UNIVERSIDAD MIGUEL HERNÁNDEZ DE ELCHE

ESCUELA POLITÉCNICA SUPERIOR DE ELCHE

GRADO EN INGENIERÍA ELECTRÓNICA Y AUTOMÁTICA INDUSTRIAL

"Localización robusta de robots móviles mediante imágenes panorámicas, redes neuronales convolucionales y funciones de pérdida de triplete".

TRABAJO DE FIN DE GRADO

Julio - 2023

AUTOR: Marcos Alfaro Pérez

DIRECTOR/ES: Luis Payá Castelló

Biblioteca
UNIVERSITAS Miguel Hernández

CONTENIDO

LISTA DE FIGURAS **4**

LISTA DE TABLAS **9**

1 INTRODUCCIÓN **11**

2 ESTADO DEL ARTE **22**
2.1 CREACIÓN DE MAPAS Y LOCALIZACIÓN DE ROBOTS MÓVILES 22
2.2 DESCRIPCIÓN DE IMÁGENES 23
2.3 APRENDIZAJE PROFUNDO 24

3 HERRAMIENTAS UTILIZADAS **27**
3.1 VISIÓN OMNIDIRECCIONAL 27
3.2 BASE DE DATOS COLD 29
3.3 REDES NEURONALES TRIPLETAS 32
3.4 ARQUITECTURA INTERNA DE LA RED VGG16 33
3.5 DESCRIPTORES DE APARIENCIA GLOBAL 35
3.6 FUNCIONES DE PÉRDIDA 37

4 LOCALIZACIÓN MEDIANTE REDES NEURONALES TRIPLETAS **41**
4.1 INTRODUCCIÓN A LA TAREA DE LOCALIZACIÓN 41
4.2 ADAPTACIÓN DE LAS REDES SIMPLES A LAS REDES TRIPLETAS 41
4.3 LOCALIZACIÓN JERÁRQUICA MEDIANTE REDES TRIPLETAS 44
4.3.1 LOCALIZACIÓN GRUESA 44
4.3.2 LOCALIZACIÓN FINA 45
4.4 LOCALIZACIÓN GLOBAL MEDIANTE REDES TRIPLETAS . . . 47

5 EXPERIMENTOS Y RESULTADOS **50**
5.1 CONJUNTOS DE DATOS DE ENTRENAMIENTO, VALIDACIÓN Y TEST DE LA RED 50
5.2 LOCALIZACIÓN JERÁRQUICA 51
5.2.1 LOCALIZACIÓN GRUESA 51
5.2.2 LOCALIZACIÓN FINA 54

5.3 LOCALIZACIÓN GLOBAL 61
5.4 EXPERIMENTOS ADICIONALES 65

6 CONCLUSIONES Y LÍNEAS DE INVESTIGACIÓN FUTURAS 72

BIBLIOGRAFÍA 75

Biblioteca
UNIVERSITAS Miguel Hernández

LISTA DE FIGURAS

1-1 Robot quirúrgico Da Vinci diseñado por Intuitive Surgical https://www.intuitive.com/en-us. 11
1-2 Robot Curiosity utilizado por la NASA para la exploración del planeta Marte https://www.nasa.gov/. 11
1-3 Robot móvil autónomo Flexley Tug de ABB empleado para el transporte de piezas de vehículos en una industria https://new.abb.com/es. 12
1-4 Robot Husky A-200 ClearPath Robotics. 14
1-5 Cámara omnidireccional con un sistema catadióptrico. 14
1-6 Imágenes de la base de datos Digiface, creada por Bae *et al.* [3], utilizada para el reconocimiento de caras. 15
1-7 Sistema de detección de objetos de un Tesla Autopilot para la navegación autónoma [16]. 15
1-8 Relación entre *Deep Learning*, *Machine Learning* e Inteligencia Artificial. 16
1-9 Esquema de una neurona artificial. 16
1-10 Funciones de activación (a) Sigmoide, (b) ReLU y (c) Tangente hiperbólica. 17
1-11 (a) Red de una sola capa, (b) Red multicapa poco profunda y (c) Red multicapa profunda. 18
1-12 Estructura de las Redes (a) Simples, (b) Siamesas y (c) Tripletas. . . 20
1-13 Proceso de entrenamiento de una red neuronal mediante funciones de pérdida de triplete. 21

3-1 Tipos de espejo para el sistema catadióptrico: (a) Espejo hiperbólico, (b) Espejo parabólico. 28
3-2 (a) Imagen omnidireccional, (b) Imagen omnidireccional convertida a panorámica. 28
3-3 Ejemplo de imágenes contenidas en la base de datos COLD-Friburgo para las tres condiciones de iluminación: (a) Nublado, (b) Noche y (c) Soleado. Las imágenes contenidas en esta base de datos pueden descargarse a través de su página web https://www.cas.kth.se/COLD/. 29
3-4 Ejemplo de robot móvil autónomo ActivMedia Pioneer-3. 30
3-5 Estancias del edificio Friburgo de la base de datos COLD. 30
3-6 Planta del edificio Friburgo de la base de datos COLD. 31

3-7 Estructura de las redes tripletas. 33
3-8 Ejemplo de operaciones de Pooling: (a) Max Pooling, (b) Average Pooling y (c) Sum Pooling. 35
3-9 Arquitectura interna del modelo de red VGG16. 36
3-10 Ejemplo de una predicción correcta y otra errónea de la red. 40

4-1 Entrenamiento de las Redes Neuronales Tripletas. 42
4-2 Arquitectura interna de la red VGG16 adaptada a la tarea de localización. 43
4-3 Combinación de tres imágenes ancla (I_a), positiva (I_+) y negativa (I_-) para el entrenamiento de la red en la localización gruesa. 44
4-4 Imágenes representativas de cada habitación para realizar el test en la localización gruesa. 45
4-5 Comparación de una imagen I_{test} con las imágenes del modelo visual de la habitación predicha por la red en el test de la localización fina. . 46
4-6 Proceso de test en la localización jerárquica. 47
4-7 Comparación de una imagen I_{test} con las imágenes del modelo visual de todo el edificio en el test de la localización global. 48
4-8 Proceso de test en la localización global. 49

5-1 Matrices de confusión obtenidas en el test de la red entrenada con la función de pérdida *Lazy Triplet Loss* para a) Nublado, b) Noche y c) Soleado 53
5-2 Recall para K = 1, 2, 4 y 8 en la localización fina para cada función de pérdida en a) Nublado, b) Noche y c) Soleado. 56
5-3 Recall para K = 1, 2, 4 y 8 en la localización fina con la función de pérdida *Lazy Triplet Loss* para cada valor de r en a) Nublado, b) Noche y c) Soleado. 58
5-4 Predicciones de la red para la localización fina con la función de pérdida *Lazy Triplet Loss* con r=0'3 m en la habitación 1PO-A para a) Nublado, b) Noche y c) Soleado. 60
5-5 Recall para K = 1, 2, 4 y 8 en la localización global para cada función de pérdida en a) Nublado, b) Noche y c) Soleado. 63
5-6 Predicciones de la red para la localización global con la función de pérdida *Triplet Margin Loss* para a) Nublado, b) Noche y c) Soleado. 64
5-7 Recall para K = 1, 2, 4 y 8 en la localización jerárquica para cada función de pérdida en a) Nublado, b) Noche y c) Soleado. 66
5-8 Recall para K = 1, 2, 4 y 8 en la localización jerárquica con la función de pérdida *Lazy Triplet Loss* para cada valor de r en a) Nublado, b) Noche y c) Soleado. 70

5-9 Predicciones de la red para la localización jerárquica con la función de pérdida *Lazy Triplet Loss* con r=0'3 m para a) Nublado, b) Noche y c) Soleado. 71

Biblioteca
UNIVERSITAS Miguel Hernández

Biblioteca
UNIVERSITAS Miguel Hernández

LISTA DE TABLAS

3-1 Número de imágenes por habitación del conjunto de datos de entrenamiento. 32

5-1 Tabla resumen de los conjuntos de datos de entrenamiento, validación y test. 50
5-2 Resultados de la localización gruesa para cada función de pérdida en las 3 condiciones de iluminación. 52
5-3 Error medio cometido en metros en la localización fina para cada función de pérdida en las 3 condiciones de iluminación. 55
5-4 Error medio cometido en metros en la localización fina para cada valor de r en las 3 condiciones de iluminación. 57
5-5 Error medio cometido en metros en la localización fina para la función de pérdida *Lazy Triplet Loss* con r=0'3 m en las 3 condiciones de iluminación y errores mínimos que se pueden cometer para cada habitación. 59
5-6 Error medio cometido en metros en la localización global para cada función de pérdida en las 3 condiciones de iluminación. 62
5-7 Error medio cometido en metros en la localización jerárquica para cada función de pérdida en las 3 condiciones de iluminación. 67
5-8 Error medio cometido en metros en la localización jerárquica para cada valor de r en las 3 condiciones de iluminación. 68
5-9 Error medio cometido en metros en la localización jerárquica para la función de pérdida *Lazy Triplet Loss* con r=0'3 m en las 3 condiciones de iluminación y errores mínimos que se pueden cometer para cada habitación. 68

Biblioteca
UNIVERSITAS Miguel Hernández

1 INTRODUCCIÓN

Desde que se inventó el primer robot hasta el día de hoy, nuestras vidas han cambiado considerablemente. Actualmente, existen robots que realizan todo tipo de tareas que suponían un gran esfuerzo para las personas, como por ejemplo las intervenciones quirúrgicas mínimamente invasivas (Figura **1-1**), la exploración de zonas inaccesibles para el ser humano (Figura **1-2**) o el transporte de carga en una industria (Figura **1-3**).

Figura 1-1: Robot quirúrgico Da Vinci diseñado por Intuitive Surgical https://www.intuitive.com/en-us.

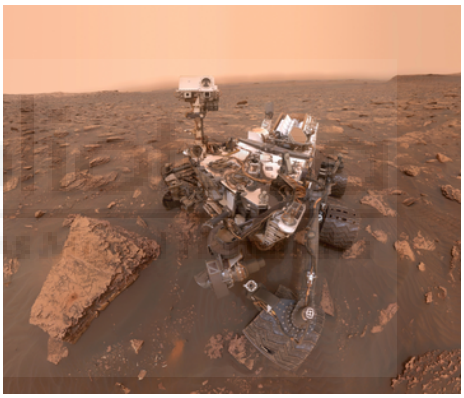


Figura 1-2: Robot Curiosity utilizado por la NASA para la exploración del planeta Marte https://www.nasa.gov/.

La robótica es una disciplina que abarca una gran variedad de campos de la ingeniería, como por ejemplo la eléctrica, la mecánica, la electrónica, la informática o las telecomunicaciones. En las últimas décadas se han producido avances en cada una de estas ramas que han supuesto un aumento considerable del uso de robots en múltiples aplicaciones.

Como consecuencia de esta revolución ha surgido la robótica móvil, una disciplina que se centra en el desarrollo de robots capaces de moverse por el entorno que les rodea, obtener información y realizar acciones sobre dicho entorno. Para que un

Figura 1-3: Robot móvil autónomo Flexley Tug de ABB empleado para el transporte de piezas de vehículos en una industria https://new.abb.com/es.

robot navegue de forma segura debe ser capaz de generar un mapa del entorno, saber cuál es su posición dentro de ese mapa y determinar cuál es el camino óptimo para desplazarse entre dos puntos del entorno. Por tanto, podemos agrupar las principales tareas que llevan a cabo los robots móviles en tres grupos: la creación de mapas, la localización y la planificación de trayectorias. Cuando se desea obtener la posición de un robot pero no se dispone de un mapa del entorno, se suelen emplear técnicas de SLAM (*Simultaneous Localization And Mapping*), que consisten en generar un mapa del entorno al mismo tiempo que se obtiene la posición del robot en el mismo.

Para poder realizar estas tareas, el robot debe llevar sensores incorporados que le permitan obtener la información necesaria del entorno. Los sensores más utilizados en robótica móvil son los siguientes:

- **GPS**: Sistema que permite obtener la posición de un dispositivo receptor sobre la Tierra. El dispositivo recibe múltiples señales de una constelación de satélites en órbita y calcula su ubicación exacta. Este sistema se utiliza principalmente en navegación y geolocalización. Su principal desventaja es su baja precisión, con errores de unos pocos metros, por lo que se suele combinar con otro tipo de sensores o bien se utilizan GPS diferenciales que tienen un error de posición mucho menor, alcanzando errores de unos pocos centímetros.
- **Sensores de rango**: Se utilizan para medir la distancia a la que se encuentra un objeto o una superficie. Estos sensores emiten una señal o una radiación y calculan la distancia en función del tiempo que tarda en reflejarse en el objeto y regresar.

  - **Láser**: Utilizan un haz de luz láser para medir la distancia al objeto. Este tipo de sensores permite calcular distancias con una gran precisión. Generalmente, se emplean en aplicaciones que requieren una gran exactitud, como por ejemplo la conducción autónoma de vehículos o el mapeo tridimensional.
  - **Infrarrojos**: Emiten rayos de luz infrarroja y miden la intensidad de la luz reflejada, y la distancia al objeto se calcula en función de esta magnitud. Este tipo de sensores se suele utilizar en aplicaciones de seguridad y de detección de objetos a corta distancia.
  - **Ultrasonidos**: Emiten pulsos de sonido a alta frecuencia y calculan el tiempo que tarda en reflejarse en el objeto y regresar. A partir de la velocidad del sonido, pueden medir la distancia al objeto. Este tipo de sensores se utiliza en tareas de detección de objetos y en los sistemas de estacionamiento de vehículos.

- **Sensores ópticos**: Este tipo de sensores utilizan una o varias cámaras para extraer información de la escena. Su principal ventaja es que permiten extraer una gran cantidad de información a un coste muy bajo. Debido a su gran versatilidad, se emplean en todo tipo de aplicaciones.
  - **Cámara estándar**: Sistema formado por una única cámara. No permite medir la distancia a los objetos de la escena, pero se puede extraer información relevante de la escena, como por ejemplo colores, texturas y formas que permitirán abordar tareas de alto nivel como la detección de personas, la segmentación de objetos u otro tipo de tareas. Como solo son capaces de extraer información de una parte de la escena, es frecuente combinarlos con otras cámaras u otro tipo de sensores.
  - **Par estereoscópico**: Sistema formado por un par de cámaras que están separadas una cierta distancia "baseline". Esto provoca una disparidad entre las imágenes capturadas por cada cámara, lo que permite calcular la profundidad a la que se encuentran los elementos de la escena mediante triangulación y búsqueda de correspondencias.
  - **Cámaras omnidireccionales**: Estas cámaras proporcionan información de la escena en todos los ángulos de visión, por lo que con una única imagen se puede representar completamente el entorno que rodea al sensor. Además, la imagen capturada es la misma independientemente de la orientación de la cámara, salvo por una cierta rotación. Las cámaras omnidireccionales pueden tener distintas configuraciones. Las más destacadas son los sistemas multicámara y los sistemas catadióptricos, formados por

una única cámara y un espejo cóncavo que suele ser hiperbólico o parabólico.

