligeramente la arquitectura reemplazando o eliminando alguna de sus capas con el fin de que el modelo se adecúe a nuestra tarea. Aquellas capas que se introduzcan llevarán asociados

unos pesos iniciales aleatorios, pero el conjunto de capas que se conserve mantendrá los pesos adquiridos en su entrenamiento previo. Esta técnica es conocida como *Transfer Learning*.

<table>
<tr><td>AlexNet</td><td>DenseNet</td><td>VGG11</td><td>VGG13</td><td>VGG16</td><td>VGG19</td></tr>
<tr><td>input size
227x227x3</td><td>min input size
29x29x3</td><td colspan="4">input size
224x224x3</td></tr>
<tr><td>conv2d-64</td><td>conv2d-112</td><td>conv2d-64</td><td>conv2d-64
conv2d-64</td><td>conv2d-64
conv2d-64</td><td>conv2d-64
conv2d-64</td></tr>
<tr><td>maxpool</td><td>maxpool</td><td colspan="4">maxpool</td></tr>
<tr><td>conv2d-192</td><td>conv2d-56 x 6</td><td>conv2d-128</td><td>conv2d-128
conv2d-128</td><td>conv2d-128
conv2d-128</td><td>conv2d-128
conv2d-128</td></tr>
<tr><td>maxpool</td><td>averagepool</td><td colspan="4">maxpool</td></tr>
<tr><td>conv2d-384
conv2d-256
conv2d-256</td><td>conv2d-28 x 12</td><td>conv2d-256
conv2d-256</td><td>conv2d-256
conv2d-256</td><td>conv2d-256
conv2d-256
conv2d-256</td><td>conv2d-256
conv2d-256
conv2d-256
conv2d-256</td></tr>
<tr><td rowspan="5">maxpool</td><td>averagepool</td><td colspan="4">maxpool</td></tr>
<tr><td>conv2d-14 x 24</td><td>conv2d-512
conv2d-512</td><td>conv2d-512
conv2d-512</td><td>conv2d-512
conv2d-512
conv2d-512</td><td>conv2d-512
conv2d-512
conv2d-512
conv2d-512</td></tr>
<tr><td>averagepool</td><td colspan="4">maxpool</td></tr>
<tr><td>conv2d-7 x 16</td><td>conv2d-512
conv2d-512</td><td>conv2d-512
conv2d-512</td><td>conv2d-512
conv2d-512
conv2d-512</td><td>conv2d-512
conv2d-512
conv2d-512
conv2d-512</td></tr>
<tr><td>averagepool</td><td colspan="4">maxpool</td></tr>
<tr><td>fc-4096</td><td rowspan="3">fc-1000</td><td colspan="4">fc-4096</td></tr>
<tr><td>fc-4096</td><td colspan="4">fc-4096</td></tr>
<tr><td>fc-1000</td><td colspan="4">fc-1000</td></tr>
<tr><td colspan="6">SoftMax</td></tr>
</table>

Tabla 3-2: Redes Neuronales Convolucionales empleadas para conformar la arquitectura de Red Neuronal Siamesa. Para cada arquitectura, se enumeran las diferentes capas de arriba a abajo, comenzando con la imagen de entrada.

Simple 1	Simple 2
Imagen de entrada (128 x 512 x 3)	
conv2d-3 conv2d-8 conv2d-16	conv2d-3 conv2d-16 conv2d-32
maxpool	
fc-500	
fc-500	
fc-5	

Tabla 3-3: Dos modelos adicionales para comparar con otro tipo de arquitecturas más complejas.

3.5. Descriptores de apariencia global

La descripción de imágenes de forma global es una técnica muy popular para abordar las tareas de mapeo y localización visual. Estas técnicas consisten en extraer las características más representativas de una imagen. Dichas características se pueden extraer mediante métodos analíticos (*hand-crafted*) o bien, mediante métodos basados en aprendizaje profundo a partir de una de las capas de la red. En trabajos anteriores como en [4], se ha propuesto entrenar una CNN con el objetivo de clasificar estancias y emplear esa misma red para obtener un descriptor a partir de una de sus capas intermedias.

En cambio, en el presente trabajo los entrenamientos están focalizados en la generación de descriptores globales que permitan generar una función de similitud entre un par de imágenes de entrada. La localización visual se realiza tanto de forma jerárquica como de forma global. En ambas técnicas se emplean Redes Neuronales Siamesas que por cada imagen de entrada generan un descriptor. Los descriptores correspondientes a las dos imágenes de entrada se comparan y se mide su similitud mediante la distancia entre ellos. En función de si la red ha sido entrenada para determinar si dos imágenes pertenecen a la misma estancia o a la misma pose, estos vectores descriptores serán más parecidos en dichas situaciones. Esto permite llevar la localización mediante una búsqueda del vecino más cercano.

3.6. Aumento de datos

Una red neuronal convolucional requiere de un gran volumen de datos de entrenamiento para la correcta generalización del modelo. Sin embargo, no siempre se dispone de un conjunto de datos de entrenamiento lo suficientemente extenso y entonces, el desempeño del modelo podría no ser el deseado. Para paliar dicha situación, en el presente trabajo se propone el empleo de la técnica de aumento de datos o Data Augmentation, la cual consiste

en crear nuevas instancias (imágenes) por medio de la aplicación de una serie de filtros y efectos sobre los datos originales. Además, los efectos propuestos en el presente trabajo están específicamente diseñados para la recreación de condiciones de operación reales ante diferentes condiciones de iluminación como pueden ser la aparición de reflejos y sombras en zonas locales de la imagen, o el cambio de iluminación global de la escena debido a las condiciones meteorológicas. Es por ello que los efectos aplicados se dividen en efectos locales y globales:

- **Efectos locales**: Se trata de reproducir los focos de luz o sombras que inciden en objetos, paredes y suelo. A esto le llamamos cambios de iluminación local, ya que sólo afecta a una parte de la imagen. La forma de las diferentes fuentes de luz puede variar significativamente. Son comunes las formas circulares de tipo farola o las formas cuadradas y trapezoidales como los reflejos producidos en el suelo cuando la luz exterior atraviesa las ventanas. Para simular estas fuentes de luz, modificamos la intensidad de diferentes regiones de la imagen de acuerdo las formas descritas anteriormente. De esta manera, se incrementan los valores de los píxeles de una región cuando se pretende reproducir un reflejo y se disminuye para simular el efecto de una sombra. Para que sea realista, debe haber un gradiente de intensidad entre la zona central de la región hasta el borde de la misma. El tamaño y forma de dichas regiones se selecciona de forma aleatoria, así como el valor de intensidad máximo o mínimo del reflejo o sombra respectivamente. En la Figura **3-7** se muestran algunos ejemplos de estos efectos.

(a) Efecto de reflejo en forma de foco.

(b) Efecto de sombra en forma de foco.

(c) Efecto de reflejo de forma cuadrada.

(d) Efecto de reflejo de forma trapezoidal.

Figura 3-7: Efectos locales para llevar a cabo un aumento de datos centrado en variaciones de iluminación.

- **Efectos globales**:En algunos casos pueden producirse variaciones globales de la iluminación. Para modelar estos cambios de iluminación, es necesario modificar los píxeles de toda la imagen, en lugar de hacerlo en una región.

  1. **Oscuridad y brillo**: Se añade un valor constante a todos los píxeles para modelar un efecto de brillo global en la imagen o se resta para simular una oscuridad global.

$$I_s = I \pm c$$

Donde I_s es la nueva imagen, I la imagen original y c es la constante que puede tomar un valor positivo o negativo. Las Figuras 3.8(b) y 3.8(c) muestran este efecto.

2. **Nitidez y desenfoque**: Encontrar bordes más nítidos entre los objetos contribuirá a proporcionar una mejor separabilidad entre ellos (Figura 3.8(d)). Por el contrario, los efectos de desenfoque dificultan la separabilidad entre los objetos provocando así imágenes más borrosas causadas por la baja iluminación o por movimientos de la cámara al capturar las imágenes (Figura 3.8(e)). Ambos efectos deben ser incorporados en la propuesta de aumento de datos puesto que se pueden dar en condiciones reales de uso. Ambos tipos de efecto se pueden conseguir convolucionando las siguientes máscaras sobre la imagen original:

Efecto nitidez **Efecto desenfoque**

$$m_{sh} = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix} \qquad m_{bl} = \frac{1}{25} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

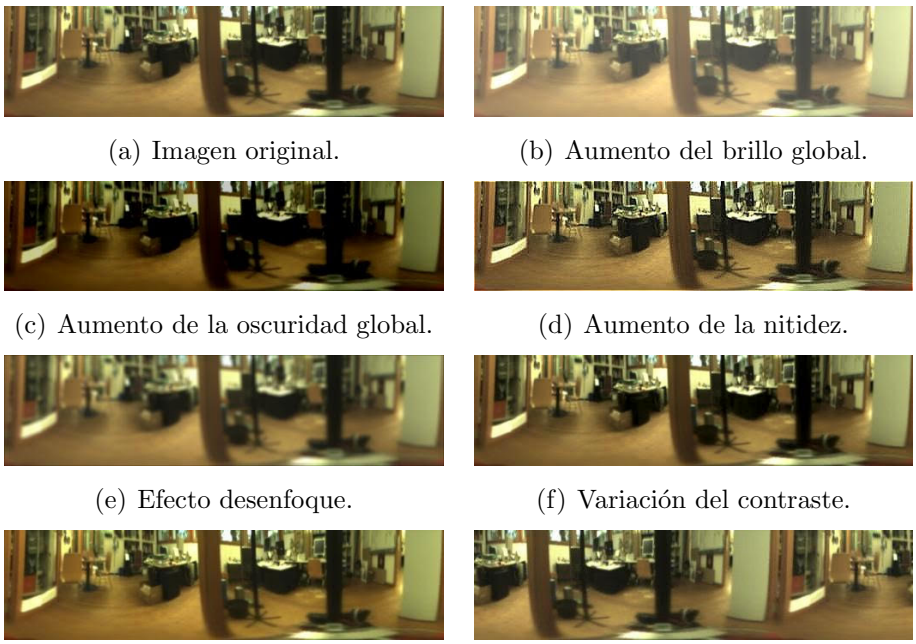
3. **Variación de contraste**: El contraste de la imagen desempeña un papel muy importante a la hora de resaltar los diferentes objetos de la escena. Por otro lado, las imágenes con bajo contraste suelen tener un aspecto más suave y con menos sombras y reflejos. El contraste se ha modificado siguiendo la siguiente ecuación:

$$I_s = 64 + c * (I - 64)$$

donde I_s es la nueva imagen, I es la imagen original y c es el factor de contraste. Para $c > 1$ el contraste aumenta y para $c < 1$ disminuye. Adicionalmente, se ha añadido una ecualización de la imagen lo que permite distribuir uniformemente los valores del histograma y de esta forma obtener el efecto de aumento del contraste. La Figura 3.8(f) muestra dicho efecto.

4. **Cambios en la saturación**: La saturación del color de la imagen está relacionada con la intensidad del color. Cuanto menor sea la saturación, menos colorida será la imagen, incluso puede parecer una imagen en escala de grises si la saturación es muy baja. Por el contrario, se obtienen colores más vivos cuanto mayor sea la saturación del color. Mediante este efecto se pueden simular situaciones en las que la iluminación cambia significativamente (Figura 3.8(g)). La saturación del color se puede llevar a cabo convirtiendo la imagen RGB a HSV. Posteriormente, es posible cambiar directamente la saturación operando con dicho canal.

5. **Rotaciones**: Consiste en introducir un desplazamiento circular aleatorio a las columnas de la imagen original con el fin de simular un cambio de orientación por parte del robot. Esta característica se debe a que la imagen omnidireccional cubre los 360 grados alrededor del plano del suelo y por tanto, si se gira la cámara, la información que aparece es la misma pero incluyendo un desplazamiento circular de sus columnas. La figura 3.8(h) muestra el efecto de rotación para un giro de 115 grados. Este efecto se ha aplicado de forma aleatoria entre 10 y 350 grados.



(a) Imagen original. (b) Aumento del brillo global.

(c) Aumento de la oscuridad global. (d) Aumento de la nitidez.

(e) Efecto desenfoque. (f) Variación del contraste.

(g) Cambios en la saturación. (h) Efecto de rotación.

Figura 3-8: Efectos globales de aumento de datos sobre una imagen ejemplo de la base de datos COLD.

4 Localización mediante Redes Neuronales Siamesas (SNNs)

La localización de robots móviles empleando Redes Neuronales Siamesas se ha propuesto tanto de forma jerárquica como de forma global. Para ello, se hará uso de la base de datos presentada en el apartado 3.2 y de las arquitecturas de Redes Neuronales Convolucionales descritas en el apartado 3.4. Por tanto, será necesario adaptar dichas redes para desempeñar la función deseada tal y como se detalla en el apartado 4.2. A continuación, se van a introducir los dos métodos de localización propuestos en la presente investigación.

4.1. Introducción a la tarea de localización

El proceso de localización del robot se ha abordado desde dos perspectivas:

- **Localización jerárquica.** Este tipo de localización consiste en determinar la posición en la que se encuentra el robot en dos pasos. El primer paso se denomina localización gruesa y tiene como propósito determinar la estancia en la que se encuentra el robot. Posteriormente, en el paso de localización fina se estiman las coordenadas en las que se encuentra el robot dentro de la habitación predicha. En el apartado 4.3 se va a realizar una descripción mucho más detallada del método propuesto con Redes Neuronales Siamesas.

- **Localización global.** La localización global tiene como objetivo determinar las coordenadas en las que se encuentra el robot a lo largo del entorno completo, sin dividirlo en estancias. De esta manera, se consigue el objetivo en un único paso, pero tiene un mayor coste computacional ya que requiere de un mayor número de operaciones. En el apartado 4.4 se va a realizar una descripción mucho más detallada del método propuesto con Redes Neuronales Siamesas.

Cabe destacar que se han empleado Redes Neuronales Siamesas para llevar a cabo ambos tipos de localización, variando el etiquetado y entrenamiento en función del objetivo.

4.2. Adaptación de las CNNs a las SNNs

Tal y como se ha descrito en el capítulo anterior, las Redes Neuronales Siamesas (SNNs) se caracterizan por estar compuestas por dos subredes idénticas, las cuales tienen la peculiaridad de compartir los pesos que las componen y de generar un vector descriptor por cada imagen de entrada. Ambas subredes convergen en la capa de comparación donde se establece la diferencia entre el par de descriptores mediante la distancia euclídea, de manera que a la salida se obtiene un valor que indica la disimilitud entre las imágenes de entrada (Figura **4-1**). Cada par de imágenes llevará asociada una etiqueta en la que se definirá el grado de diferencia entre ellas. En el caso más simple, esta etiqueta tomará el valor 0 cuando sean iguales y 1 cuando sean diferentes.

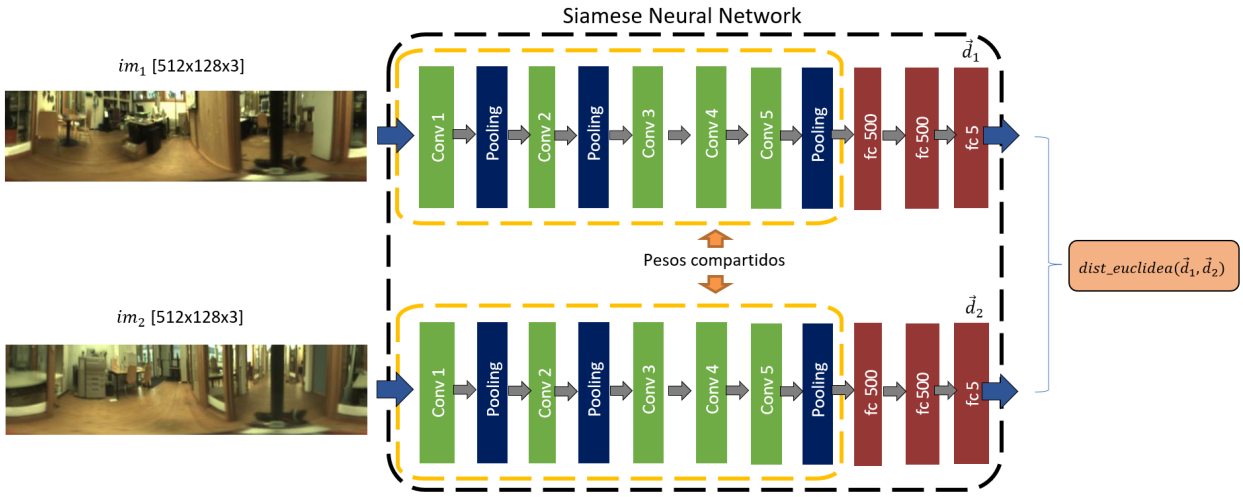


Figura 4-1: Ejemplo de arquitectura de una Red Neuronal Siamesa.

Por otro lado, cabe destacar que las subredes que componen las SNNs son CNNs convencionales ligeramente modificadas con el propósito de generar un vector descriptor por cada subred. Para ello, debemos cambiar la estructura final de dichas CNNs de tal manera que sustituimos las capas de clasificación por un conjunto de capas FullyConnected y ReLU. Estas capas nos permiten transformar la matriz resultante de las capas convolucionales en un vector, el cual será empleado para calcular la diferencia entre el par de imágenes de entrada. Estas imágenes tienen una dimensión de 512x128x3 por lo que también habrá que redimensionar la capa de entrada de cada subred. A modo de ejemplo, en la Tabla **4-1** se muestran las diferentes arquitecturas que se han empleado para la extracción de características, así como las capas finales para la generación del vector descriptor. Por simplificar, en dicha tabla no se han representado las capas de activación ReLU. En cuanto a las capas finales de escalado a vector, se han estudiado diferentes dimensiones de las capas FullyConnected y tamaños del vector de salida final tal y como se muestra en la sección experimental. Si

comparamos esta tabla con la presentada en el apartado 3.4 (Tabla **3-2**), se puede apreciar cómo se han modificado las capas finales para llevar a cabo el escalado a vector.

<table>
<tr><th>AlexNet</th><th>DenseNet</th><th>VGG11</th><th>VGG13</th><th>VGG16</th><th>VGG19</th></tr>
<tr><td colspan="6">Imagen de entrada (128x512x3)</td></tr>
<tr><td>conv2d-64</td><td>conv2d-112</td><td>conv2d-64</td><td>conv2d-64
conv2d-64</td><td>conv2d-64
conv2d-64</td><td>conv2d-64
conv2d-64</td></tr>
<tr><td>maxpool</td><td>maxpool</td><td colspan="4">maxpool</td></tr>
<tr><td>conv2d-192</td><td>conv2d-56 x 6</td><td>conv2d-128</td><td>conv2d-128
conv2d-128</td><td>conv2d-128
conv2d-128</td><td>conv2d-128
conv2d-128</td></tr>
<tr><td>maxpool</td><td>averagepool</td><td colspan="4">maxpool</td></tr>
<tr><td>conv2d-384
conv2d-256
conv2d-256</td><td>conv2d-28 x 12</td><td>conv2d-256
conv2d-256</td><td>conv2d-256
conv2d-256</td><td>conv2d-256
conv2d-256
conv2d-256</td><td>conv2d-256
conv2d-256
conv2d-256
conv2d-256</td></tr>
<tr><td rowspan="5">maxpool</td><td>averagepool</td><td colspan="4">maxpool</td></tr>
<tr><td>conv2d-14 x 24</td><td>conv2d-512
conv2d-512</td><td>conv2d-512
conv2d-512</td><td>conv2d-512
conv2d-512
conv2d-512</td><td>conv2d-512
conv2d-512
conv2d-512
conv2d-512</td></tr>
<tr><td>averagepool</td><td colspan="4">maxpool</td></tr>
<tr><td>conv2d-7 x 16</td><td>conv2d-512
conv2d-512</td><td>conv2d-512
conv2d-512</td><td>conv2d-512
conv2d-512
conv2d-512</td><td>conv2d-512
conv2d-512
conv2d-512
conv2d-512</td></tr>
<tr><td>averagepool</td><td colspan="4">maxpool</td></tr>
<tr><td colspan="6">fc-500</td></tr>
<tr><td colspan="6">fc-500</td></tr>
<tr><td colspan="6">fc-5</td></tr>
</table>

Tabla 4-1: Redes Neuronales Convolucionales adaptadas a la arquitectura de Red Neuronal Siamesa.

Una vez tenemos establecida la arquitectura de la Red Neuronal Siamesa, se entrena el modelo tomando como partida los pesos y umbrales de las CNNs preentrenadas, definidas en el apartado 3.4. Durante el proceso de entrenamiento, se utilizan pares de imágenes panorámicas como datos de entrada a la red y el grado de diferencia entre dicho par de imágenes como etiqueta. De esta manera se llevará a cabo una actualización de los pesos de las redes con cada par de imágenes de entrada (Figura **4-2**). Este método de entrenamiento se conoce como aprendizaje supervisado ya que las imágenes de entrada llevan asociadas una

etiqueta con la salida correcta de la red y en función del error existente entre la predicción de la SNN y la salida correcta, se actualizarán los pesos de cada uno de los nodos. En las secciones 4.3 y 4.4 se describirá con mayor nivel de detalle el proceso de entrenamiento y de etiquetado de los pares de imágenes de entrenamiento.