Los sensores ópticos, debido a su versatilidad y a la abundante información que proporcionan con un número reducido de imágenes, son una de las opciones más utilizadas en el área de la robótica móvil. En este trabajo, se han empleado imágenes capturadas mediante una cámara omnidireccional con un sistema catadióptrico que contiene un espejo hiperbólico para realizar la tarea de localización de un robot móvil (Figuras **1-4** y **1-5**). Las imágenes han sido convertidas a imágenes panorámicas para abordar este problema.

Figura 1-4: Robot Husky A-200 ClearPath Robotics.

Figura 1-5: Cámara omnidireccional con un sistema catadióptrico.

Para realizar la descripción de imágenes, existen dos grandes líneas de investigación: mediante descriptores globales y descriptores basados en puntos característicos. La descripción global de imágenes consiste en extraer características a partir de la información general de la imagen, mientras que la descripción basada en puntos característicos únicamente se centra en aquellos puntos fácilmente reconocibles en una imagen, como los bordes o las esquinas. Para abordar la tarea de localización, en este trabajo se han empleado descriptores globales.

En la actualidad, el desarrollo de herramientas de Inteligencia Artificial han supuesto una nueva revolución en la sociedad. La Inteligencia Artificial es una rama de conocimiento que trata de crear sistemas y programas para resolver problemas que normalmente requieren de la inteligencia humana. Su principal ventaja es la gran flexibilidad que presentan, que permite adaptar este tipo de herramientas a una gran variedad de aplicaciones, como por ejemplo los chatbots como ChatGPT,

el diagnóstico temprano de enfermedades, el reconocimiento facial (Figura **1-6**) o la conducción autónoma (Figura **1-7**).

Figura 1-6: Imágenes de la base de datos Digiface, creada por Bae *et al.* [3], utilizada para el reconocimiento de caras.

Figura 1-7: Sistema de detección de objetos de un Tesla Autopilot para la navegación autónoma [16].

La Inteligencia Artificial también ha permitido desarrollar nuevos métodos para la creación de mapas y la localización de robots móviles, como por ejemplo el uso de Redes Neuronales Convolucionales (CNNs), enmarcadas en el aprendizaje profundo o *Deep Learning*, una rama de la Inteligencia Artificial. A diferencia del *Machine Learning*, en el que la extracción de características de los datos de entrada se realiza de forma manual, se emplean redes neuronales con múltiples capas que son capaces de extraer sus propias características (Figura **1-8**). Las redes neuronales tratan de imitar el proceso de aprendizaje humano. Cuando las personas reciben información nueva, se modifican las conexiones internas entre las neuronas de su cerebro. De la misma forma, cuando las redes neuronales reciben nueva información, modifican los pesos y umbrales que enlazan las neuronas.

En la Figura **1-9** se muestra el esquema de una neurona artificial y los elementos que la componen, donde $x_1, x_2, ..., x_n$ son las entradas de la neurona, que al mismo tiempo son las salidas de las capas anteriores, $w_1, w_2, ..., w_n$ son los pesos asociados a cada una de las entradas y w_0 es el sesgo, un valor seleccionado de forma aleatoria al inicializar el modelo y que pertenece constante durante el entrenamiento de la red. Al introducir un nuevo dato, las entradas de cada neurona toman un cierto valor y cada una de ellas es multiplicada por sus respectivos pesos. Se realiza la suma de todas ellas y a partir del valor obtenido se calcula la salida mediante la función de

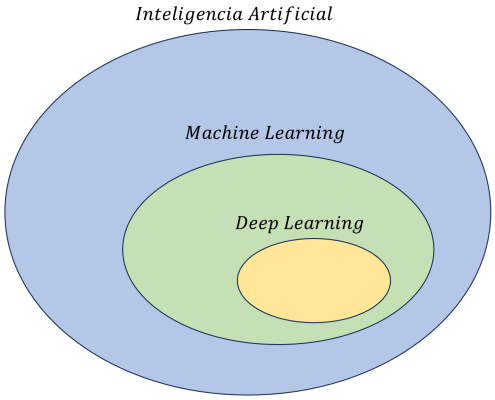


Figura 1-8: Relación entre *Deep Learning*, *Machine Learning* e Inteligencia Artificial.

activación. Según la salida obtenida en la última capa y la diferencia de ésta con la salida esperada, se reajustarán los pesos de la red en mayor o menor medida. Por tanto, la información relacionada con el aprendizaje de la red neuronal está contenida en los pesos de la red.

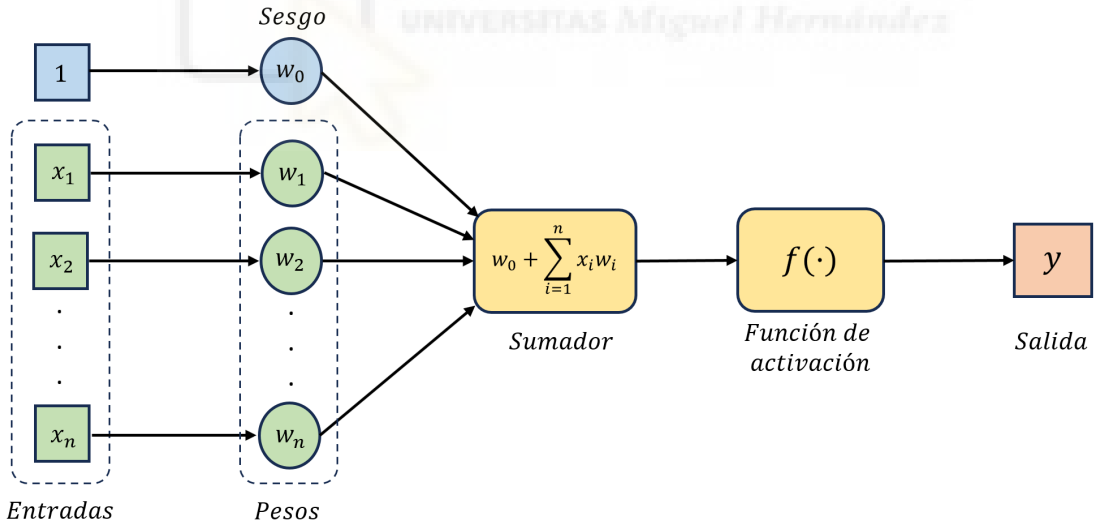


Figura 1-9: Esquema de una neurona artificial.

Puesto que las entradas y la salida dependen del dato de entrada, el sesgo es un valor constante y los pesos se están reajustando continuamente, el único parámetro seleccionable por el programador es la función de activación. Generalmente, las

funciones de activación más utilizadas son las siguientes:

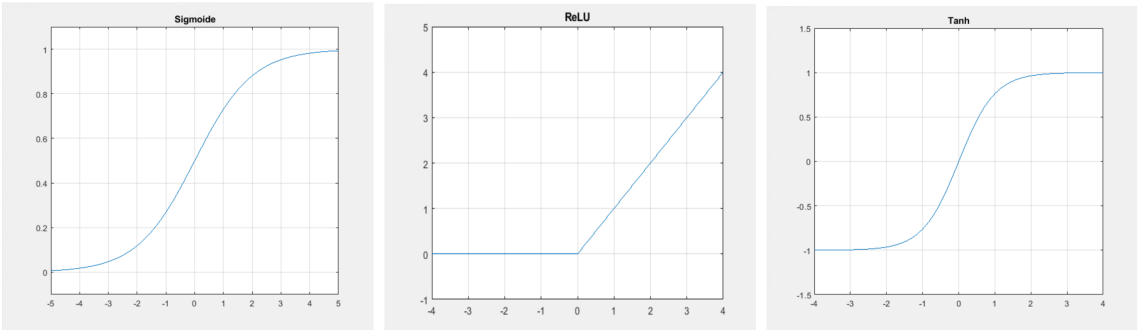
- **Sigmoide**: Función no lineal que se puede expresar mediante la fórmula matemática $f(x) = 1/(1+e^{-x})$. Para los valores negativos grandes la salida toma el valor 0 y para los valores positivos elevados toma el valor 1, mientras que la máxima variación de la salida se produce para valores próximos a cero. Su gráfica se caracteriza por tener forma de "S".

- **ReLU**: Función que para entradas negativas devuelve una salida nula y para entradas positivas la salida toma el valor de la entrada.

- **Tangente hiperbólica (tanh)**: Función que se puede representar mediante la expresión matemática $f(x) = (e^{x} - e^{-x})/(e^{x} + e^{-x})$. Para entradas negativas muy elevadas, la salida toma el valor -1, mientras que para entradas positivas muy grandes la salida toma el valor 1. Para entradas próximas a cero, la salida toma valores próximos a cero. Como la función sigmoide, su gráfica tiene una forma de "S".

- **Softmax**: Función también conocida como la función exponencial normalizada. Para valores negativos de la entrada muy elevados la salida toma el valor 0, mientras que para valores positivos muy grandes la salida toma el valor 1. Esta función de activación se suele utilizar en las neuronas de la última capa para tareas de clasificación, ya que la salida será un vector de probabilidades normalizadas a 1.

En la Figura **1-10** se muestran las funciones de activación sigmoide, ReLU y tangente hiperbólica.



(a) (b) (c)

Figura 1-10: Funciones de activación (a) Sigmoide, (b) ReLU y (c) Tangente hiperbólica.

Las neuronas de una red pueden distribuirse y conectarse entre ellas de muchas formas, dando lugar a múltiples configuraciones de redes neuronales. Las redes neuronales se organizan en capas, que pueden ser de entrada, de salida o capas ocultas. Las neuronas de las capas de entrada no tienen pesos asociados a ellas, únicamente transmiten la información sin modificarla, mientras que las presentes en las capas de salida y las capas ocultas sí que tienen pesos asociados. Las capas ocultas son las capas intermedias de la red y reciben ese nombre porque no están directamente conectadas a los datos de entrada y salida. En función del número de capas ocultas, las redes neuronales se pueden agrupar en dos clases: las redes de una sola capa y las redes multicapa, que a su vez se pueden subdividir en dos grupos según el número de capas ocultas que contienen, las redes poco profundas y las redes profundas. A medida que aumenta el número de capas, la red es capaz de aprender relaciones más complejas y jerárquicas a partir de los datos de entrada. En la Figura **1-11** se muestra un ejemplo de cada tipo de red.

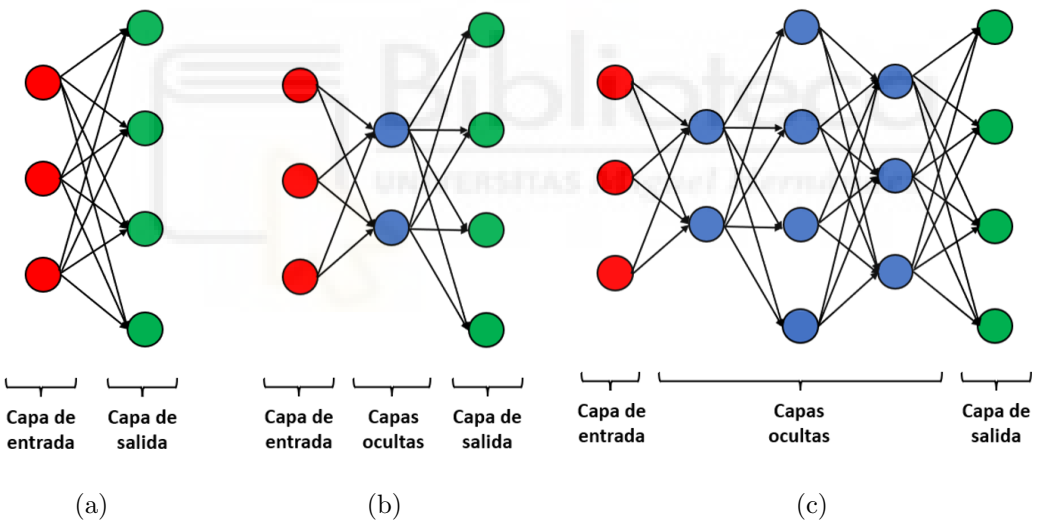


Figura 1-11: (a) Red de una sola capa, (b) Red multicapa poco profunda y (c) Red multicapa profunda.

Para resolver un problema específico, las redes neuronales deben ser entrenadas para realizar dicha tarea. El proceso de entrenamiento consiste en introducir datos a la red para los cuales la red devolverá una salida. Según el tipo de aprendizaje, la salida obtenida se compara con una etiqueta que indica la salida real para determinar el error cometido, como es el caso del aprendizaje supervisado, o bien se compara con las salidas obtenidas para otros datos para tratar de encontrar correspondencias o patrones entre los datos, como sucede en el aprendizaje no supervisado. Dentro

del aprendizaje supervisado se encuentran las funciones de pérdida, que consisten en expresiones matemáticas que tratan de determinar el error cometido en la predicción. La función de pérdida recibe como entrada la salida de la red y la etiqueta correspondiente. A partir de estos datos, la función es evaluada y devuelve un valor. En función del resultado obtenido, el algoritmo de optimización empleado reajustará los pesos de la red en mayor o menor medida.

En los últimos años, se han explorado distintas arquitecturas de redes neuronales, dando lugar a las arquitecturas compuestas. Por ejemplo, las Redes Siamesas están formadas por dos redes neuronales idénticas que funcionan en paralelo y comparten sus pesos internos, pero cada subred recibe un dato de entrada distinto y por tanto devuelve una salida distinta. Este tipo de redes son entrenadas con pares de datos, que pueden ser similares o diferentes. La red debe ser capaz de aprender patrones de similitud o diferencia entre los datos.

Continuando esta línea surgieron las Redes Tripletas, las cuales contienen tres subredes idénticas en paralelo que comparten sus pesos. De la misma forma que en las Redes Siamesas, cada subred recibe una entrada distinta y devuelve una salida distinta. Las redes tripletas son entrenadas con combinaciones de tres datos: un dato ancla, un dato positivo y un dato negativo, los dos primeros similares y el último diferente. La red debe encontrar similitudes entre los dos primeros y diferencias del tercero con los dos anteriores.