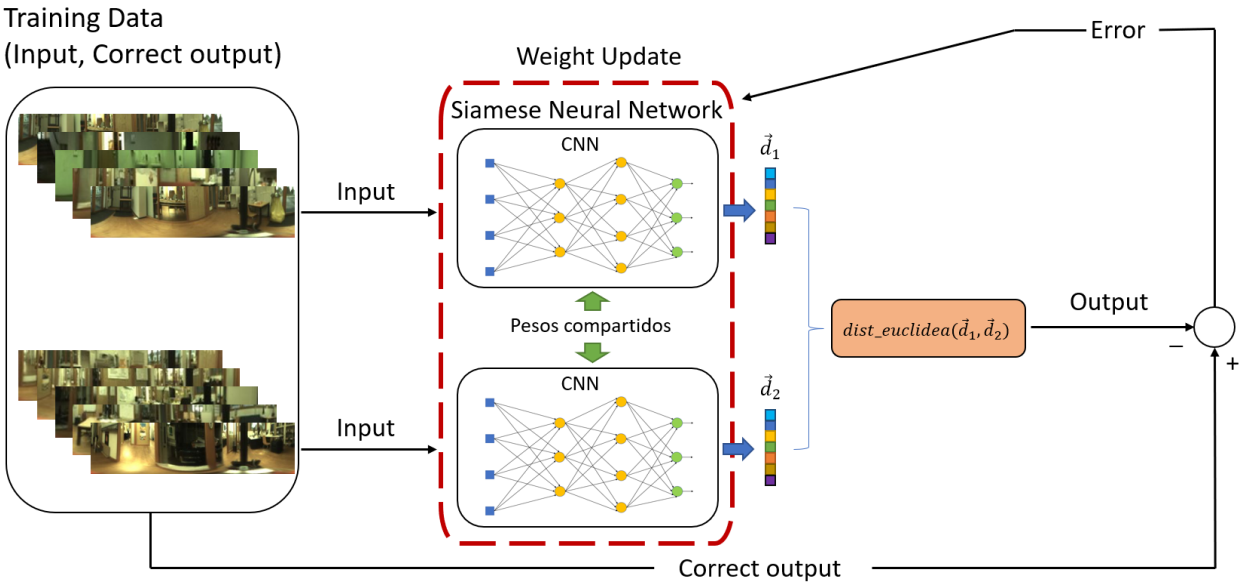


Figura 4-2: Aprendizaje supervisado de una Red Neuronal Siamesa.

4.3. Localización jerárquica mediante SNNs

4.3.1. Localización gruesa

El objetivo principal de esta tarea consiste en predecir la estancia en la que el robot capturó la imagen. Para conseguir este objetivo es necesario entrenar la Red Neuronal Siamesa con una serie de pares de imágenes capturadas por el robot de manera que las instantáneas pertenecientes a la misma habitación se etiquetan como 0 y las de diferente, se etiquetan como 1, tal y como se ejemplifica en la Tabla **4-2**. La ratio de pares de imágenes pertenecientes a la misma/diferente habitación puede variarse en la fase de entrenamiento.

Para llevar a cabo el testeo se precisará de una serie de imágenes representativas de cada estancia, las cuales se obtienen a partir del ground truth o modelo visual (Figura **4-4**). Por consiguiente, se hará una comparación entre la imagen test y las 9 imágenes representativas, una para cada una de las habitaciones, de tal modo que la imagen más similar proporcionará la habitación en la que la imagen test fue capturada. Se considerará que el par de imágenes se encuentra en la misma habitación si la predicción de diferencia está por debajo de 0.5. Si no, consideraremos que se encuentran en estancias diferentes.

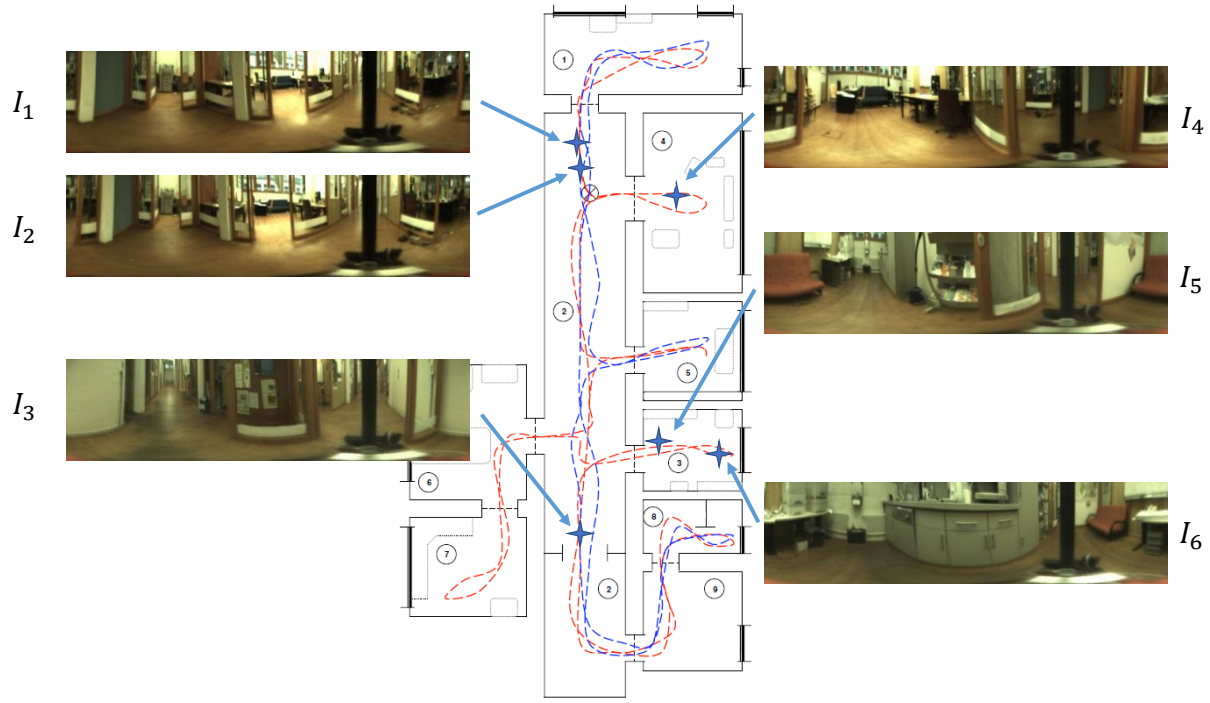


Figura 4-3: Ejemplo de imágenes para explicar el etiquetado en la fase de localización gruesa o identificación de estancias.

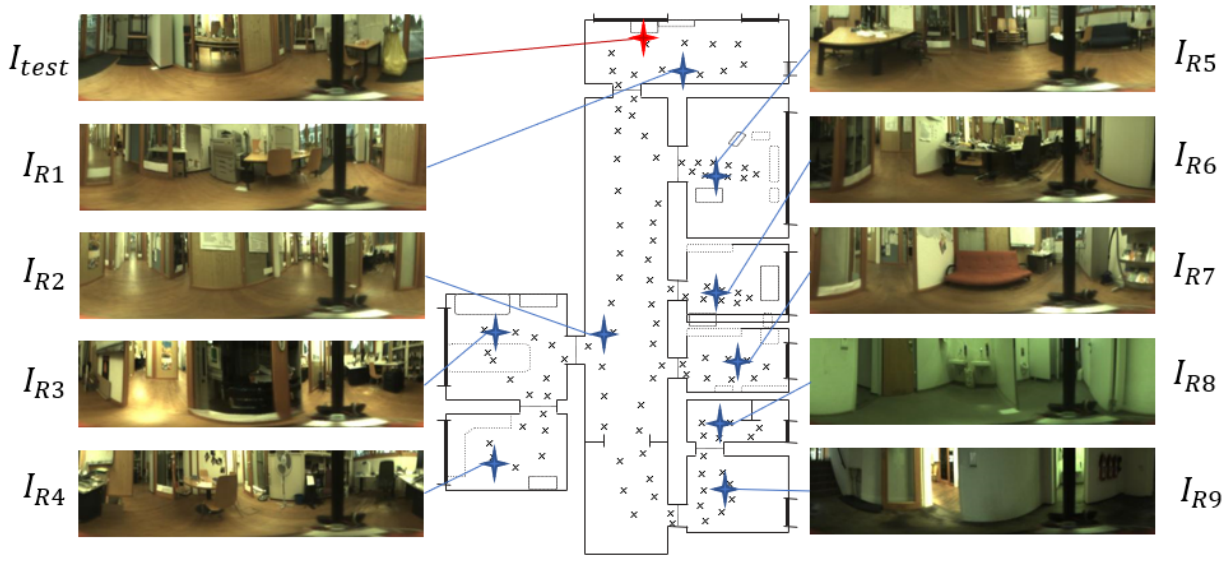


Figura 4-4: Imágenes representativas por cada habitación ($I_{R1}, I_{R2}, ..., I_{R9}$) e imagen test ($I_{test}$) a localizar.

Par de imágenes	Valor de la etiqueta
$I_1 - I_2$	0
$I_1 - I_3$	0
$I_1 - I_4$	1
$I_1 - I_5$	1
$I_4 - I_5$	1
$I_5 - I_6$	0

Tabla 4-2: Ejemplo descriptivo del etiquetado binario para abordar el entrenamiento de la Red Neuronal Siamesa con el fin de identificar la estancia en la que el robot capturó la imagen acorde con la Figura **4-3**.

4.3.2. Localización fina

La localización fina tiene como propósito estimar la pose del robot dentro de la estancia predicha en el paso anterior. Para ello, se entrenará una Red Neuronal Siamesa para cada una de las estancias que componen la base de datos. El etiquetado de los pares de imágenes continúa siendo entre 0 y 1, pero a diferencia del apartado anterior, ahora no se trabajará con valores binarios, sino que serán valores reales. La etiqueta de los pares de imágenes tomadas en la misma habitación se obtiene mediante el cociente de la distancia euclídea entre esas imágenes y la distancia euclídea máxima que se podría dar en esa habitación (Tabla **4-3**). De esta manera, imágenes capturadas en poses cercanas tendrán una etiqueta con un valor próximo a 0, mientras que las imágenes más alejadas entre sí tendrán una etiqueta próxima a 1.

$$Etiqueta(I_j, I_k) = \frac{dist\_euclidea(I_j, I_k)}{max\_dist\_estancia}$$

Una vez se haya entrenado una red siamesa para cada una de las estancias se puede llevar a cabo el testeo. Para ello, se precisará del modelo visual de cada una de las estancias, así como las redes neuronales siamesas entrenadas para cada habitación. Por tanto, en la fase de localización fina se hará una comparación entre la imagen test y las imágenes que conforman el modelo visual de la habitación previamente predicha. Esta comparación se realizará mediante la red siamesa entrenada para dicha estancia. Por ejemplo, si la estancia predicha anteriormente es la cocina, se comparará únicamente la imagen de test con las imágenes que conforman el modelo visual de dicha estancia (Figura **4-6**), de tal modo que la imagen más similar proporcionará la posición de la imagen test dentro de la estancia preseleccionada en el paso anterior.

En la Figura **4-5** se muestra un diagrama que explica el procedimiento global para llevar a cabo la localización jerárquica mediante Redes Neuronales Siamesas. La imagen test (I_{test}) se compara con las imágenes representativas por cada habitación ($I_{R1}, I_{R2}, ..., I_{R9}$) mediante

la SNN entrenada en el apartado 4.3 y se determina la habitación en la que se encuentra la imagen test. Posteriormente, dicha imagen test (I_{test}) se compara con el modelo visual de la habitación previamente predicha ($Imágenes_{RoomK}$) haciendo uso de la SNN entrenada en el presente apartado para llevar a cabo la localización fina en dicha habitación ($RoomK$). De esta forma se obtienen las coordenadas $(x_{R_{K,i}}, y_{R_{K,i}})$ en las que la imagen test fue capturada, es decir, se estima la localización del robot.

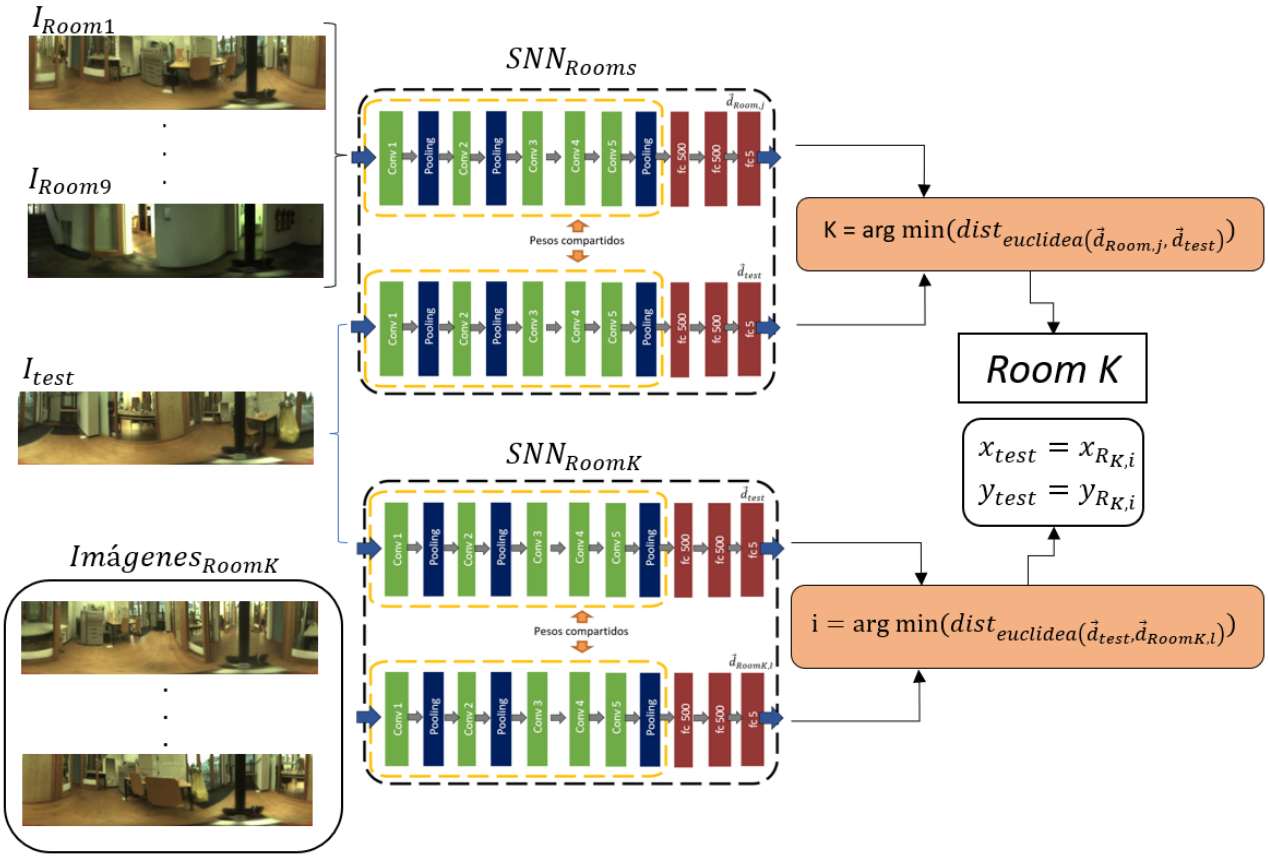


Figura 4-5: Diagrama representativo de la localización jerárquica mediante SNNs. La imagen test (I_{test}) se compara con las imágenes representativas por cada habitación $(I_{R1}, I_{R2}, ..., I_{R9})$ para determinar la habitación en la que se encuentra el robot. Posteriormente, dicha imagen test (I_{test}) se compara con el modelo visual de la habitación previamente predicha ($Imágenes_{RoomK}$) haciendo uso de la SNN entrenada para llevar a cabo la localización fina en dicha habitación ($RoomK$). De esta forma, se estima la posición del robot $(x_{R_{K,i}}, y_{R_{K,i}})$.

Nomenclatura de la estancia	Estancia	Máxima distancia entre imágenes
1P0-A	One-person office	3.0181 m
2P01-A	Two-person office 1	3.6886 m
2P02-A	Two-peroson office 2	2.3537 m
CR-A	Corridor	18.9940 m
KT-A	Kitchen	4.3956 m
LO-A	Large office	3.1061 m
PA-A	Print Area	3.5094 m
ST-A	Stairs Area	2.2909 m
TL-A	Bath room	2.9762 m

Tabla 4-3: Distancia máxima por habitación de la base de datos COLD Friburgo.

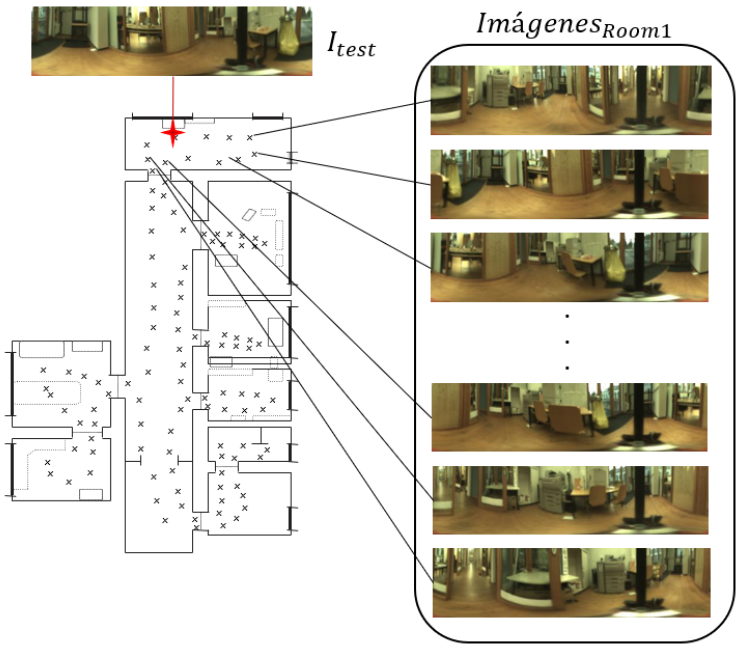


Figura 4-6: Imágenes representativas de la habitación 1 ($Imágenes_{Room1}$) y la imagen test (I_{test}) a localizar.

4.4. Localización global mediante SNNs

La localización global tiene como objeto estimar la pose del robot a lo largo de todo el mapa en un único paso. Para desempeñar esta tarea se entrenará una única Red Neuronal Siamesa para llevar a cabo la localización. El etiquetado de los pares de imágenes se establecerá de tal forma que imágenes capturadas en diferentes estancias tendrán como etiqueta un 1 e imágenes capturadas en la misma habitación tendrán asociada una etiqueta que será el resultado del cociente entre la distancia euclídea de ese par de imágenes y la distancia euclídea máxima que se puede dar entre dos imágenes que se encuentren en la misma habitación, es decir, 18.99 metros correspondientes a la distancia máxima que se puede dar en el pasillo. La formula que encontramos a continuación y la Tabla **4-4** detallan el etiquetado en cuanto a la localización global.

$Etiqueta(I_j, I_k) = \frac{dist\_euclidea(I_j,I_k)}{max\_dist\_edificio}$ si I_j y I_k pertenecen a la misma estancia.

$Etiqueta(I_j, I_k) = 1$ si I_j y I_k pertenecen a estancias diferentes.

Pair	**Euclidean distance**	**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	$\frac{2,48}{18,99} = 0{,}131$

Tabla 4-4: Ejemplo descriptivo del etiquetado para abordar el entrenamiento de la Red Neuronal Siamesa con el fin de identificar la pose en la que el robot capturó la imagen acorde con la Figura **4-3**.

Una vez entrenada la red, se realizará el testeo de tal forma que la imagen test se comparará con el modelo visual del edificio completo (Figura **4-7**). De esta manera, la imagen del modelo visual con menor diferencia nos permitirá establecer la pose del robot cuando capturó la imagen test (Figura **4-8**).

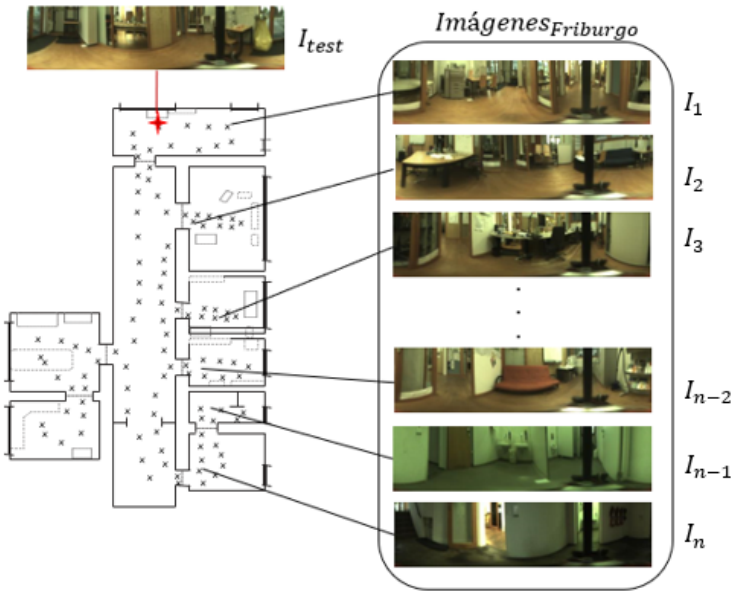


Figura 4-7: Ejemplo de imágenes representativas del modelo visual que conforman el edificio.