En la Figura **1-12** se muestra una comparativa de la estructura de las redes simples, siamesas y tripletas, donde x_i e y_i son los datos de entrada y salida de la red, respectivamente.

En este trabajo, se han empleado redes tripletas para llevar a cabo la tarea de localización de robots móviles. Las redes tripletas utilizadas reciben tres imágenes panorámicas y cada una de las subredes que la conforman obtiene el descriptor global de una de las imágenes. Los descriptores obtenidos son comparados mediante una función de pérdida de triplete.

Las funciones de pérdida de triplete minimizan su valor cuando los descriptores de las imágenes ancla y positiva evaluadas por la red son similares y el descriptor de la imagen negativa es diferente a los dos anteriores. El parámetro que determina cómo de similares deben ser el dato positivo y el negativo respecto al ancla es el margen. Cuando este valor es cero, la función de pérdida devolverá un valor próximo a cero cuando el dato negativo es similar a los dos anteriores. Por otro lado, cuanto mayor sea el margen, la diferencia del dato negativo respecto a los otros dos datos debe ser

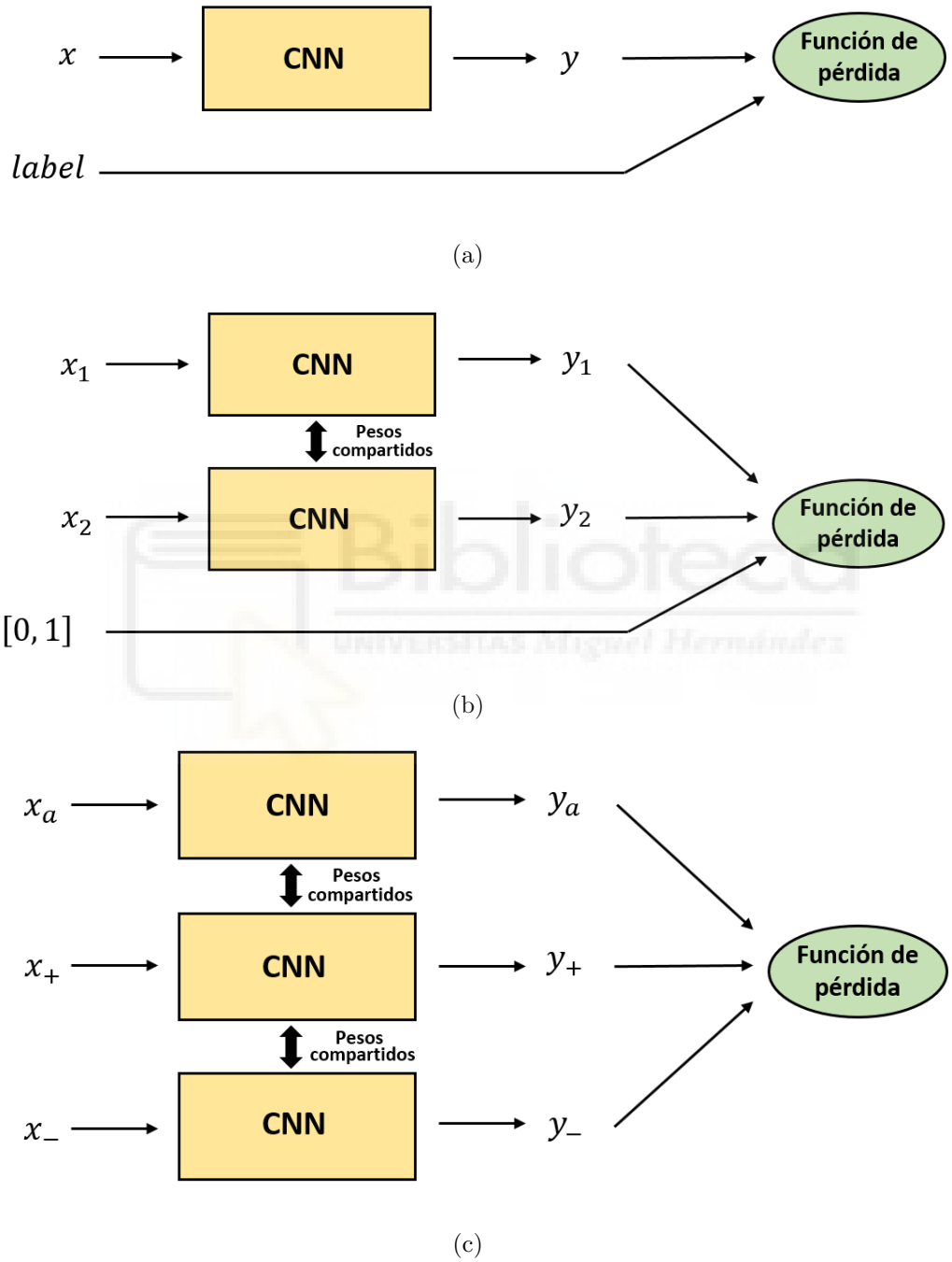


Figura 1-12: Estructura de las Redes (a) Simples, (b) Siamesas y (c) Tripletas.

mayor para reducir el error cometido (Figura **1-13**).

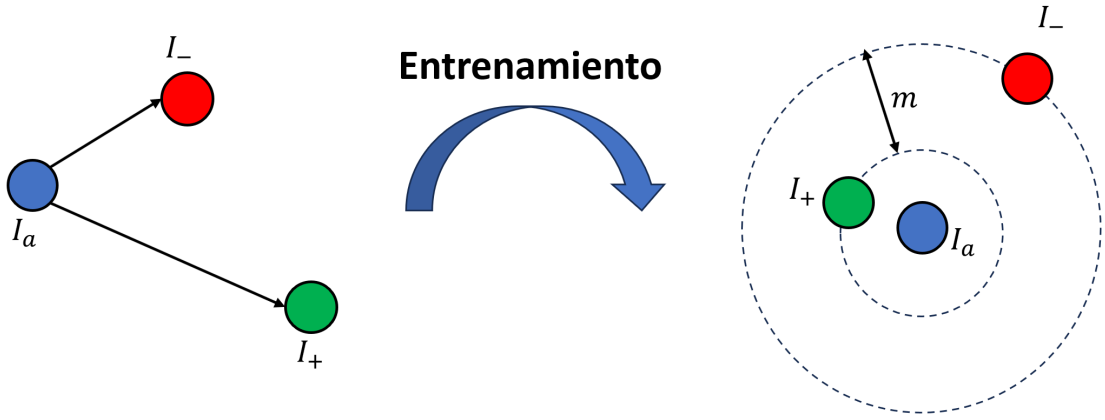


Figura 1-13: Proceso de entrenamiento de una red neuronal mediante funciones de pérdida de triplete.

Existe una gran variedad de funciones de pérdida de triplete. Algunas de ellas realizan la comparación de los descriptores mediante la distancia euclídea. Otras, en cambio, utilizan la similitud coseno o la distancia coseno para comparar los descriptores. Cada una de ellas realiza una función matemática diferente dando lugar a un abanico de complejidad que abarca desde sumas y restas hasta funciones exponenciales y logarítmicas. En este trabajo se ha realizado el entrenamiento de Redes Neuronales Convolucionales Tripletas empleando distintas funciones de pérdida de triplete y analizando su influencia en el desempeño de la red para la tarea de localización.

El resto del trabajo está estructurado de la siguiente forma. El capítulo 2 corresponde al Estado del Arte, en el que se han comentado las investigaciones más recientes en relación a la localización de robots móviles. En el capítulo 3 se han enumerado las herramientas y métodos empleados para llevar a cabo la tarea de localización de robots móviles. En el capítulo 4 se ha desarrollado el método propuesto en este trabajo para realizar la localización jerárquica y global mediante Redes Neuronales Tripletas. En el capítulo 5 se muestran los experimentos realizados y los resultados obtenidos. Por último, en el capítulo 6 se explican las conclusiones extraídas de los experimentos y se comentarán las posibles líneas de investigación futuras.

2 ESTADO DEL ARTE

Actualmente, los robots móviles autónomos son empleados en una gran variedad de aplicaciones debido a su capacidad para realizar tareas que suponen una mayor dificultad para las personas. De la misma forma, con el aumento de la capacidad de cómputo de los ordenadores actuales las herramientas de Inteligencia Artificial se han convertido en una de las principales opciones para abordar la creación de mapas y la localización de robots móviles. En los últimos años han surgido distintas líneas de investigación que tratan de proponer soluciones a estas tareas.

2.1. CREACIÓN DE MAPAS Y LOCALIZACIÓN DE ROBOTS MÓVILES

La creación de mapas y la localización son tareas esenciales para la navegación segura de los robots móviles. En cuanto a la creación de mapas, Wolf y Sukhatme [42] realizaron un mapping semántico para representar la transitabilidad de entornos de exterior, combinando técnicas de *Machine Learning* con algoritmos de mapping tradicionales, empleando Modelos de Markov Ocultos y Máquinas de Vector Soporte.

Para abordar el problema de localización, Leonard y Durrant-Whyte [19] emplearon "balizas geométricas", que consisten en puntos o zonas del entorno que permiten reconocer con facilidad en qué zona de la escena se encuentra el robot. Posteriormente, Betke y Gurvits [5] utilizaron marcas de referencia a partir de las cuales el robot es capaz de estimar su posición y orientación.

En los últimos años, los esfuerzos se han centrado en el desarrollo de técnicas de SLAM (*Simultaneous Localization and Mapping*) con el objetivo de llevar a cabo la tarea de localización de robots móviles aún cuando no se dispone de un mapa del entorno. Por ejemplo, Thrun *et al.* [37] y Hahnel *et al.* [14] emplearon algoritmos probabilísticos para realizar las tareas de mapeo y localización simultáneamente, mientras que Wolf y Sukhatme [41] diseñaron un algoritmo capaz de diferenciar las partes estáticas y cambiantes de una escena.

Los sensores ópticos han demostrado aportar una gran cantidad de información a costa de un coste muy bajo. Es por ello que cada vez más se escogen este tipo de sensores para extraer información del entorno. Dentro de este campo destacan, entre otros, Murray y Jennings [25], que utilizaron un sistema de visión estéreo trinocular para llevar a cabo las tareas de *mapping* y *path planning*, o Royer *et al.* [29], que abordaron el problema de localización en exteriores de un robot móvil con sistema monocular en tres etapas. En primer lugar, un robot sigue una trayectoria y graba la secuencia. En segundo lugar, se reconstruye el mapa tridimensional a partir de esta secuencia. Por último, el robot hace uso del mapa 3D para seguir la misma secuencia u otra ligeramente distinta si es conveniente.

Dentro de los sensores de visión, las cámaras omnidireccionales se han convertido en una de las opciones más utilizadas. Esto se debe a que permiten extraer información en un amplio campo de visión, lo cual permite reducir el número de imágenes necesarias para crear un mapa del entorno. Además, proporcionan la misma información independientemente de la orientación del robot. Menegatti *et al.* [22], por ejemplo, hicieron uso de este tipo de cámaras para llevar a cabo la tarea de localización en interiores mediante búsqueda de correspondencias y algoritmos de Monte Carlo. Además, Gaspar *et al.* [12] llevó a cabo la navegación de robots móviles mediante una cámara omnidireccional con un sistema catadióptrico.

2.2. DESCRIPCIÓN DE IMÁGENES

En lo que respecta a la descripción de imágenes, existen dos grandes líneas de investigación: el uso de descriptores globales y la descripción basada en puntos característicos.

Dentro del primer grupo, Payá *et al.* [26] llevaron a cabo el modelado de entornos mediante imágenes omnidireccionales y descriptores de apariencia global, mientras que Murillo *et al.* [24] realizaron una localización mediante descriptores gist.

Por otro lado, Murillo *et al.* [23] abordaron el problema de *Image Retrieval* mediante descriptores SURF en imágenes omnidireccionales. Además, Andreasson *et al.*, [1] realizaron un estudio con distintos descriptores locales para la tarea de localización de robots móviles, obteniendo un mejor rendimiento con descriptores SIFT modificados. También, Se *et al.* [32] llevaron a cabo la tarea de SLAM haciendo uso de descriptores SIFT, realizando una comparativa entre dos métodos basados en la transformada de Hough y el algoritmo RANSAC.

La descripción de imágenes no solo se puede llevar a cabo mediante técnicas analíticas, también se utilizan herramientas de Inteligencia Artificial. Por ejemplo, Sarlin *et al.* [31] entrenaron una red neuronal convolucional capaz de extraer descriptores globales y características locales a partir de las imágenes de entrada para realizar una localización jerárquica.

2.3. APRENDIZAJE PROFUNDO

Aunque el auge de la Inteligencia Artificial se ha producido en los últimos años, los primeros trabajos y conceptos teóricos relacionados con este campo se realizaron a mediados del siglo XX. En 1943, McCulloch y Pitts [21] sentaron las bases teóricas del funcionamiento de las redes neuronales artificiales y sus posibles aplicaciones. Quince años más tarde, Rosenblatt [28] publicó "Perceptrón", en el que se realizó la primera aproximación a las redes neuronales artificiales. En 1974, Paul Werbos desarrolló en su tesis "Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences" el concepto teórico de la retropropagación, que hacía referencia a un nuevo método para la actualización de los pesos y umbrales, con el objetivo de entrenar redes neuronales de forma más eficiente. Una década más tarde, Rumelhart *et al.* [30] desarrollaron el algoritmo que llevó a la realidad la idea propuesta por Werbos, dando lugar a la posterior creación de las primeras redes neuronales convolucionales (CNNs) a finales del siglo XX.

La primera CNN que surgió fue LeNet, creada por Lecun *et al.* [18], que era capaz de clasificar imágenes de caracteres sencillos escritos a mano. En los últimos años han surgido nuevas redes orientadas a la tarea de clasificación de imágenes, como es el caso de AlexNet (Krizhevsky *et al.*, [17]), GoogLeNet (Szegedy *et al.*, [35]) y los modelos VGG (Simonyan y Zisserman, [33]), entre otras. Estas arquitecturas han sido entrenadas con la base de datos ImageNet (Deng *et al.*, [10]) y son capaces de clasificar 1000 objetos diferentes. Los estudios más recientes exploran el uso de transformers visuales, otro tipo de redes neuronales enmarcadas dentro del *Deep Learning*, para abordar este tipo de tareas (Dosovitskiy *et al.*, [11]). Por otro lado, también hay estudios que han tratado de crear arquitecturas orientadas a una tarea específica. Por ejemplo, Li *et al.* [20] desarrollaron una CNN poco profunda capaz de clasificar imágenes de pulmones sanos y dañados.