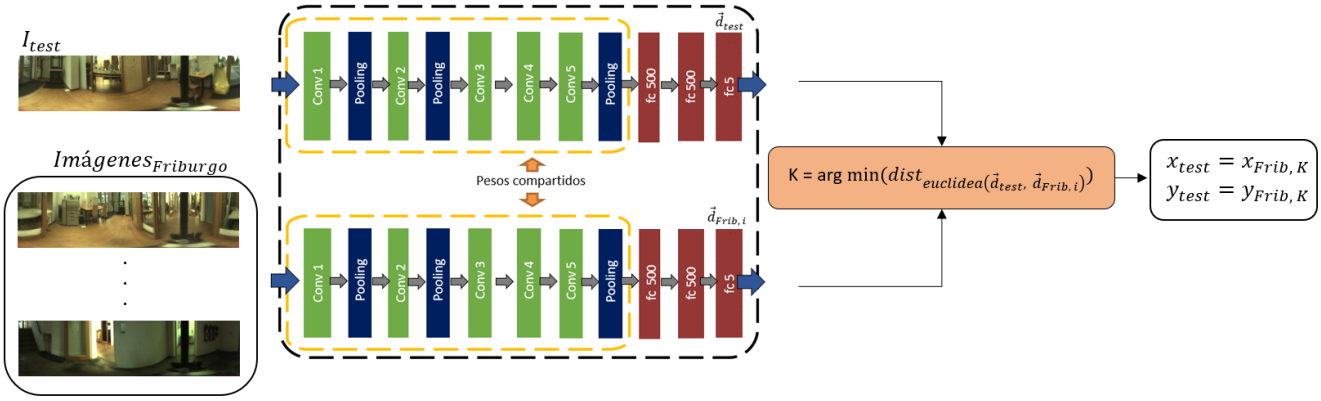


Figura 4-8: Diagrama representativo de la localización global mediante SNNs. La imagen test (I_{test}) se compara, una a una, con las imágenes representativas del modelo visual completo ($Imágenes_{Fribourgo}$) mediante la SNN entrenada en presente apartado y se determinan las coordenadas ($x_{Frib,K}, y_{Frib,K}$) en las que la imagen test fue capturada, es decir, se obtiene la localización del robot.

4.5. Aumento de datos

El aumento de datos tiene como propósito crear nuevas instancias de entrenamiento que permitan obtener redes neuronales más robustas. Para ello, en el presente trabajo se han aplicado un total de 13 efectos diferentes, tal y como se ha descrito en la sección 3.6.

- **Efectos locales:** Reflejo o sombra con forma circular, rectangular o trapezoidal.
- **Efectos globales:** Brillo, oscuridad, nitidez, desenfoque, contraste, saturación y rotación.

Además, algunos efectos se combinan con otros para obtener un aumento de datos más robusto. Pero no todos los efectos se combinan entre sí. La variación de iluminación global se combina con un único efecto local. Las combinaciones de efectos de iluminación pueden ser un efecto local de brillo con un efecto global de brillo, un efecto local de brillo con un efecto global de oscuridad o un efecto local de oscuridad con un efecto global de brillo. Por ejemplo, la oscuridad global se podría combinar con un reflejo en forma de círculo o foco. Además, también se combinar los efectos locales entre sí. Por último, el efecto de rotación se combina individualmente con todos los efectos y las combinaciones descritas anteriormente. De esta forma, se obtiene un total de 977856 imágenes tal y como se detallará en el apartado 5.1 de la siguiente sección.

5 Experimentos y resultados

La presente sección tiene como objetivo describir los experimentos realizados y presentar los resultados obtenidos para cada uno de ellos. A continuación, se mostrarán los conjuntos de entrenamiento y testeo empleados.

5.1. Conjunto de datos de entrenamiento y test

Estancia	**1P0-A**	**2P01-A**	**2P02-A**	**CR-A**	**KT-A**	**LO-A**	**PA-A**	**ST-A**	**TL-A**
Conjunto entreno 1	518	694	428	3258	674	395	804	495	619
Conjunto entreno 2	76736	82016	55616	416416	80608	46464	99968	53152	66880
Conjunto test 1	218	233	158	1183	229	132	284	151	190
Conjunto test 2	168	215	168	1114	270	121	241	198	212
Conjunto test 3	123	187	109	793	213	102	191	180	216
Conjunto test 4	509	635	435	3090	712	355	716	529	618
Modelo visual	44	46	31	238	46	26	57	30	38

Tabla 5-1: Tabla resumen del conjunto de datos de entrenamiento y testeo.

En la Tabla 5.1 se muestra el número de imágenes por estancia de cada uno de los conjuntos de datos que se van a emplear en el presente trabajo. Se dispone de dos conjuntos de entrenamiento, el conjunto de entrenamiento 1 consta de 8486 imágenes capturadas bajo las condiciones de iluminación de nublado, soleado y noche. El conjunto de entrenamiento 2, se ha obtenido aplicando un aumento de datos a la secuencia de nublado del conjunto de entrenamiento 1, de esta forma, se han generado 977856 imágenes. En cuanto a los conjuntos de test: el conjunto test 1 consta de 2595 imágenes bajo la condición de iluminación de nublado, el conjunto test 2 contiene imágenes capturadas bajo la condición de iluminación

de noche y consta de 2707 imágenes, el conjunto test 3 consta de 2114 imágenes bajo la condición de iluminación de soleado y el conjunto de test 4 esta compuesto por los conjuntos de test anteriores. Cabe destacar que las imágenes pertenecientes a los conjuntos de test son diferentes a las imágenes que componen el conjunto entrenamiento 1. Por último, el modelo visual se ha obtenido tras muestrear el recorrido bajo la condición de iluminación de nublado del conjunto de entrenamiento 1, obteniendo un total de 556 imágenes.

De este modo, los conjuntos de entrenamiento se emplearán para llevar a cabo el entrenamiento de las Redes Neuronales Siamesas, y los de test evaluarán el desempeño de las redes bajo las tres condiciones de iluminación. El modelo visual es el mapa de que dispone el robot para llevar a cabo la localización, por lo que será empleado en la fase de testeo tanto de la localización jerárquica como de la global.

5.2. Localización jerárquica

Tal y como se ha comentado en el apartado 4.3, la localización jerárquica consta de dos pasos, la localización gruesa que consiste en identificar la estancia en la que se encuentra el robot y la localización fina que consiste en obtener la posición exacta donde el robot capturó la imagen.

5.2.1. Estudio de los parámetros más característicos que influyen en entrenamientos de Redes Neuronales Siamesas.

En este apartado se ha realizado un profundo estudio de los parámetros que influyen en el entrenamiento de Redes Neuronales Siamesas para llevar a cabo la identificación de imágenes capturadas en la misma estancia e imágenes capturadas en estancias diferentes. Este estudio se ha realizado por ser de rápido computo y estar altamente ligado con localización gruesa o Room Retrieval. De esta manera, obtenemos una rápida y buena aproximación de qué valores de parámetros de entrenamiento son los mejores para abordar la localización gruesa.

Las SNNs deben ser entrenadas con pares de imágenes, en este caso entrenaremos con pares de imágenes de la misma estancia y de diferentes. De esta manera, uno de los elementos que estudiaremos en profundidad son las repercusiones que tiene en el desempeño de la red variar el porcentaje de imágenes iguales o diferentes en la fase de entrenamiento.

Por otro lado, también vamos a evaluar las diferentes arquitecturas propuestas en las Tablas **4-1** y **3-3**. Además, se estudiará el desempeño de la red en función de las dimensiones de las capas FullyConnected que conforman la fase de escalado a vector, así como el tamaño del vector descriptor que caracteriza a cada imagen de entrada. También se han estudiado

otros parámetros de entrenamiento como lo son el número de épocas y tamaño de los lotes de entrenamiento (Batch Size).

Estudio de las diferentes arquitecturas para la extracción de características:

En esta apartado se evalúa la primera parte de la Red Neuronal Siamesa la cual se encarga de extraer las características principales de las imágenes de entrada y cuyas capas parten de los pesos y umbrales obtenidos en el preentrenamiento. Se han evaluado dos redes neuronales simples creadas desde cero y sin preentrenamiento (Tabla **3-3**). Además, también se han empleado 10 redes neuronales comúnmente conocidas como es el caso de AlexNet, DenseNet, VGG11, VGG13, VGG16, VGG19, VGG11bn, VGG13bn, VGG16bn y VGG19bn. Las variantes bn no se han mostrado en la Tabla **4-1** por simplificar, pero disponen de la misma arquitectura que sus respectivas VGG, pero con una capa de BatchNorm2d tras cada convolucional conv2d.

Los experimentos que se muestran a continuación han sido desarrollados mediante un entrenamiento en el que el porcentaje de imágenes iguales (capturadas dentro de la misma estancia) es del 50 % y el de diferentes (capturadas en estancias distintas) es del 50 %. Además, el tamaño del lote es de 256 y la longitud del entrenamiento es de 5 épocas. En cuanto a las dimensiones de las capas de la fase de escalado a vector, toman el valor de 500-500-5 neuronas en cada capa FullyConnected, donde el vector descriptor (capa final) dispone de 5 neuronas. Por otro lado, el algoritmo de optimización del entrenamiento es el Stochastic Gradient Descent (SGD) con un Learning Rate de 0.001 y un Momentum de 0.9.

Los resultados de dichos experimentos se representan en la Tabla **5-2** de manera que se muestra la exactitud global del entrenamiento, la de testeo y además la exactitud para predecir tanto imágenes iguales como diferentes. Como podemos observar, los mejores resultados de forma global se dan para VGG13 y VGG16, obteniendo una exactitud para predecir imágenes pertenecientes a la misma habitación del 99.44 % y 99.47 % respectivamente. En cambio, la exactitud para predecir imágenes diferentes es menor, obteniendo una exactitud del 79.86 % y 78.91 % respectivamente. De forma general, las redes son capaces de predecir con mayor exactitud que dos imágenes pertenecen a la misma habitación, frente a dos imágenes pertenecientes a estancias diferentes. Esto se debe a que hay muchas más combinaciones posibles de imágenes diferentes que de iguales, por lo que resultará interesante variar el porcentaje de imágenes iguales/diferentes del entrenamiento. Además, cabe destacar el buen desempeño de las redes Simple 1 y Simple 2, pese a solo disponer de 3 capas convolucionales. El problema de estas redes es que sobreentrenan rápidamente tal y como se puede apreciar en la exactitud de entrenamiento, la cual es bastante superior a la exactitud global de test.

Network	Global Train Accuracy	Global Test Accuracy	Same Room Accuracy	Different Room Accuracy
Simple 1	97.52 %	84.59 %	98.16 %	**71.03 %**
Simple 2	98.20 %	86.45 %	98.87 %	74.06 %
Alexnet	87.62 %	86.10 %	98.78 %	73.41 %
Densenet	92.12 %	86.06 %	97.61 %	74.52 %
VGG11	92.55 %	87.43 %	99.08 %	75.78 %
VGG11bn	93.10 %	87.51 %	97.49 %	77.53 %
VGG13	94.66 %	**89.65 %**	99.44 %	**79.86 %**
VGG13bn	91.61 %	88.52 %	98.26 %	78.77 %
VGG16	93.30 %	89.19 %	**99.47 %**	78.91 %
VGG16bn	90.90 %	**82.04 %**	**92.68 %**	73.39 %
VGG19	94.42 %	89.17 %	99.30 %	79.04 %
VGG19bn	92.03 %	86.58 %	95.52 %	77.64 %

Tabla 5-2: Exactitud de diferentes arquitecturas de extracción de características para la identificación de estancias iguales y diferentes.

Estudio de la ratio entre imágenes iguales y diferentes en el entrenamiento:

En este apartado se ha realizado un profundo estudio de cómo afecta el porcentaje de imágenes de una categoría con respecto al desempeño de la red. Para ello se han realizado diferentes experimentos modificando la cantidad de pares de imágenes de estancias iguales y diferentes. Con el fin de reducir la cantidad de resultados, solamente se mostrarán los más representativos en el caso de las redes VGG13, VGG16 y AlexNet aunque se ha hecho un estudio mucho más extenso. Al igual que en el apartado anterior, el tamaño del lote es de 256 y la etapa de escalado a vector está compuesta por tres capas FullyConnected de tamaño 500-500-5.

Los resultados obtenidos se pueden observar en las Tablas **5-3**, **5-4** y **5-5**. Éstas muestran la relación entre el porcentaje de imágenes pertenecientes a estancias iguales-diferentes y la exactitud de clasificación. Si las observamos detenidamente, se puede apreciar que la exactitud para clasificar imágenes pertenecientes a estancias diferentes aumenta cuando tenemos una proporción mayor de imágenes de entrenamiento diferentes frente a iguales, independientemente de la cantidad de pares de imágenes diferentes. Es decir, conforme va aumentando la proporción de imágenes iguales en el entrenamiento, la exactitud de diferentes va decayendo, aunque la cantidad de imágenes de entrenamiento pertenecientes a habitaciones diferentes sea la misma o mayor. Esto es debido a que se ha empleado como función de error el *contrastive loss* que impide el correcto ajuste a ambos conjuntos de datos: el correcto ajuste en la predicción de imágenes iguales provoca un desajuste en la predicción de imágenes diferentes

y viceversa. Es por ello que resultaría interesante en futuras investigaciones el empleo del *circle loss* como función de error [39].

Epoch	Percentage Images (same-different)	Global Accuracy	Same Room Accuracy	Different Room Accuracy
7	5 %-95 %	89.88 %	92.03 %	87.73 %
9	5 %-95 %	91.89 %	92.51 %	91.27 %
11	5 %-95 %	92.20 %	92,71 %	91,70 %
7	10 %-90 %	92.72 %	98.13 %	87.30 %
9	10 %-90 %	94.76 %	98.69 %	90.82 %
11	10 %-90 %	95.08 %	98.90 %	91.25 %
7	25 %-75 %	93.10 %	99.09 %	87,12 %
9	25 %-75 %	93.46 %	99.06 %	87.86 %
11	25 %-75 %	93.53 %	99.21 %	87.85 %

Tabla 5-3: Exactitud de clasificación variando la ratio entre imágenes iguales y diferentes de entrenamiento para **VGG13**.

Epoch	Percentage Images (same-different)	Global Accuracy	Same Room Accuracy	Different Room Accuracy
7	5 %-95 %	94.35 %	96.47 %	92.23 %
9	5 %-95 %	94.94 %	96.48 %	93.39 %
11	5 %-95 %	94.24 %	97.77 %	90.72 %
7	10 %-90 %	93.04 %	97.16 %	88.92 %
9	10 %-90 %	94.26 %	97.18 %	91.35 %
11	10 %-90 %	93.59 %	97.96 %	89.22 %
7	25 %-75 %	92.46 %	99.21 %	85.71 %
9	25 %-75 %	92.28 %	99.30 %	85.25 %
11	25 %-75 %	91.78 %	98.81 %	84.74 %
7	40 %-60 %	92.95 %	99.38 %	86.52 %
9	40 %-60 %	92.72 %	99.48 %	85.95 %
11	40 %-60 %	93.28 %	99.50 %	87.05 %

Tabla 5-4: Exactitud de clasificación variando la ratio entre imágenes iguales y diferentes de entrenamiento para **VGG16**.

Epoch	Percentage Images (same-different)	Global Accuracy	Same Room Accuracy	Different Room Accuracy
7	5 %-95 %	92.36 %	90.11 %	94.60 %
11	5 %-95 %	93.58 %	94.08 %	93.07 %
14	5 %-95 %	93.68 %	94.14 %	93.22 %
7	10 %-90 %	92.05 %	94.65 %	89.44 %
11	10 %-90 %	93.41 %	96.84 %	89.98 %
14	10 %-90 %	93.01 %	97.19 %	88.82 %
7	25 %-75 %	90.91	97.54 %	84.28 %
11	25 %-75 %	91.16 %	98.92 %	83.39 %
14	25 %-75 %	90.59 %	98.28 %	82.19 %
7	40 %-60 %	88.33 %	98.80 %	77.85 %
11	40 %-60 %	88.65 %	99.07 %	78.23 %
14	40 %-60 %	88.54 %	99.25 %	77.82 %

Tabla 5-5: Exactitud de clasificación variando la ratio entre imágenes iguales y diferentes de entrenamiento para **AlexNet**.

Batch Size	Epoch	Percentage 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 %

Tabla 5-6: Exactitud de clasificación variando el tamaño del lote de entrenamiento para **VGG16**.

Batch Size	Epoch	Percentage 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 %

Tabla 5-7: Exactitud de clasificación variando el tamaño del lote de entrenamiento para **AlexNet**.

Desempeño de la Red en función del tamaño del lote:

Hasta este momento todos los experimentos se han realizado utilizando 256 como tamaño de lote, pero se han probado otros valores para tratar de optimizar los resultados. Las Tablas **5-6** y **5-7** muestran la exactitud de clasificación en función de los diferentes tamaños de lote. A partir de ellas se puede concluir que, en términos generales, la exactitud global aumenta cuando el tamaño del lote es menor.

Elección del tamaño de las capas que conforman el escalado a vector:

Tal y como se ha explicado en la Sección 4.2, las capas convolucionales se emplean para llevar a cabo la extracción de características y dan como resultado una matriz o mapa de características. Dicho mapa se convierte en un vector descriptor mediante una serie de capas FullyConnected. Es por ello que resulta de interés el estudio del tamaño de las capas que conforman la fase de escalado a vector, así como el tamaño del descriptor final. Los experimentos realizados en este apartado se han realizado con un porcentaje del 10 % de imágenes de entrenamiento pertenecientes a la misma habitación y un porcentaje del 90 % de imágenes pertenecientes a diferentes estancias.

Las diferentes configuraciones de escalado a vector que se han probado se resumido en el Tabla **5-8**. Las Tablas **5-9** y **5-10** muestran los resultados obtenidos en función de las tres configuraciones de escalado a vector presentadas. Las diferencias entre todos los resultados

en función del tamaño de las capas totalmente conectadas no son significativas. Los mejores resultados globales se obtienen con 3 capas totalmente conectadas con 1000-1000-10 neuronas cada una.

version 1	**version 2**	**version 3**
FC-500	FC - 500	FC - 1000
FC - 500	FC - 100	FC - 1000
FC - 5	FC - 10	FC - 10

Tabla 5-8: Configuraciones de las capas FullyConnected de la etapa de escalado a vector.

Scaled 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 %

Tabla 5-9: Exactitud de clasificación variando las capas de escalado a vector para **VGG16**.

Scaled 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 %

Tabla 5-10: Exactitud de clasificación variando las capas de escalado a vector para **AlexNet**.

Finalmente, el mejor resultado se obtiene utilizando VGG16 como red de extracción de características, 3 capas totalmente conectadas (1000-1000-10), 7 épocas y un tamaño de lote de 16. Con esta configuración se obtiene un 96.16 % de exactitud global, un 98.90 % de exactitud de imágenes pertenecientes a la misma estancia y un 93.41 % de exactitud de imágenes pertenecientes a diferentes estancias.

5.2.2. Localización gruesa

En el presente apartado se van a evaluar las Redes Neuronales Siamesas entrenadas en los apartados anteriores para desempeñar la identificación de la estancia en la que se encuentra la imagen capturada por el robot. Para ello, tal y como se ha explicado en la Sección 4.3.1, se ha de comparar la imagen test con cada una de las imágenes representativas de cada habitación de manera que aquella que sea más similar nos dará la habitación en la que se encuentra el robot. En la Tabla **5-11** se muestra una selección de resultados de Room Retrieval a partir de las redes entrenadas en los apartados anteriores con el conjunto de entrenamiento 1. Además, escogiendo la configuración que mejores resultados ha dado en dicha tabla, se obtienen los resultados equivalentes con la red entrenada con el conjunto de entrenamiento 2 que contiene el aumento de datos (Tabla **5-12**).

Network	Epochs	Percentage Images (same-different)	Batch Size	Cloudy accuracy	Night accuracy	Sunny accuracy
VGG16	10	25 %-75 %	256	99.64 %	96.49 %	93.38 %
VGG16	10	40 %-60 %	256	99.82 %	97.05 %	96.93 %
VGG16	13	40 %-60 %	256	99.82 %	97.23 %	96.93 %
VGG16	13	40 %-60 %	16	100 %	97.05 %	96.45 %
AlexNet	10	25 %-75 %	256	99.82 %	97.23 %	96.45 %
VGG19	10	35 %-65 %	256	99.82 %	97.05 %	96.22 %

Tabla 5-11: Exactitud del proceso de Room Retrieval (localización gruesa) mediante SNNs entrenadas con el conjunto de entrenamiento 1 que contiene imágenes capturadas bajo las 3 condiciones lumínicas.

Network	Epochs	Percentage Images (same-different)	Batch Size	Cloudy accuracy	Night accuracy	Sunny accuracy
VGG16	13	40 %-60 %	16	100 %	98.15 %	86.52 %

Tabla 5-12: Exactitud del proceso de Room Retrieval (localización gruesa) mediante una SNN entrenada con el conjunto de entrenamiento 2 que contiene el aumento de datos.

5.2.3. Localización fina

En este paso se lleva a cabo la localización dentro de las diferentes estancias una vez conocemos en qué habitación se encuentra el robot. Para ello, es necesario entrenar una Red Neuronal Siamesa por cada estancia obteniendo un total de 9 SNNs diferentes. La arquitectura que se ha seleccionado ha sido VGG16, con un tamaño de lote de 16, 30 épocas de entrenamiento y un tamaño de capas FullyConnected de 500-500-5. El entrenamiento se ha llevado a cabo tanto con el conjunto de entrenamiento 1 que contiene imágenes de las tres condiciones lumínicas como con el conjunto de entrenamiento 2 que contiene el aumento de datos. Para llevar a cabo el test, se han empleado los conjuntos de test 1, 2 y 3 que corresponden a las condiciones de iluminación de nublado, noche y soleado respectivamente. En las Tablas **5-13** y **5-14** se muestran los resultados de dichos entrenamientos.