En cuanto a la localización de robots móviles, en los últimos años han surgido estudios que han empleado CNNs para abordar este problema. Tai y Liu [36] entrenaron una CNN para llevar a cabo una localización en interiores evitando el impacto con obstáculos haciendo uso de un único sensor RGB-D, mientras que Xu *et al.* [44]

utilizaron una cámara monocular y un sensor láser para entrenar una CNN capaz de realizar una localización gruesa y una localización fina, y emplearon un algoritmo basado en una red LSTM que permitía al robot regresar al camino especificado cuando se desviaba de su trayectoria. Por otro lado, Arroyo *et al.* [2] entrenaron a una CNN con el objetivo de llevar a cabo una localización en exteriores robusta frente a cambios de estación, comprimiendo los descriptores obtenidos por la red en un número reducido de bits y comparando los nuevos descriptores mediante la distancia Hamming.

Otras investigaciones se centraron en entrenar CNNs con imágenes omnidireccionales para llevar a cabo la tarea de localización. Por ejemplo, Wang *et al.* [40] entrenaron una CNN capaz de calcular la distancia entre la imagen de test y la más cercana del mapa, con el fin de realizar el seguimiento de una trayectoria. Ballesta *et al.* [4] emplearon descriptores holísticos para llevar a cabo una localización jerárquica, mientras que Cebollada *et al.* [8] y Cabrera *et al.* [6] entrenaron una CNN para identificar la estancia en la que se encuentra un robot y para extraer un descriptor global a partir de una de las capas de la red con el objetivo de identificar la posición del robot dentro de una estancia.

En ocasiones, a la hora de entrenar una CNN para realizar una tarea es conveniente partir de los pesos y umbrales de modelos de red preentrenados para la misma tarea u otra similar, con el fin de aprovechar los conocimientos ya adquiridos por la red. Esta técnica es conocida como *Transfer Learning*. Wozniak *et al.* [43], por ejemplo, reentrenaron una CNN partiendo de los pesos de la red VGG-F para realizar una clasificación entre 16 habitaciones diferentes. Por otro lado, Ballesta *et al.* [4] emplearon esta técnica para transformar una CNN utilizada para la clasificación entre habitaciones en una CNN de regresión para determinar las coordenadas en las que se encuentra el robot dentro de la estancia seleccionada en la fase de clasificación. Otros trabajos mencionados anteriormente también hicieron uso de *Transfer Learning* ([44], [8] y [6]).

En los últimos años, muchas investigaciones se centran en el desarrollo de arquitecturas compuestas para llevar a cabo la tarea de localización de robots móviles. Por ejemplo, Yin *et al.* [45] realizaron una localización 3D jerárquica con nubes de puntos empleando Redes Siamesas y un sensor LiDAR, mientras Zhang *et al.* [46] entrenaron este tipo de redes para llevar a cabo un seguimiento visual.

Posteriormente, se empezaron a utilizar Redes Tripletas para llevar a cabo la tarea de *Image Retrieval* debido a las ventajas que presentan frente a las Redes Siamesas, como un mayor número de ejemplos de entrenamiento y la misma proporción de

imágenes similares y diferentes en el proceso de aprendizaje de la red. Por ejemplo, Gordo *et al.* [13] emplearon descriptores R-MAC para entrenar redes tripletas para la tarea de *Image Retrieval*. También se han realizado algunos trabajos con Redes Cuadrupletas, como es el caso de Chen *et al.* [9], que entrenaron este tipo de redes para la tarea de identificación de personas.

Con la aparición de las Redes Tripletas, muchos estudios se han centrado en desarrollar funciones de pérdida de triplete que maximicen el rendimiento de la red para tareas de *Image Retrieval*. Por ejemplo, Hermans *et al.* [15] realizaron una comparación entre distintas funciones de pérdida, entre ellas las funciones *Triplet Margin Loss*, *Lifted Embedding Loss* o *Batch Hard Loss*, para la tarea de reconocimiento de personas. Por otro lado, Uy y Lee [38] emplearon las funciones de pérdida *Lazy Triplet Loss* y *Lazy Quadruplet Loss* para abordar el problema de localización mediante redes MLP entrenadas con nubes de puntos.

Otros estudios se centran en desarrollar funciones de pérdida que utilizan métricas distintas a la distancia euclídea. Dentro de este grupo destacan Sun *et al.* [34], que crearon la función de pérdida *Circle Loss*, que emplea la similitud coseno para comparar los descriptores obtenidos de la red, para llevar a cabo diferentes tareas de *Image Retrieval* como el reconocimiento de caras y de personas, o Wang *et al.* [39], que utilizaron la función de pérdida *Angular Loss*, cuya principal novedad es el uso de un margen angular.

3 HERRAMIENTAS UTILIZADAS

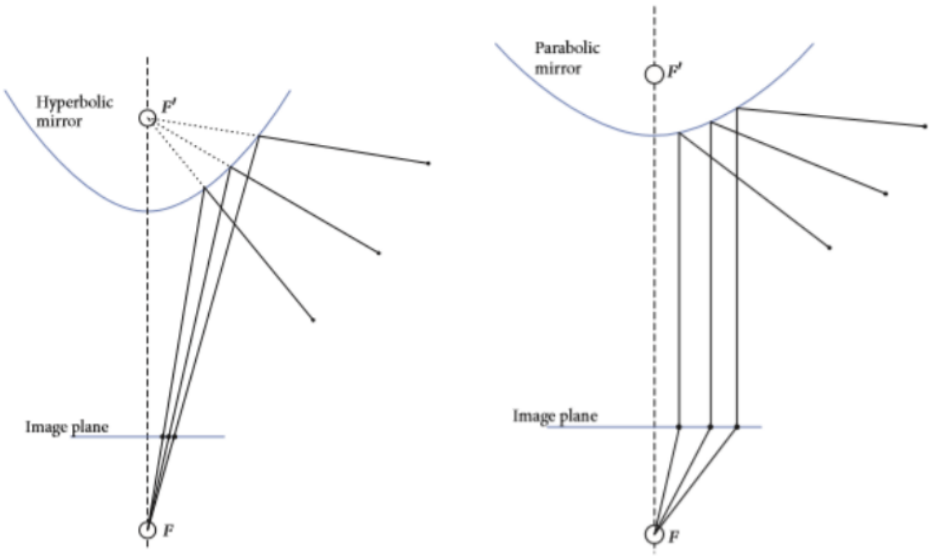
En este trabajo se han empleado distintas herramientas para realizar la tarea de localización de robots móviles, entre ellas la visión omnidireccional y las redes neuronales convolucionales. Asimismo, en lo que respecta a estas redes, se ha realizado un estudio de las funciones de pérdida de triplete y de la influencia del proceso de selección de los ejemplos de entrenamiento. En los siguientes apartados se describe cada uno de estos conceptos.

3.1. VISIÓN OMNIDIRECCIONAL

Los sensores ópticos son cada vez más utilizados en el área de robótica móvil, ya que a partir de una imagen permiten extraer una extensa variedad de información a un bajo coste y peso. Dentro del campo de la robótica móvil, se pueden emplear para abordar tareas como la detección y elusión de objetos, segmentación de zonas transitables, *path planning*, creación de mapas y localización entre otras tareas. Dentro de este grupo destacan las cámaras omnidireccionales, también conocidas como cámaras de 360 grados.

Las cámaras omnidireccionales pueden tener distintas configuraciones. Generalmente, se suele optar por un sistema catadióptrico debido a su sencillez. Mediante una única cámara, los rayos de luz que inciden sobre un espejo cóncavo son reflejados, formando una imagen omnidireccional. Según el tipo de curvatura del espejo, el sistema catadióptrico puede ser hiperbólico (Figura **3-1** (a)) o parabólico (Figura **3-1** (b)), entre otros.

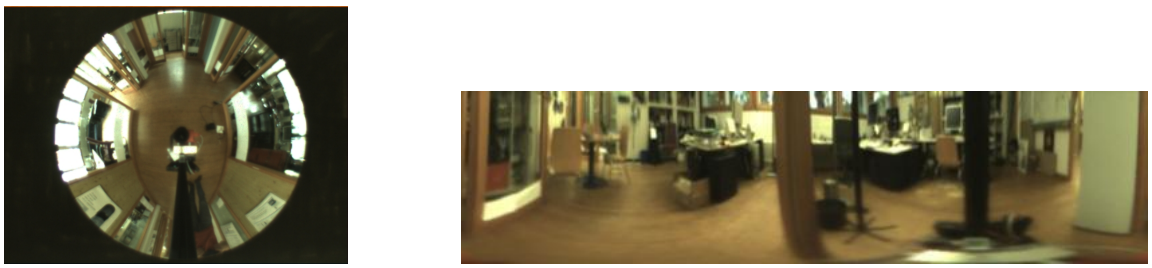
Una cámara omnidireccional es una de las más idóneas para realizar la tarea de localización de robots móviles por varios motivos. En primer lugar, son capaces de capturar la luz en un amplio campo de visión, lo cual permite extraer información abundante del entorno con una menor cantidad de imágenes. Otra de las ventajas de este tipo de cámaras es que permiten obtener la misma información independientemente de la orientación del robot.



(a) (b)

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

En el presente trabajo, se han convertido las imágenes omnidireccionales a panorámicas, ya que presentan mejores resultados en la tarea de localización ([8], [6]). Como la red neuronal emplea convoluciones matriciales, tener la información en formato panorámico permite una mejor extracción más natural de las características de la imagen con este tipo de convoluciones.



(a) (b)

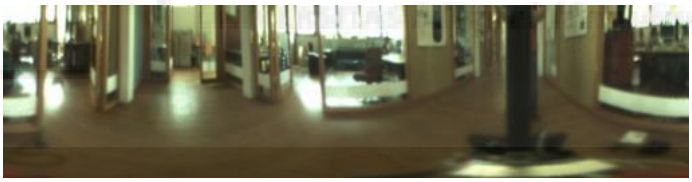
Figura 3-2: (a) Imagen omnidireccional, (b) Imagen omnidireccional convertida a panorámica.

3.2. BASE DE DATOS COLD

Para este trabajo se han utilizado imágenes del conjunto de datos COLD-Friburgo, que forma parte de la base de datos COLD (COsy Localization Database) [27]. COLD contiene imágenes omnidireccionales de 640x480 píxeles con 3 capas RGB, que se han transformado posteriormente en imágenes panorámicas de 512x128x3 píxeles (Figura **3-3**).

(a)

(b)



(c)

Figura 3-3: Ejemplo de imágenes contenidas en la base de datos COLD-Friburgo para las tres condiciones de iluminación: (a) Nublado, (b) Noche y (c) Soleado. Las imágenes contenidas en esta base de datos pueden descargarse a través de su página web https://www.cas.kth.se/COLD/.

La base de datos COLD-Friburgo contiene imágenes tomadas por un robot móvil, el cual lleva incorporado un sistema de visión catadióptrico omnidireccional con un espejo hiperbólico (Figura **3-4**). El robot recorre una trayectoria en el interior del edificio a lo largo de 9 habitaciones distintas (Figura **3-5**), tomando una fotografía cada 0,08 s. El edificio a través del cual se han capturado las diferentes fotografías que conforman la base de datos es el *Autonomous Intelligent Systems Laboratory* de la Universidad de Friburgo, Alemania. En su interior se pueden encontrar estancias de varios tipos, como por ejemplo oficinas, unas escaleras, un baño, una cocina, una sala con impresora y un pasillo que une todas las habitaciones de la planta

baja del edificio (Figura **3-6**). En la Tabla **3-1** se muestra el número de imágenes por estancia que conforman el conjunto de datos de entrenamiento. Se han tomado imágenes en distintos instantes temporales y bajo tres condiciones de iluminación distintas: nublado, soleado y noche. Además, la base de datos incluye personas en movimiento y cambios en la ubicación del mobiliario durante el proceso de adquisición de imágenes. Todo ello proporciona un conjunto de imágenes completo y con muchos ejemplos desafiantes debido a los cambios de iluminación y en el entorno.

Figura 3-4: Ejemplo de robot móvil autónomo ActivMedia Pioneer-3.

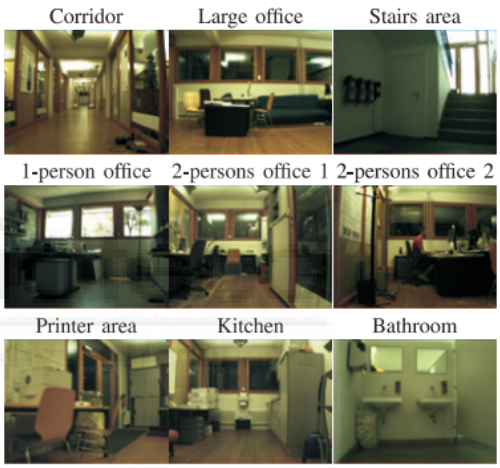


Figura 3-5: Estancias del edificio Friburgo de la base de datos COLD.

El objetivo del presente trabajo es realizar el entrenamiento de una red neuronal con un conjunto reducido de imágenes tomadas en un instante temporal y con una condición de iluminación concretos. La red permite comprimir la información visual de una imagen en un único descriptor, que permitirá obtener la posición del robot mediante la búsqueda del vecino más cercano entre una imagen de test y el mapa del edificio.

De acuerdo con la filosofía de este trabajo, para el entrenamiento y la validación de la red se ha escogido un conjunto de imágenes cloudy, debido a que se trata de la condición de iluminación más estándar y la que presenta un menor contraste de luz entre el interior y el exterior del edificio, mientras que para el test de la red se han utilizado las tres condiciones de iluminación, con el fin de comprobar la robustez

de la red frente a cambios lumínicos. Para ello, se ha utilizado la secuencia Path 2 Cloudy 3 para el entrenamiento y la validación de las redes neuronales tripletas. Se ha muestreado la carpeta uniformemente y dividido esta secuencia en dos conjuntos totalmente independientes. De esta manera, a partir de las 2778 imágenes que conforman el recorrido, se han seleccionado 556 imágenes para realizar el entrenamiento y otras 556 imágenes para llevar a cabo la validación. Para el testeo de redes tripletas, se han utilizado las secuencias Path 2 Cloudy 2, Path 2 Night 2 y Path 2 Sunny 2, que contienen 2595 imágenes en condición de iluminación nublado, 2707 imágenes en condición de noche y 2114 imágenes en condición soleado respectivamente.