5.3. Localización global

La localización global consiste en estimar la pose del robot en un único paso, comparando la imagen capturada por el robot con todas las imágenes que conforman el modelo visual, tal y como se ha explicado con mayor profundidad en la Sección 4.4. En la presente sección se han realizado diversos experimentos para elegir la mejor configuración de entrenamiento. Como en los experimentos anteriores VGG16 es la red que mejores resultados ha presentado, se ha seleccionado para abordar el problema de localización global.

Como parámetros de entrenamiento se ha seleccionado el optimizador Stochastic Gradient Descent (SGD) con un Learning Rate 0.001 y un Momentum de 0.9. Además, se ha escogido un tamaño de lote de 16 y 30 épocas de entrenamiento. A continuación, se va a evaluar el desempeño de las Redes Neuronales Siamesas para llevar a cabo la tarea de localización en función del tamaño del las capas FullyConnected de la fase de escalado a vector, el porcentaje de imágenes pertenecientes a la misma o a diferente habitación y la contribución del aumento de datos. Las redes siamesas cuyos resultados aparecen en las Tablas **5-15** y **5-16** se han entrenado con el conjunto de entrenamiento 1, el cual dispone de imágenes capturadas bajo las tres condiciones lumínicas. La Tabla **5-17** muestra los resultados de las redes cuyo entrenamiento se ha realizado con el conjunto de entrenamiento 2.

Elección del tamaño de las capas que conforman el escalado a vector:

La Tabla **5-15** muestra el error medio de localización en función de los diferentes tamaños de las capas que conforman el escalado a vector. Para ello, se ha empleado el conjunto de test 4, el cual contiene imágenes capturadas bajo las tres condiciones lumínicas. Como recordatorio, las capas que llevan a cabo el escalado se encargan de transformar los mapas de activación provenientes de las capas convolucionales en un vector unidimensional. Los

resultados muestran que los vectores finales de menor dimensión funcionan mejor para el problema de localización absoluta. La red siamesa es capaz de realizar la localización con un error medio de 0.5821 metros cuando se utilizan como capas de escalado a vector tres capas FullyConnected de tamaño 500, 500 y 5.

Room	**Error Cloudy**	**Error Night**	**Error Sunny**
1P0-A	0.0312 m	0.2434 m	0.2937 m
2P01-A	0.0653 m	0.2345 m	0.5137 m
2P02-A	0.0363 m	0.2897 m	0.2897 m
CR-A	0.0436 m	0.4140 m	0.7903 m
KT-A	0 m	0.2704 m	0.3027 m
LO-A	0.0839 m	0.3686 m	0.3710 m
PA-A	0 m	0.2821 m	0.2522 m
ST-A	0.0766 m	0.1981 m	0.3262 m
TL-A	0 m	0.1737 m	0.2145 m
Error medio	0.0368 m	0.3185 m	0.4941 m

Tabla 5-13: Exactitud para determinar la posición la que la imagen fue capturada mediante 9 SNNs entrenadas con el conjunto de entrenamiento que contiene imágenes capturadas bajo las 3 condiciones lumínicas.

Room	**Error Cloudy**	**Error Night**	**Error Sunny**
1P0-A	0.0305 m	0.2229 m	0.3674 m
2P01-A	0.0586 m	0.218 m	0.426 m
2P02-A	0.0363 m	0.2864 m	0.302 m
CR-A	0.022 m	0.2959 m	1.0683 m
KT-A	0 m	0.2534 m	0.4819 m
LO-A	0.0798 m	0.3332 m	0.557 m
PA-A	0 m	0.2241 m	0.4472 m
ST-A	0.0742 m	0.2073 m	0.3842 m
TL-A	0 m	0.1525 m	0.3785 m
Error medio	0.0267 m	0.2578 m	0.6609 m

Tabla 5-14: Exactitud para determinar la posición la que la imagen fue capturada mediante 9 SNNs entrenadas con el conjunto de entrenamiento que contiene el aumento de datos.

Network	Scaled Layers	Epoch	Percentage Images (same-different)	Global Error
VGG16	500-500-5	30	50 %-50 %	0.5821 m
VGG16	1000-1000-10	30	50 %-50 %	0.5904 m
VGG16	4096-4096-1000	30	50 %-50 %	0.8313 m

Tabla 5-15: Estudio del tamaño de las capas que realizan el escalado para llevar a cabo la localización global y haciendo uso de **VGG16**.

Estudio de la ratio entre imágenes iguales y diferentes en el entrenamiento:

La Tabla **5-16** muestra el error medio de localización en función del porcentaje de imágenes pertenecientes a la misma estancia y a diferente para un entrenamiento genérico de 30 épocas. Estos resultados se han obtenido a partir de los conjuntos de test 1, 2 y 3 que corresponden a las condiciones de iluminación de nublado, noche y soleado respectivamente. Para ello, se va a emplear la SNN que mejores resultados ha presentado hasta el momento, es decir, VGG16 para la extracción de características y las capas FullyConnected de tamaño 500-500-5 para llevar a cabo el escalado a vector. Los resultados muestran que el menor error se obtiene con un 40 % de imágenes iguales y un 60 % de diferentes. Estudiando los resultados, si hay un porcentaje realmente grande de imágenes pertenecientes a la misma estancia en el entrenamiento, los resultados de localización se verán perjudicados. En cambio, si hay una mayor cantidad de imágenes pertenecientes a estancias diferentes, los resultados mejoran considerablemente. De esta forma, encontramos que se obtienen los mejores resultados para un porcentaje de imágenes iguales del 40 % y un porcentaje de imágenes diferentes del 60 %.

Percentage Images (same-different)	Global Error	Cloudy Error	Night Error	Sunny Error
80 %-20 %	0.6284 m	0.1826 m	0.5600 m	0.8023 m
70 %-30 %	0.6035 m	0.1753 m	0.5375 m	0.7706 m
60 %-40 %	0.6006 m	0.1802 m	0.4991 m	0.8649 m
50 %-50 %	0.5821 m	0.1747 m	0.4837 m	0.8383 m
40 %-60 %	0.5097 m	0.1481 m	0.4547 m	0.6508 m
30 %-70 %	0.5193 m	0.1518 m	0.4908 m	0.6630 m
20 %-80 %	0.5202 m	0.1521 m	0.4917 m	0.6642 m

Tabla 5-16: Estudio de la ratio entre imágenes iguales-diferentes de entrenamiento para llevar a cabo la localización global mediante la red **VGG16** para la extracción de características.

Por último, en la Tabla **5-17** se van a mostrar los resultados con el empleo del Conjunto de datos de entrenamiento que contiene el aumento de datos y empleando como imágenes de test los conjuntos test 1, 2 y 3. Para ello, se va a partir de las mejores configuraciones obtenidas hasta ahora y se van a mostrar los resultados en función del número de imágenes de entrenamiento.

Number of images	**Percentage Images (same-different)**	**Cloudy Error**	**Night Error**	**Sunny Error**
1.333.440	40 %-60 %	0.0608 m	0.4682 m	1.2703 m
10.667.520	40 %-60 %	0.0418 m	0.3176 m	1.0050 m
3.666.960	40 %-60 %	0.0407 m	0.4028 m	1.3733 m
10.756.416	40 %-60 %	0.0375 m	0.2675 m	1.051 m
20.534.976	40 %-60 %	0.0325 m	0.2573 m	0.9913 m

Tabla 5-17: Estudio del efecto del aumento de datos para llevar a cabo la localización global mediante la red **VGG16** para la extracción de características.

6 Conclusiones y líneas de trabajo futuras

En el presente trabajo se ha propuesto un método de localización jerárquica y/o global mediante el empleo de Redes Neuronales Siamesas. La tarea de localización, junto con la de mapping, es una de las principales tareas que debe abordar un robot móvil autónomo. En los experimentos llevados a cabo se han testado diferentes arquitecturas de CNN para conformar la Red Neuronal Siamesa, algunas de las cuales son AlexNet, DenseNet, VGG11, VGG13, VGG16, VGG19, VGG11bn, VGG13bn, VGG16bn y VGG19bn. Las arquitecturas que mejor desempeño han proporcionado han sido VGG13 y VGG16. Las arquitecturas VGG han presentado de forma general buenos resultados, aunque sus variantes normalizadas (bn) han tenido un peor rendimiento.

Como se ha mencionado en las secciones anteriores, las Redes Neuronales Siamesas se han conformado a partir de otras arquitecturas, pero únicamente se ha aprovechado de ellas la capas convolucionales correspondientes a la fase de extracción de características y adicionalmente, se han añadido una serie de capas FullyConnected para llevar a cabo la conversión de los mapas de activación resultantes de las capas convolucionales a un vector descriptor. En el presente trabajo, se han estudiado diferentes tamaños de las capas que conforman la fase de escalado a vector, así como el tamaño del descriptor final. De esta manera, en la Sección 5.2.1 se han testado diferentes tamaños para desempeñar la tarea inicial de identificación de estancias iguales y diferentes, y se ha observado que el desempeño de la red para esta tarea es ligeramente superior cuando las capas FullyConnected tienen un tamaño de 1000-1000-10. En cambio, en los entrenamientos llevados a cabo en la localización global, el error de localización disminuye drásticamente en aquellas redes que tienen un tamaño de capas FullyConnected de 500-500-5.

En cuanto a los parámetros que afectan a la fase de entrenamiento y en concreto, el porcentaje de imágenes pertenecientes a estancias iguales y diferentes, se ha apreciado que si hay un porcentaje realmente grande de imágenes pertenecientes a la misma estancia en el entrenamiento, los resultados de localización se ven perjudicados. En cambio, si hay una mayor cantidad de imágenes pertenecientes a estancias diferentes, los resultados mejoran considerablemente.

Además, se ha propuesto el uso de la técnica de aumento de datos con el objetivo de realizar entrenamientos más robustos que permitan mejorar el desempeño de la red. Los efectos propuestos en este aumento de datos tratan de recrear condiciones reales de operación y además, se han generado una serie de efectos especialmente diseñados para aumentar la robustez frente a cambios lumínicos en la escena. En cuanto a los resultados obtenidos, el desempeño de la red se ve especialmente beneficiado cuando se trabaja en condiciones de iluminación de nublado y noche. En el caso de la condición de iluminación de nublado, cuando se realiza el entrenamiento con el aumento de datos, se reduce el error en torno a 12 centímetros en el caso de localización global y en torno a 1 centímetro en el paso fino de la localización jerárquica. En cuanto a la condición de iluminación de noche, se reduce el error en torno a 20 centímetros en el caso de la localización global y en torno a 6 centímetros en el caso del paso fino de la localización jerárquica. En cambio, en la condición de iluminación de soleado, cuando el entrenamiento de la red se realiza con el aumento de datos, el error de localización aumenta en 34 centímetros en el caso de la localización global y alrededor de 17 centímetros en el paso fino de la localización jerárquica.

Si comparamos los resultados obtenidos mediante los dos métodos de localización propuestos, observamos que la localización jerárquica presenta mejores resultados que la localización global. De esta forma, se obtiene un error de localización en condiciones de iluminación de nublado de 3.25 centímetros mediante la localización global frente a los 2.67 centímetros de la localización jerárquica. En la condición de iluminación de noche, se obtiene un error de 25.73 centímetros mediante la localización global y un error de 25.78 centímetros mediante la jerárquica. Finalmente, en cuanto al error de localización en la condición de iluminación de soleado, para la localización global se obtiene un error de 65.08 centímetros frente a los 49.41 centímetros de la localización jerárquica.

Como líneas de investigación futuras se propone extender las técnicas de localización propuestas a entornos de exterior, los cuales son más desafiantes por tratarse de entornos desestructurados y cambiantes. Además, se podrían considerar otro tipo de sensores para llevar a cabo la localización, como es el caso de los sensores LiDAR. En ese sentido, se podrían emplear arquitecturas que contengan convoluciones 3D con el fin de extraer las características más descriptivas de las nubes de puntos capturadas por el sensor.

Por último, continuando con la línea propuesta en el presente trabajo, se podría estudiar el uso de Redes Neuronales Tripletas, las cuales se caracterizan por estar compuestas por 3 subredes de arquitectura idéntica y con pesos compartidos. De esta forma, se ha demostrado en trabajos previos que las arquitecturas múltiples permiten solucionar tres de los retos más importantes en la recuperación de imágenes como lo son la brecha semántica, el aprendizaje de similitud y el espacio de almacenamiento (Fierro *et al.*, [12]).

Bibliografía

[1] K. Amer, R. Samy, M. ElHakim, M. Shaker, and M. ElHelw. Convolutional neural network-based deep urban signatures with application to drone localization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops*, Oct 2017.

[2] H. Andreasson, A. Treptow, and T. Duckett. Localization for mobile robots using panoramic vision, local features and particle filter. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 3348–3353, 2005.

[3] Josep Aulinas, Yvan Petillot, Joaquim Salvi, and Xavier Lladó. The slam problem: A survey. NLD, 2008. IOS Press.

[4] J. Cabrera, S. Cebollada, M. Flores, O. Reinoso, and Payá L. Training, optimization and validation of a cnn for room retrieval and description of omnidirectional images. *SN Computer Science*, 3, 2022.

[5] S. Cebollada, L. Payá, M. Flores, V. Román, A. Peidró, and O. Reinoso. A deep learning tool to solve localization in mobile autonomous robotics. *ICINCO 2020: 17th Intl. Conf. On Informatics in Control, Automation and Robotics*, 2020.

[6] S. Cebollada, L. Payá, V. Román, and O. Reinoso. Hierarchical localization in topological models under varying illumination using holistic visual descriptors. *IEEE Access*, 7, 2019.

[7] Sergio Cebollada, Luis Payá, María Flores, Adrián Peidró, and Oscar Reinoso. A state-of-the-art review on mobile robotics tasks using artificial intelligence and visual data. *Expert Systems with Applications*, 167:114195, 2021.

[8] Y. Chen, R. Chen, M. Liu, A. Xiao, D. Wu, and S. Zhao. Indoor visual positioning aided by cnn-based image retrieval: Training-free, 3d modeling-free. *Sensors*, 18(8), 2018.

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[10] J. Ding, B. Chen, H. Liu, and M. Huang. Convolutional neural network with data augmentation for sar target recognition. *IEEE Geoscience and remote sensing letters*, 13(3):364–368, 2016.

[11] H.J.S. Feder, J.J. Leonard, and C.M. Smith. Adaptive mobile robot navigation and mapping. *The International Journal of Robotics Research*, 18(7):650–668, 1999.

[12] A.N. Fierro, M. Nakano, K. Yanai, and H.M. Pérez. Redes convolucionales siamesas y tripletas para la recuperación de imágenes similares en contenido. *Información tecnológica*, 30:243 – 254, 12 2019.

[13] G. Gallegos and P. Rives. Indoor slam based on composite sensor mixing laser scans and omnidirectional images. *IEEE International Conference on Robotics and Automation*, 2010.

[14] Emilio Garcia-Fidalgo and Alberto Ortiz. Vision-based topological mapping and localization methods: A survey. *Robotics and Autonomous Systems*, 64:1–20, 2015.

[15] S. Gatesichapakorn, J. Takamatsu, and M. Ruchanurucks. Ros based autonomous mobile robot navigation using 2d lidar and rgb-d camera. In *2019 First International Symposium on Instrumentation, Control, Artificial Intelligence, and Robotics (ICA-SYMP)*, pages 151–154, 2019.

[16] D. Hahnel, R. Triebel, W. Burgard, and S. Thrun. Map building with mobile robots in dynamic environments. *IEEE International Conference on Robotics and Automation*, 2003.

[17] Charles C. Kemp, Aaron Edsinger, and Eduardo Torres-Jara. Challenges for robot manipulation in human environments [grand challenges of robotics]. *IEEE Robotics Automation Magazine*, 14(1):20–29, 2007.

[18] Pileun Kim, Jingdao Chen, and Yong K. Cho. Slam-driven robotic mapping and registration of 3d point clouds. *Automation in Construction*, 89:38–48, 2018.

[19] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[20] S. Kuutti, S. Fallah, K. Katsaros, M. Dianati, F. Mccullough, and A. Mouzakitis. A survey of the state-of-the-art localization techniques and their potentials for autonomous vehicle applications. *IEEE Internet of Things Journal*, 5(2):829–846, 2018.

[21] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(Nov):2278–2324, 1998.

[22] María Leyva-Vallina, Nicola Strisciuglio, and Nicolai Petkov. Generalized contrastive optimization of siamese networks for place recognition, 2021.

[23] David G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision-Volume 2 - Volume 2*, ICCV '99, page 1150, USA, 1999. IEEE Computer Society.

[24] E. Martín, J.L. Lázaro, F.J. Meca, D. Salido, F. Espinosa, and L. Pallarés. Infrared sensor system for mobile-robot positioning in intelligent spaces. *Sensorial Systems Applied to Intelligent Spaces*, 11(5):5416–5438, 2011.

[25] L. Moreno, J.M. Armingol, S. Garrido, A. de la Escalera, and M.A. Salichs. A genetic algorithm for mobile robot localization using ultrasonic sensors. *Journal of Intelligent and Robotic Systems*, 34:135–154, 2002.

[26] D. Murray and C. Jennings. Stereo vision based mapping and navigation for mobile robots. *Proceedings of International Conference on Robotics and Automation*, 1997.

[27] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011.

[28] K. Ohno, T. Tsubouchi, B. Shigematsu, and S. Yuta. Differential gps and odometry-based outdoor navigation of a mobile robot. *Advanced Robotics*, 18:611–635, 2004.

[29] I. Ohya, A. Kosaka, and A. Kak. Vision-based navigation by a mobile robot with obstacle avoidance using single-camera vision and ultrasonic sensing. *IEEE Transactions on Robotics and Automation*, 14(6)(Dec):969–978, 1998.

[30] L. Payá, F. Amorós, L. Fernández, and O. Reinoso. Performance of global-appearance descriptors in map building and localization using omnidirectional vision. *Sensors*, 14(2):3033–3064, 2014.

[31] L. Payá, A. Peidró, F. Amorós, D. Valiente, and O. Reinoso. Modeling environments hierarchically with omnidirectional imaging and global-appearance descriptors. *Remote sensing*, 10(4):522, 2018.

[32] L. Payá and O. Reinoso. A state-of-the-art review on mapping and localization of mobile robots using omnidirectional vision sensors. *Journal of sensors*, 2017.

[33] Henri Rebecq, Timo Horstschaefer, Guillermo Gallego, and Davide Scaramuzza. Evo: A geometric approach to event-based 6-dof parallel tracking and mapping in real time. *IEEE Robotics and Automation Letters*, 2(2):593–600, 2017.

[34] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408, 1958.

[35] Jose-Raul Ruiz-Sarmiento, Cipriano Galindo, and Javier Gonzalez-Jimenez. Building multiversal semantic maps for mobile robot operation. *Knowledge-Based Systems*, 119:257–272, 2017.

[36] J. Sandino, G. Pegg, F. Gonzalez, and G. Smith. Aerial mapping of forests affected by pathogens using uavs, hyperspectral sensors, and artificial intelligence. *Sensors*, 18(4), 2018.

[37] C. Sprunk, G. D. Tipaldi, A. Cherubini, and W. Burgard. Lidar-based teach-and-repeat of mobile robot trajectories. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3144–3149, 2013.

[38] M. Sualeh and G. Kim. Simultaneous localization and mapping in the epoch of semantics: A survey. *International Journal of Control, Automation and Systems*, 17, 2019.

[39] Yifan Sun, Changmao Cheng, Yuhan Zhang, Chi Zhang, Liang Zheng, Zhongdao Wang, and Yichen Wei. Circle loss: A unified perspective of pair similarity optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[40] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Ravinovich. Going deeper with convolutions. *In Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[41] L. Tai and M. Liu. Imagement classification with deep convolutional neural networks. *In Advances in neural information processing systems*, pages 1097–1105, 2012.

[42] L. Tai and M. Liu. Mobile robots exploration through cnn-based reinforcement learning. *Robotics and Biomimetics*, 2016.

[43] Z. Tao, P. Bonnifait, V. Frémont, and J. Ibañez-Guzman. Mapping and localization using gps, lane markings and proprioceptive sensors. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 406–412, 2013.

[44] J.P. Tardif, Y. Pavlidis, and K. Daniilidis. Monocular visual odometry in urban environments using an omnidirectional camera. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008.

[45] S. Thrun, W. Burgard, and D. Fox. A probabilistic approach to concurrent mapping and localization for mobile robots. *Autonomous Robots*, 5:253–271, 1998.

[46] P. J. Werbos. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 1990.

[47] R.W. Wolcott and R.M. Eustice. Robust lidar localization using multiresolution gaussian mixture maps for autonomous driving. *The International Journal of Robotics Research*, 36(3):292–319, 2017.

[48] D.F. Wolf and G.S. Sukhatme. Mobile robot simultaneous localization and mapping in dynamic environments. *Auton Robot*, 19:53–65, 2005.

[49] P. Wozniak, H. Afrisal, R. G. Esparza, and B. Kwolek. Scene recognition for indoor localization of mobile robots using deep cnn. *In Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[50] Haoran Wu, Zhiyong Xu, Jianlin Zhang, Wei Yan, and Xiao Ma. Face recognition based on convolution siamese networks. In *2017 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, pages 1–5, 2017.

[51] S. Xu, W. Chou, and H. Dong. A robust indoor localization system integrating visual localization aided by cnn-based image retrieval with monte carlo localization. *Sensors*, 19(2), 2019.

[52] L. Zhang and B. K. Ghosh. Line segment based map building and localization using 2d laser rangefinder. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, volume 3, pages 2538–2543, 2000.

[53] Zhipeng Zhang and Houwen Peng. Deeper and wider siamese networks for real-time visual tracking, 2019.