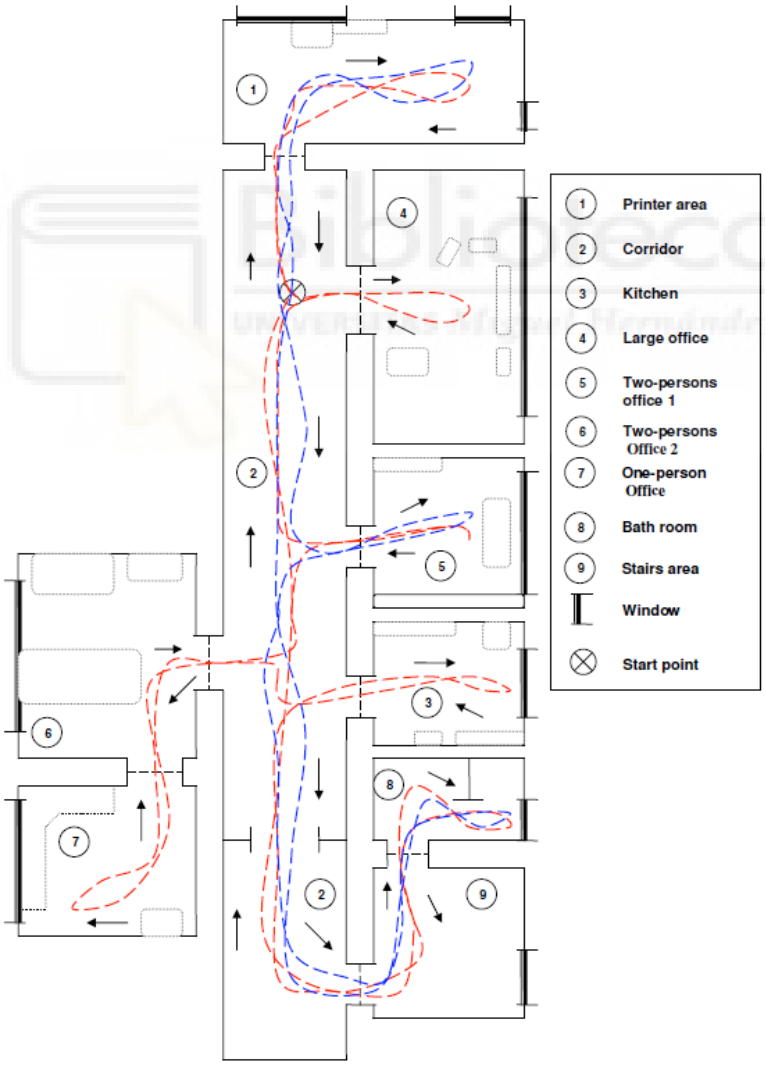


Figura 3-6: Planta del edificio Friburgo de la base de datos COLD.

Estancia	1PO-A	2PO1-A	2PO2-A	CR-A	KT-A	LO-A	PA-A	ST-A	TL-A
Nº estancia	7	5	6	2	3	4	1	9	8
Nº imágenes	44	46	31	238	46	26	57	30	38

Tabla 3-1: Número de imágenes por habitación del conjunto de datos de entrenamiento.

3.3. REDES NEURONALES TRIPLETAS

Las redes neuronales convolucionales forman parte del *Deep Learning*, una de las ramas de la inteligencia artificial. Su funcionamiento consiste en recibir un conjunto de datos en formato matricial, extraer información de estos y realizar una predicción. Durante el proceso de entrenamiento, si la predicción es errónea, el algoritmo de optimización debe ser capaz de reajustar los pesos internos para tratar de realizar las siguientes predicciones correctamente, imitando el método de aprendizaje del cerebro humano. El algoritmo de optimización empleado en el entrenamiento de la red ha sido el *Stochastic Gradient Descent* (SGD).

Hasta la fecha, la tarea de localización de robots móviles mediante redes neuronales se ha abordado tanto con arquitecturas simples como con arquitecturas compuestas. Existen estudios previos sobre la localización de robots móviles que han empleado redes simples y compuestas. Por ejemplo, Cabrera *et al.* [6] llevaron a cabo el entrenamiento de una CNN simple para abordar la tarea de localización. Por otro lado, Cabrera *et al.* [7] realizaron un estudio de las redes neuronales siamesas para la creación de modelos visuales y localización de robots móviles.

Las redes neuronales simples reciben una imagen como entrada, procesan su información, extraen características de la imagen y devuelven una predicción. Si la predicción coincide con el resultado real, se considera como un acierto de la red. Si existe un error entre la predicción y el valor real del ground truth, se debe reajustar los pesos internos de la red para tratar de corregir las futuras predicciones.

Las Redes Neuronales Siamesas, en cambio, son dos CNN idénticas que funcionan en paralelo. Las dos redes comparten la misma arquitectura interna y los mismos pesos, pero cada una recibe una imagen distinta y devuelve un descriptor distinto. Este tipo de redes presenta una ventaja respecto a las redes simples, ya que se pueden comparar dos imágenes, iguales o diferentes, al mismo tiempo, estudiando conceptos como similitud o diferencia.

Las Redes Neuronales Tripletas van un paso más allá respecto a las Redes Neuronales Siamesas. Consisten en tres CNN idénticas que funcionan en paralelo. Las tres redes comparten la misma arquitectura interna, pero reciben una imagen distinta cada una y devuelven un descriptor distinto. La ventaja principal de las Redes Tripletas respecto a las Redes Siamesas es que se pueden comparar ejemplos iguales y diferentes al mismo tiempo, permitiendo así que la red se ajuste por igual a imágenes similares y diferentes durante el proceso de entrenamiento. Otra ventaja importante es que el número de combinaciones posibles de imágenes aumenta exponencialmente respecto al uso de redes siamesas a la hora de llevar a cabo dicho entrenamiento. Esto puede ser de utilidad ante un conjunto de datos insuficiente, evitando tener que realizar un Data Augmentation o tomar una cantidad mayor de imágenes. (Figura **3-7**).

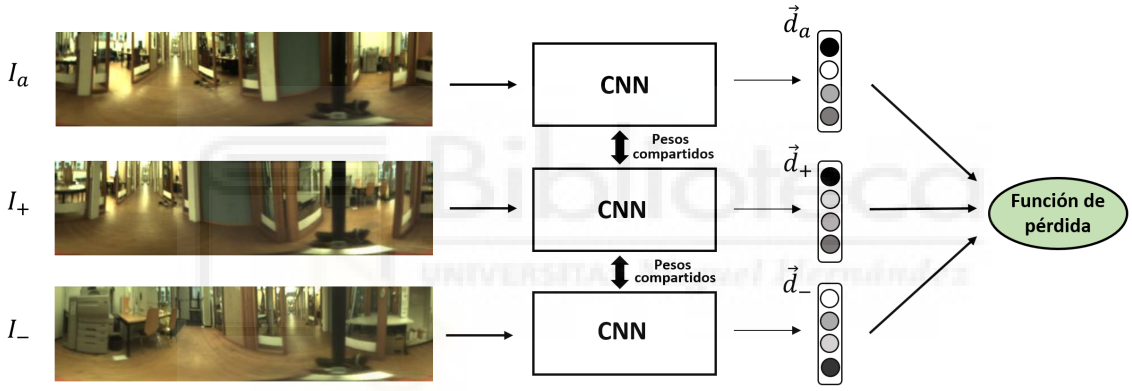


Figura 3-7: Estructura de las redes tripletas.

En este trabajo, la tarea de localización de un robot móvil se ha realizado mediante el uso de redes neuronales tripletas.

3.4. ARQUITECTURA INTERNA DE LA RED VGG16

Para realizar la tarea de localización de robots móviles, la red neuronal utilizada ha sido VGG16, ya que ha dado resultados muy positivos en experimentos similares en comparación con otros modelos de redes neuronales (Cabrera *et al.*, [7]). Esta red, creada por Simonyan y Zisserman [33], es capaz de clasificar imágenes en 1000 categorías de objetos, y de mostrar representaciones con un alto número de características para una gama extensa de imágenes. Como no es objeto de estudio del

presente trabajo, únicamente se ha utilizado este modelo de red durante el desarrollo de los experimentos realizados. Además, se ha empleado el modelo de VGG16 preentrenada como punto de partida de los entrenamientos, para conservar sus conocimientos ya adquiridos y su capacidad de extracción de características. Esta técnica es conocida como *Transfer Learning*.

Para extraer las características de una imagen y procesar la información correctamente, son necesarias varias capas. A continuación, se van a describir las capas que conforman la estructura de una red neuronal convolucional:

- **Capa convolucional:** Este tipo de capas se utilizan para extraer características relevantes de la imagen, como pueden ser bordes, esquinas o texturas. Esto se realiza mediante la aplicación de numerosos filtros de convolución sobre la imagen. Esto permite a la red ser muy útil frente a tareas como la clasificación de imágenes o el reconocimiento de objetos.

- **Capa totalmente conectada:** Su función principal es transformar la información de una matriz de características en un vector. Su nombre se debe a que cada neurona de esta capa está conectada con todas neuronas de la capa anterior. La salida será una combinación lineal de las entradas multiplicadas por los pesos internos de la capa totalmente conectada. Para realizar la clasificación de objetos se utiliza esta capa seguida de una Softmax, mientras que para la predicción de resultados se emplea seguida de una capa de regresión.

- **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:** Trabajar con redes neuronales convolucionales puede suponer un alto coste computacional. Este tipo de capas permiten sintetizar la información del mapa de características de la imagen, agrupando varios valores en un único término mediante distintos algoritmos (Figura **3-8**):

  - Average Pooling: Para cada agrupación de datos, se devuelve el valor promedio de todos los elementos.
  - Max Pooling: Para cada agrupación de datos, se devuelve el valor más alto de todos los elementos.
  - Sum Pooling: Para cada agrupación de datos, se devuelve la suma de todos los elementos.

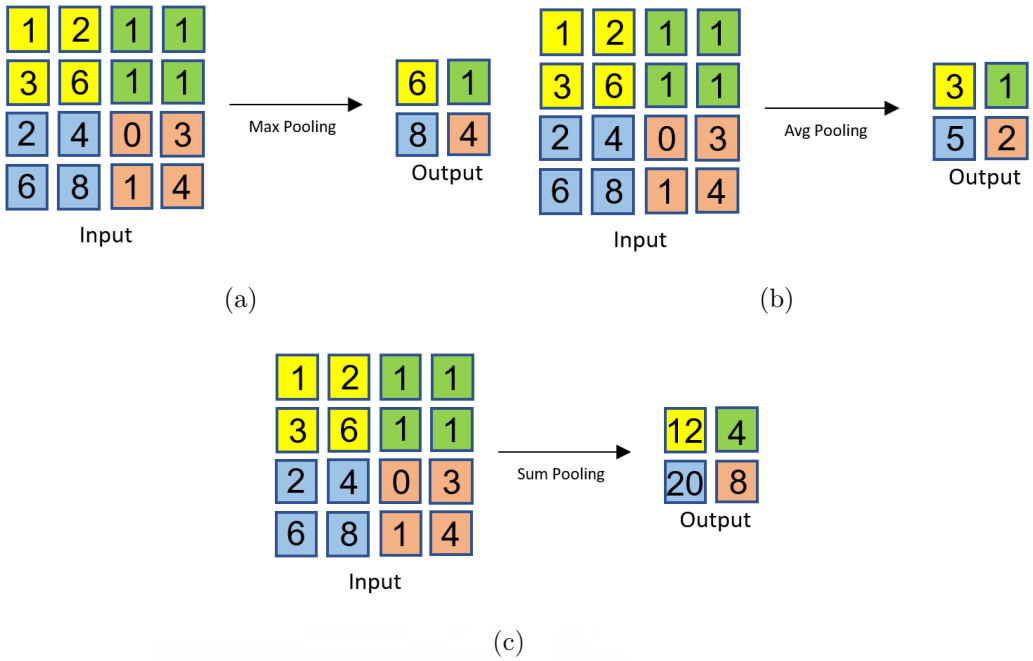


Figura 3-8: Ejemplo de operaciones de Pooling: (a) Max Pooling, (b) Average Pooling y (c) Sum Pooling.

Estas capas se pueden combinar de distintas formas y en distinto orden, dando lugar a una gran variedad de arquitecturas. La arquitectura interna del modelo de red VGG16 está descrita en [33] y se puede observar en la Figura **3-9**.

3.5. DESCRIPTORES DE APARIENCIA GLOBAL

La descripción de imágenes mediante descriptores globales es una técnica muy utilizada para la tarea de *Image Retrieval*. Se puede realizar de dos formas: mediante la extracción de características globales de la imagen de forma analítica, o bien mediante técnicas de aprendizaje profundo, mediante el uso de redes neuronales. En ambos casos, se tiene como objetivo comprimir una imagen en un único descriptor que contenga la información necesaria para agrupar las imágenes en sus diferentes categorías. En este trabajo se ha empleado la segunda técnica para realizar la localización del robot.

Para ello, se ha estudiado la similitud o la diferencia entre los vectores descriptores obtenidos de tripletas de imágenes, mediante funciones matemáticas como la distancia euclídea o la similitud coseno. Si la red ha sido entrenada correctamente, deberá

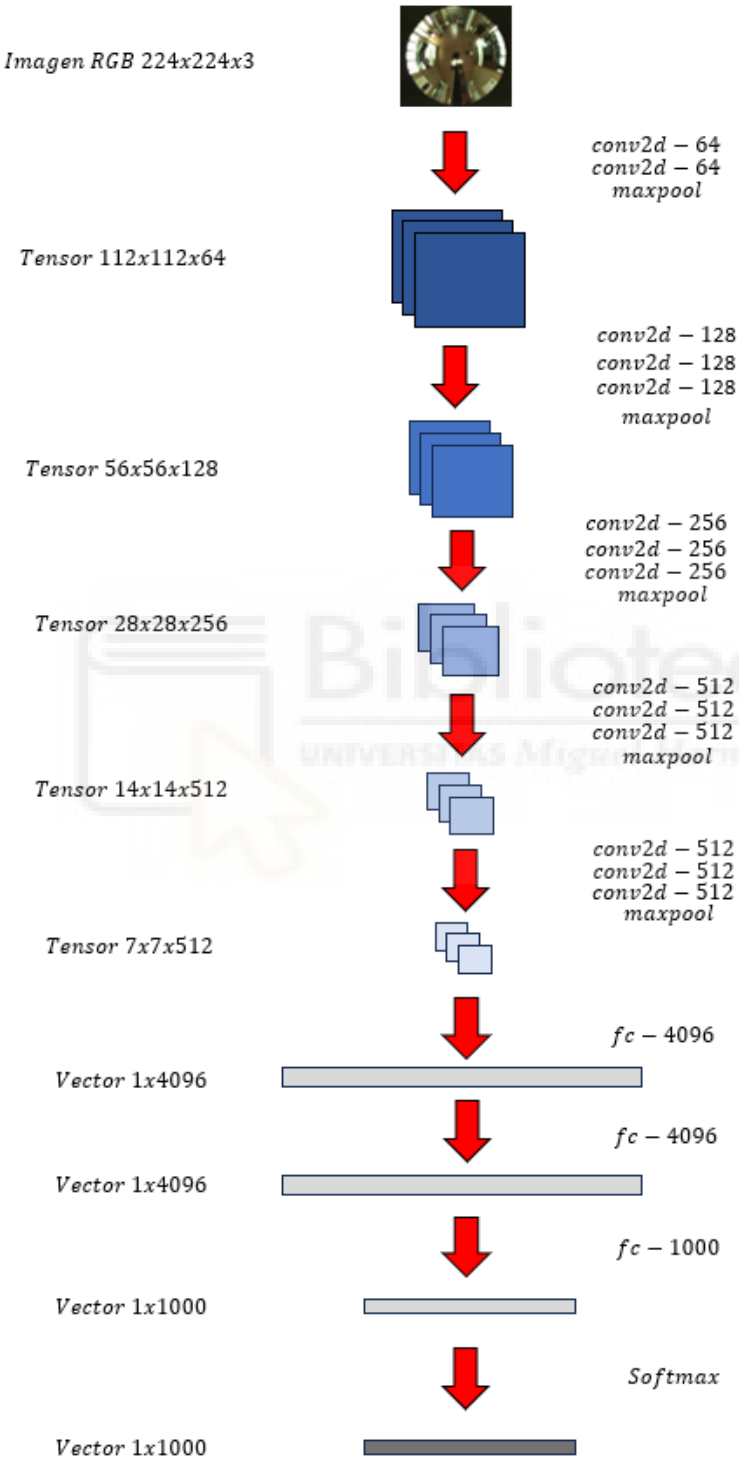


Figura 3-9: Arquitectura interna del modelo de red VGG16.

devolver una distancia menor, o bien una similitud mayor, entre imágenes similares, y al mismo tiempo devolverá una distancia mayor entre imágenes diferentes. Para realizar la asociación de imágenes, se empleará la técnica del vecino más cercano.

En este trabajo, se ha realizado la normalización de los vectores descriptores obtenidos en la capa de salida de la red neuronal, con el objetivo de manejar descriptores con el mismo módulo y de esta forma acotar las distancias entre descriptores.

3.6. FUNCIONES DE PÉRDIDA

La función de pérdida es un elemento determinante durante el entrenamiento de la red. Se encarga de calcular el error cometido por la red en su predicción. En función del error cometido, el algoritmo de optimización de la red reajustará los pesos internos de la misma en mayor o menor medida. La función de pérdida escogida para el entrenamiento de una red afecta al proceso de aprendizaje de ésta y, por lo tanto, puede ser la causa del éxito o el fracaso de la adaptación de la red al problema.

Las funciones de pérdida de triplete generalmente reciben los descriptores de tres imágenes: una imagen ancla, una imagen positiva y una imagen negativa. La imagen positiva es de la misma clase que la imagen ancla y la imagen negativa es de distinta clase. El resultado será una función matemática evaluada con los descriptores de cada imagen. Este resultado debe ser un valor cercano a 0 cuando la predicción de la red es correcta y tomará un valor mayor a medida que esta predicción se aleje de la realidad. En la Figura **3-10** se muestra un par de ejemplos correspondientes a una predicción correcta y otra errónea de la red.

En el presente trabajo se han evaluado distintas funciones de pérdida para redes neuronales tripletas y se ha realizado una comparación entre ellas. Podemos destacar las siguientes:

- **Triplet Margin Loss (Triplet Loss):** Se trata de la función de pérdida más característica de las redes neuronales tripletas. Trata de minimizar la distancia entre la imagen ancla y la imagen positiva, y maximizar la distancia entre la imagen ancla y la imagen negativa. Obtiene el valor para cada combinación de tres imágenes y calcula el promedio de todas las combinaciones del lote. Se puede definir con la siguiente expresión:

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}[D_{a,p}^{i} - D_{a,n}^{i} + m]_{+}$$

donde $D^i_{a,p}$ es la distancia euclídea entre los descriptores de la imagen ancla y la imagen positiva, $D^i_{a,n}$ es la distancia euclídea entre los descriptores de la imagen ancla y la imagen negativa, $[...]_+$ es la función relu, m es el margen y N es el número de imágenes que forman el batch.

- **Lazy Triplet Loss:** Esta función de pérdida, igual que la *Triplet Margin Loss*, trata de minimizar la distancia entre la imagen ancla y la imagen positiva, al mismo tiempo que intenta maximizar la distancia entre la imagen ancla y la imagen negativa. Sin embargo, en lugar de calcular el promedio de todas las combinaciones del lote de imágenes, devuelve el valor máximo, es decir, se queda con el valor que devuelve el ejemplo más difícil del lote y rechaza el resto. Se puede definir con la siguiente expresión:

$$\mathcal{L} = \left[\text{máx}\left(\vec{D_{a,p}} - \vec{D_{a,n}} + m\right)\right]_+$$

donde $\vec{D_{a,p}} = (D^1_{a,p}, D^2_{a,p}, ..., D^N_{a,p})$ son las distancias euclídeas entre los descriptores de cada par de imágenes ancla y positiva, $\vec{D_{a,n}} = (D^1_{a,n}, D^2_{a,n}, ..., D^N_{a,n})$ son las distancias euclídeas entre los descriptores de cada par de imágenes ancla y negativa $[...]_+$ es la función relu, m es el margen y N es el tamaño del batch.

- **Semi Hard Loss:** Esta función de pérdida es una variante de la *Lazy Triplet Loss*. Calcula la media de todas las distancias entre imágenes ancla e imágenes positivas, y obtiene la distancia mínima de todas las distancias entre imágenes ancla e imágenes negativas. Es decir, escoge el ejemplo negativo más difícil para la red.

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}\left[D^i_{a,p} - \text{mín}\left(\vec{D_{a,n}}\right) + m\right]_+$$

donde $D^i_{a,p}$ es la distancia euclídea entre los descriptores de la imagen ancla y la imagen positiva, $\vec{D_{a,n}} = (D^1_{a,n}, D^2_{a,n}, ..., D^N_{a,n})$ son las distancias euclídeas entre los descriptores de cada par de imágenes ancla y negativa, $[...]_+$ es la función relu, m es el margen y N es el tamaño del batch.

- **Batch Hard Loss:** Esta función de pérdida es otra variante de la Lazy Triplet Loss. Obtiene la distancia mínima de todas las distancias entre imágenes ancla e imágenes positivas, y obtiene la distancia máxima de todas las distancias entre imágenes ancla e imágenes negativas. Es decir, escoge el ejemplo positivo y el ejemplo negativo más difíciles para la red.

$$\mathcal{L} = \left[\text{máx}\left(\vec{D_{a,p}}\right) - \text{mín}\left(\vec{D_{a,n}}\right) + m\right]_+$$

donde $\vec{D_{a,p}} = (D^1_{a,p}, D^2_{a,p}, ..., D^N_{a,p})$ son las distancias euclídeas entre los descriptores de cada par de imágenes ancla y positiva del batch,

- **Lifted Embedding Loss:** Esta función de pérdida, descrita por Hermans *et al.* [15], se caracteriza porque además de minimizar la distancia entre la imagen ancla y la imagen positiva y maximizar la distancia entre la imagen ancla y la imagen negativa, también tiene en cuenta la distancia entre la imagen positiva y la imagen negativa, tratando de maximizar esta distancia. Se puede definir con la siguiente expresión:

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}\left[D^i_{a,p} + \ln\left(e^{m-D^i_{a,n}} + e^{m-D^i_{p,n}}\right)\right]_+$$

donde $D^i_{a,p}$ es la distancia euclídea entre los descriptores de la imagen ancla y la imagen positiva, $D^i_{a,n}$ es la distancia euclídea entre los descriptores de la imagen ancla y la imagen negativa, $D^i_{p,n}$ es la distancia euclídea entre los descriptores de la imagen positiva y la imagen negativa , $[...]_+$ es la función relu,m es el margen y N es el tamaño del batch.

- **Circle Loss:** Esta función de pérdida, empleada por Sun *et al.* [34], trata de maximizar la similitud coseno entre la imagen ancla y la imagen positiva, al mismo tiempo que trata de minimizar la similitud coseno entre la imagen ancla y la imagen negativa. Se puede definir con la siguiente expresión:

$$\mathcal{L} = \ln\left(1 + \sum_{j=1}^{N} e^{\gamma\alpha^j_n s^j_n} + \sum_{i=1}^{N} e^{-\gamma\alpha^i_p s^i_p}\right)$$

$$\alpha^i_p = \left[O_p - s^i_p\right]_+$$

$$\alpha^j_n = \left[s^j_n - O_n\right]_+$$

$$O_p = 1 - m$$

$$O_n = m$$

donde s^i_p es la similitud coseno entre los descriptores de la imagen ancla y la imagen positiva, s^j_n es la similitud coseno entre los descriptores de la imagen ancla y la imagen negativa, $[...]_+$ es la función relu, γ es un factor de escala, m es el margen y N es el tamaño del batch.

- **Angular Loss:** Esta función de pérdida, utilizada por Wang *et al.* [39], trata de minimizar el ángulo formado por el vector que une la imagen ancla y la imagen negativa y el vector que une la imagen positiva y la imagen negativa. De esta forma, minimiza la distancia entre la imagen ancla y la imagen positiva:

$$\mathcal{L} = \ln\left(1 + \sum_{i=1}^{N} e^{f^i_{a,p,n}}\right)$$

$$f^i_{a,p,n} = 4\tan^2\alpha\left(x^i_a + x^i_p\right)^T x^i_n - 2\left(1 + tan^2\alpha\right)\left(x^i_a\right)^T x^i_p$$

donde x^i_a es el descriptor de la imagen ancla, x^i_p es el descriptor de la imagen positiva, x^i_n es el descriptor de la imagen negativa, α es un margen angular y N es el tamaño del batch.

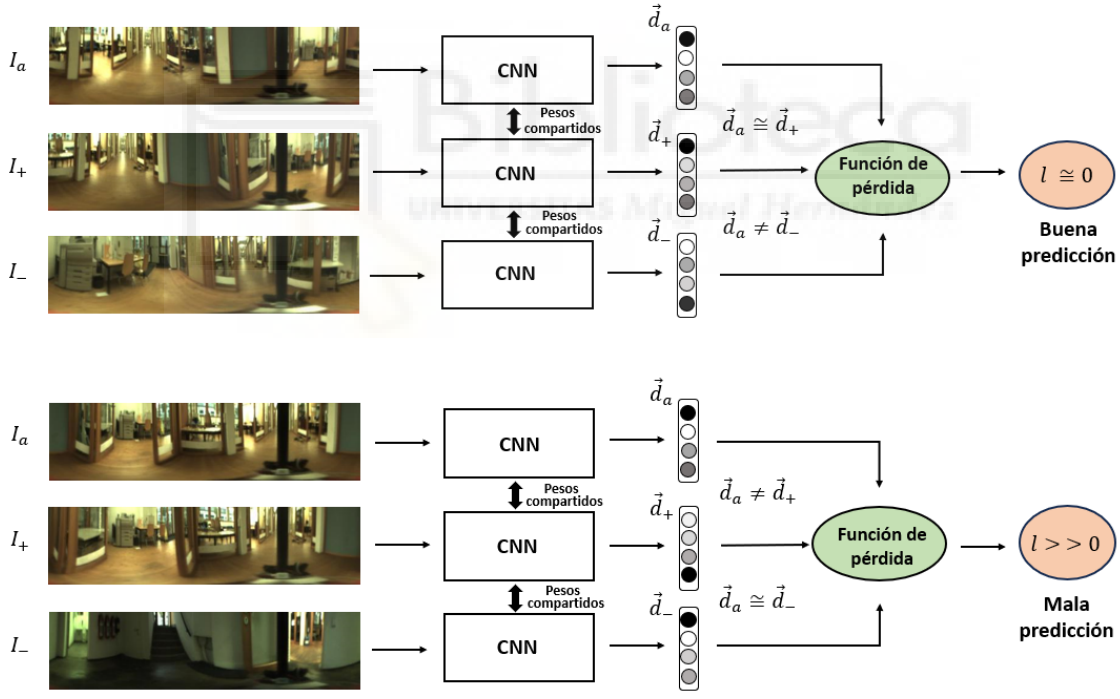


Figura 3-10: Ejemplo de una predicción correcta y otra errónea de la red.

4 LOCALIZACIÓN MEDIANTE REDES NEURONALES TRIPLETAS

4.1. INTRODUCCIÓN A LA TAREA DE LOCALIZACIÓN

En este trabajo, la localización de robots móviles se ha realizado mediante dos métodos distintos:

- **Localización jerárquica:** Este método consiste en determinar la posición en la que el robot ha capturado una imagen en dos pasos. En primer lugar, se realiza la localización gruesa, que consiste en determinar en qué estancia se ha tomado la imagen. En segundo lugar, se realiza la localización fina, cuyo objetivo es determinar las coordenadas exactas en las que el robot ha capturado la imagen dentro de dicha estancia.

- **Localización global:** Este método consiste en determinar las coordenadas exactas en las que se ha capturado una imagen sin conocer previamente en qué estancia se encuentra el robot. Este método permite realizar la localización del robot en un único paso, pero requiere un mayor coste computacional.

Los dos métodos se han llevado a cabo mediante redes neuronales tripletas adaptadas para esta tarea, modificando los procesos de entrenamiento, validación y test para cada método.

4.2. ADAPTACIÓN DE LAS REDES SIMPLES A LAS REDES TRIPLETAS

Las Redes Neuronales Tripletas están compuestas por tres CNNs que funcionan en paralelo. Las tres subredes comparten sus pesos internos, pero cada una de ellas

recibe distintas imágenes de entrada y, por lo tanto, devuelve distintos descriptores de salida. Las redes tripletas reciben combinaciones de tres imágenes (ancla, positiva, negativa) y su aprendizaje se basa en minimizar la distancia entre las imágenes ancla y positiva y la diferencia de las imágenes ancla y negativa. La función de pérdida determinará el error cometido en la predicción de la red, y el optimizador actualizará los pesos de la red en función del error cometido. En la Figura **4-1** se muestra un esquema del proceso de aprendizaje de las Redes Tripletas.

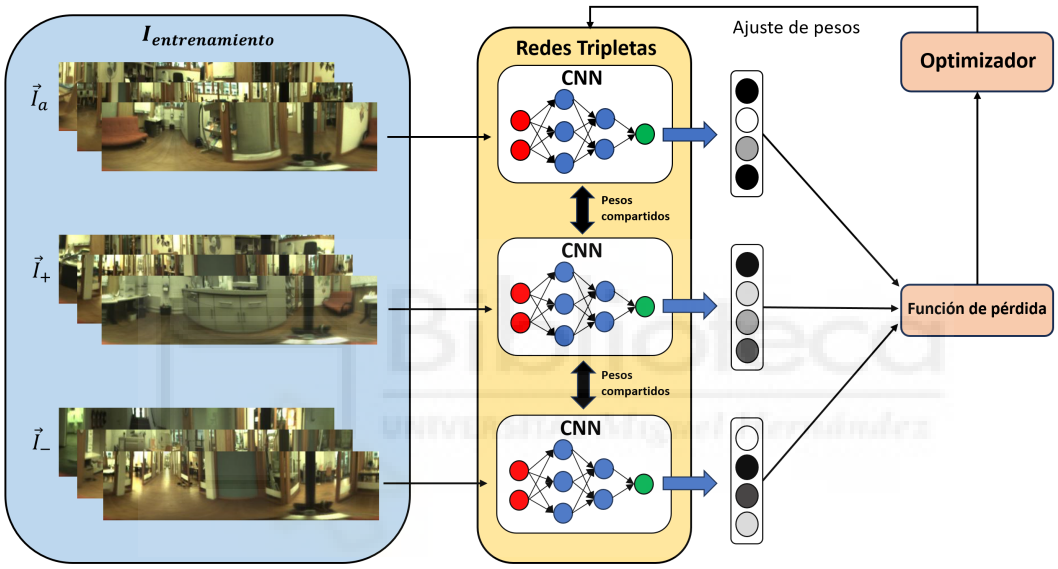


Figura 4-1: Entrenamiento de las Redes Neuronales Tripletas.

La arquitectura interna de las redes empleadas será la del modelo de red VGG16 adaptado a la tarea a desarrollar, ya que ha proporcionado buenos resultados en trabajos similares [7]. Las capas convolucionales permanecen intactas y se modifican las capas totalmente conectadas para obtener como salida un vector descriptor de 5 elementos. Se emplea la técnica del *Transfer Learning* únicamente en las capas que permanecen intactas. De esta manera, se parte de los pesos del modelo VGG16 en las capas convolucionales con el fin de aprovechar los conocimientos adquiridos en la fase de extracción de las características. En cuanto a las capas totalmente conectadas, se parte de pesos totalmente aleatorios en cada uno de los entrenamientos. En la Figura **4-2** se muestra la arquitectura interna de la red modificada. Por simplicidad no se han mostrado las capas ReLU.

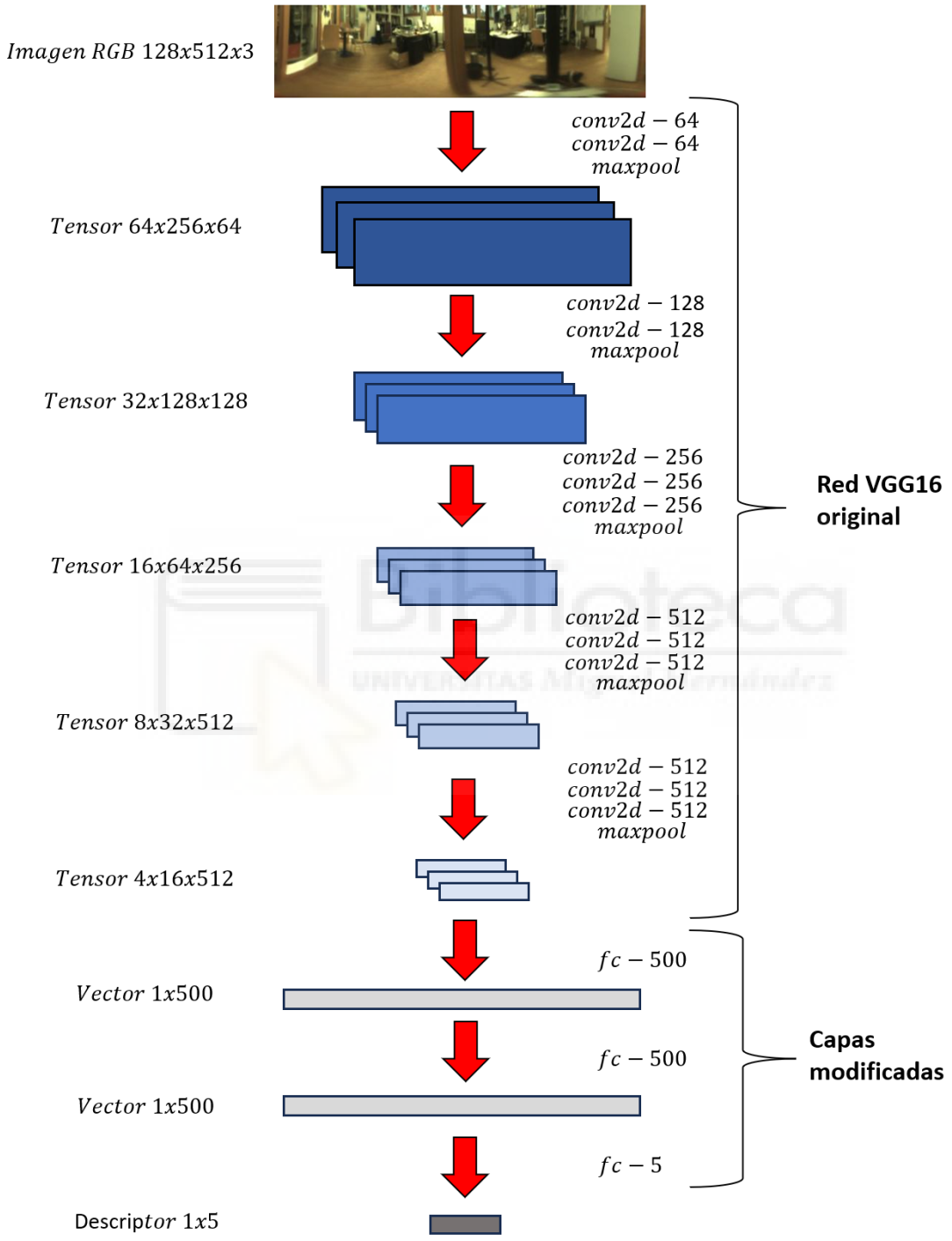


Figura 4-2: Arquitectura interna de la red VGG16 adaptada a la tarea de localización.

4.3. LOCALIZACIÓN JERÁRQUICA MEDIANTE REDES TRIPLETAS

4.3.1. LOCALIZACIÓN GRUESA

El objetivo principal de la localización gruesa es predecir la estancia en la que se ha capturado una imagen. Para ello, se realiza el entrenamiento de la red con combinaciones de tres imágenes: una imagen ancla, una imagen positiva y una imagen negativa. La imagen ancla y la imagen positiva deben pertenecer a la misma habitación, mientras que la imagen negativa debe pertenecer a una habitación distinta (Figura **4-3**).

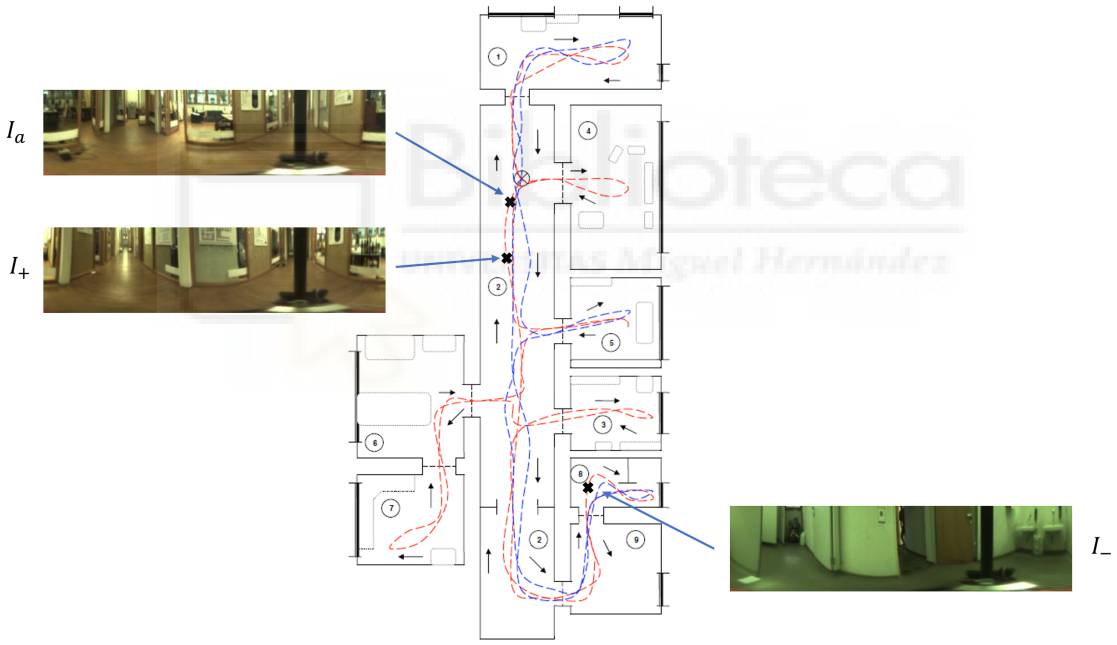


Figura 4-3: Combinación de tres imágenes ancla (I_a), positiva (I_+) y negativa (I_-) para el entrenamiento de la red en la localización gruesa.

Para el test de la red, se obtiene una imagen representativa por cada estancia del edificio, que se trata de la imagen más cercana al centro geométrico de la estancia. Cada imagen del conjunto de test se comparará con las 9 imágenes representativas obtenidas mediante la distancia euclídea o la similitud coseno entre sus descriptores (Figura **4-4**). El descriptor de la imagen representativa más similar al descriptor de la imagen de test permitirá predecir la habitación en la que se encuentra el robot. Si

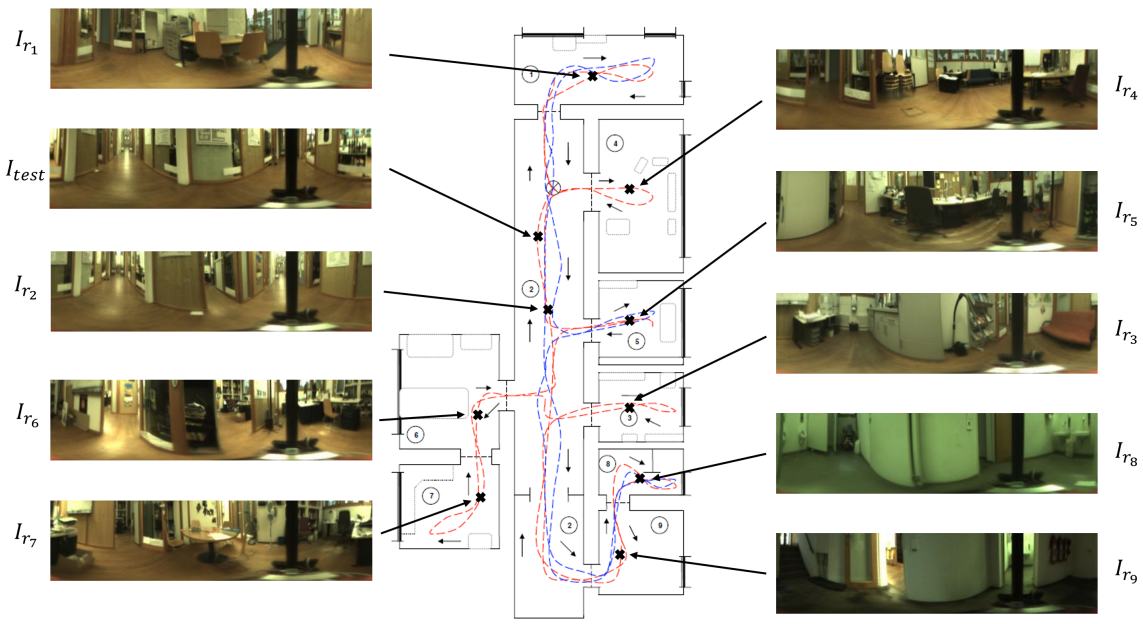


Figura 4-4: Imágenes representativas de cada habitación para realizar el test en la localización gruesa.

esta coincide con la habitación real de la imagen de test, se considerará un acierto de la red.

4.3.2. LOCALIZACIÓN FINA

La localización fina tiene por objeto determinar las coordenadas exactas en las que se ha capturado una imagen, conociendo previamente la estancia en la que se encuentra el robot. En este trabajo se ha propuesto entrenar una red para cada una de las habitaciones por separado, partiendo de los pesos de la red guardada en la localización gruesa.

Para ello, se ha realizado un entrenamiento de la red con combinaciones de tres imágenes: una imagen ancla, una imagen positiva y una imagen negativa. En este caso, las tres imágenes pertenecen a la misma habitación, pero la imagen positiva y la imagen ancla deben haber sido capturadas a una distancia menor que una distancia umbral r, mientras que la imagen negativa debe haber sido capturada a una distancia mayor que r de la imagen ancla. En este trabajo se realizará un estudio del efecto de la elección de la función de pérdida en el desempeño de la red, así como de la variación de la distancia umbral r.

El test de la red se ha realizado para cada estancia por separado, mediante la técni-

ca de los K vecinos más cercanos. Se ha obtenido el descriptor de todas las imágenes y, para cada imagen de test, se ha comparado su descriptor con los descriptores de las imágenes de su misma estancia del conjunto de datos de entrenamiento, utilizado como modelo visual, mediante la distancia euclídea o la similitud coseno (Figura **4-5**). La menor distancia indicará la imagen más cercana en el espacio del descriptor. Si la imagen predicha está dentro de las K imágenes más cercanas a la imagen de test, se considerará un acierto de la red. Para el test, se utilizarán los valores K = 1, 2, 4 y 8. La validación se realizará aplicando el mismo método con K=1, es decir, si la imagen predicha coincide con la imagen más cercana se considera un acierto de la red.

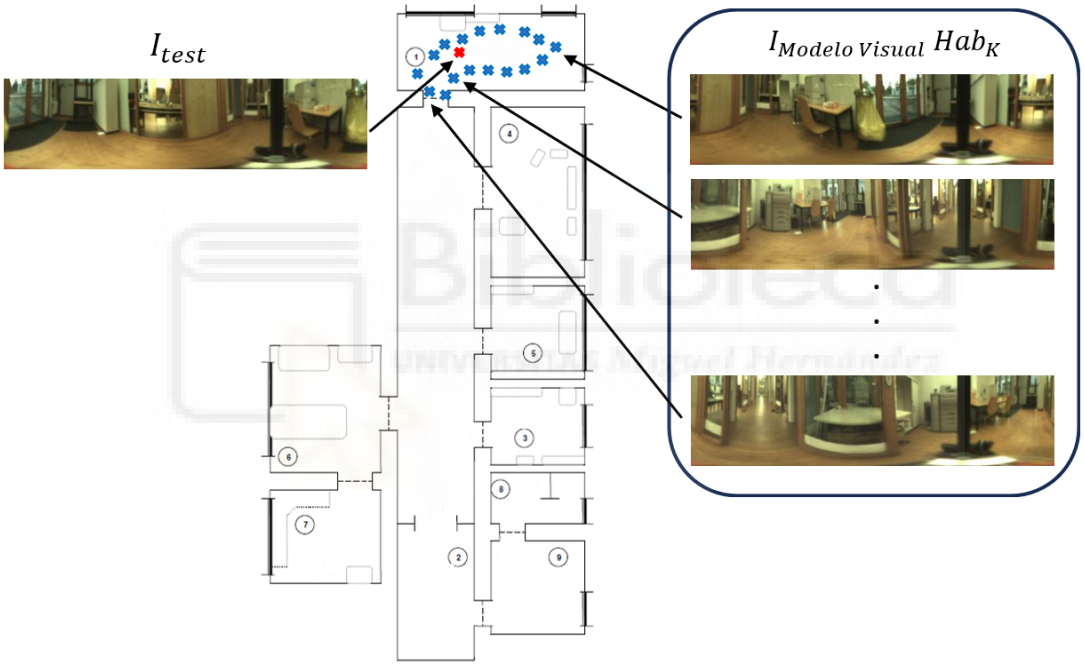


Figura 4-5: Comparación de una imagen I_{test} con las imágenes del modelo visual de la habitación predicha por la red en el test de la localización fina.

En la Figura **4-6** se muestra un esquema del proceso de localización jerárquica, dividido en dos etapas: la localización gruesa y la localización fina. En la localización gruesa, el descriptor de la imagen I_{test} se compara con los descriptores de las imágenes representativas $I_{representativas} = [I_{r_1}, I_{r_2}, ..., I_{r_9}]$. La menor distancia entre descriptores permitirá identificar la habitación en la que se encuentra el robot. En la localización fina, el descriptor de la imagen I_{test} se compara con los descriptores de las imágenes del modelo visual de la habitación predicha $I_{MV} = [I_{MV_1}, I_{MV_2}, ..., I_{MV_n}]$. La menor distancia entre descriptores indicará las coordenadas en las que se encuentra el robot.

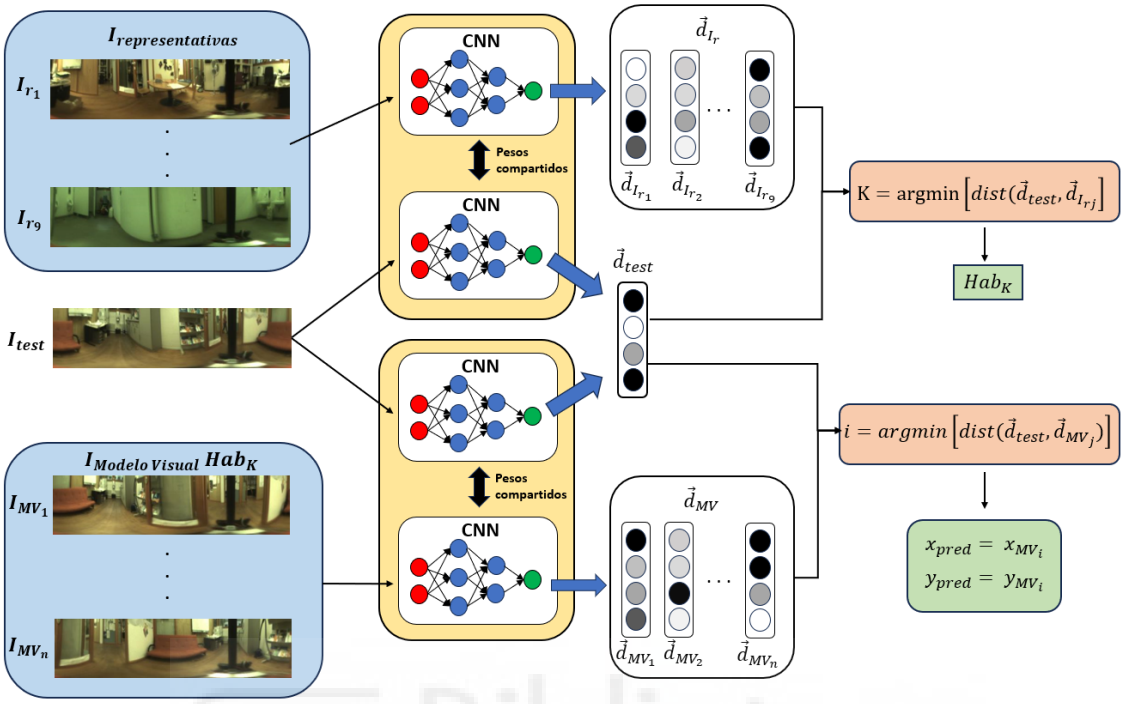


Figura 4-6: Proceso de test en la localización jerárquica.

4.4. LOCALIZACIÓN GLOBAL MEDIANTE REDES TRIPLETAS

La localización global trata de determinar las coordenadas exactas en las que se ha capturado una imagen sin predecir previamente a qué estancia pertenece. Para ello, se realiza un entrenamiento de la red utilizando combinaciones de tres imágenes. La imagen ancla y la imagen positiva deben estar a una distancia menor que una distancia umbral r, mientras que la imagen negativa debe estar a una distancia mayor que r de la imagen ancla. En este caso, no se incluye ninguna restricción relativa a la estancia en la que se han capturado las imágenes.

El test de la red se ha realizado con todas las imágenes de cada conjunto de test, sin tener en cuenta la habitación a la que pertenecen las imágenes. El test se ha llevado a cabo mediante la técnica de los K vecinos más cercanos. Para cada imagen de test, se ha obtenido su descriptor y se ha comparado con el descriptor de todas las imágenes que conforman el modelo visual del edificio (Figura **4-7**). La imagen

del modelo visual que devuelva una menor distancia euclídea o una mayor similitud coseno indicará la imagen más cercana predicha por la red. Si esta se encuentra dentro de las K imágenes más cercanas a la imagen de test, se considera un acierto de la red. Para el test, se utilizarán los valores K = 1, 2, 4 y 8, mientras que la validación de la red se realizará únicamente con K=1.

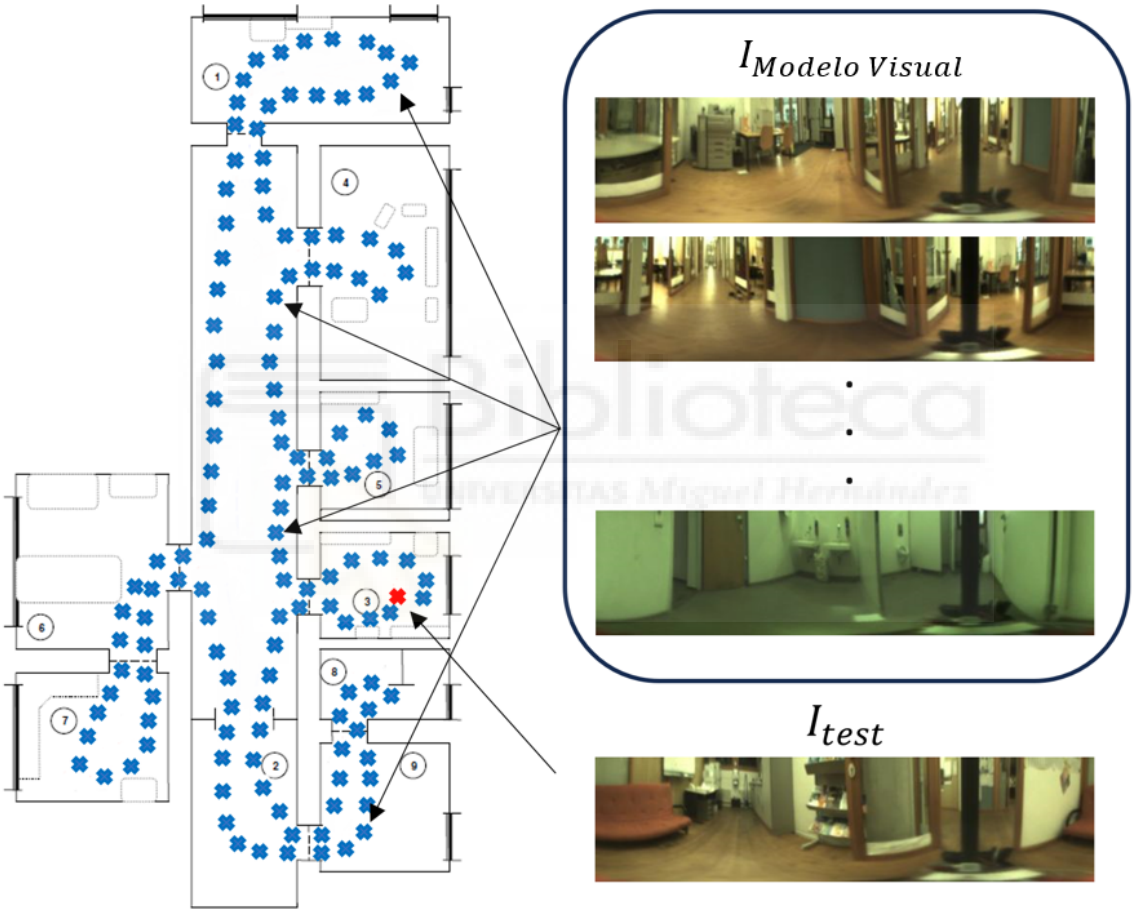


Figura 4-7: Comparación de una imagen I_{test} con las imágenes del modelo visual de todo el edificio en el test de la localización global.

En la Figura **4-8** se muestra un esquema detallado del procedimiento seguido para el test en la localización global. El descriptor de la imagen I_{test} se compara con los descriptores de las imágenes $I_{MV} = [I_{MV_1}, I_{MV_2}, ..., I_{MV_n}]$. La menor distancia entre descriptores indicará las coordenadas en las que se encuentra el robot.

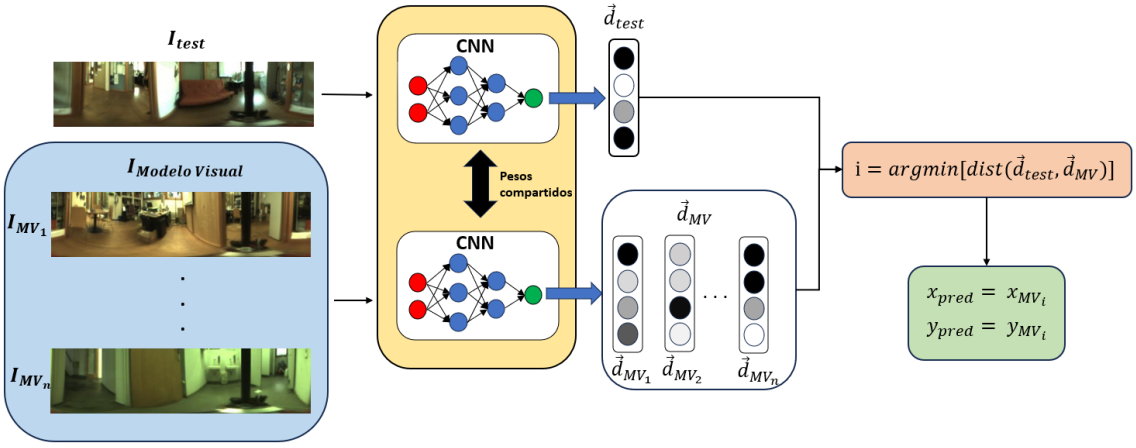


Figura 4-8: Proceso de test en la localización global.

Biblioteca
UNIVERSITAS Miguel Hernández

5 EXPERIMENTOS Y RESULTADOS

En este apartado se describirán los experimentos realizados y los resultados obtenidos para cada uno de ellos. Los experimentos se pueden dividir en dos bloques: la localización jerárquica y la localización global. Durante la realización de los experimentos, únicamente se modificará o bien la función de pérdida y los parámetros de ésta, o bien el criterio de selección de las imágenes de entrenamiento, ya que el estudio de la red escogida y la modificación de su arquitectura interna no es objeto del presente trabajo. A continuación, se describirán los conjuntos de entrenamiento, validación y testeo de la red.

5.1. CONJUNTOS DE DATOS DE ENTRENAMIENTO, VALIDACIÓN Y TEST DE LA RED

En la Tabla **5-1** se muestra el número de imágenes por cada una de las estancias del edificio para cada conjunto de datos. Se ha utilizado el mismo conjunto de entrenamiento para todos los experimentos, que contiene 556 imágenes tomadas únicamente bajo condición de iluminación nublado. El conjunto de entrenamiento se ha utilizado para generar las combinaciones de tres imágenes con las que se ha entrenado la red y también como modelo visual para la validación y el test de las redes entrenadas para las tareas de localización fina y localización global. El conjunto de validación

Estancia	**1PO-A**	**2PO1-A**	**2PO2-A**	**CR-A**	**KT-A**	**LO-A**	**PA-A**	**ST-A**	**TL-A**	**Total**
Entrenamiento	44	46	31	238	46	26	57	30	38	556
Validación	43	47	32	236	46	26	57	31	38	556
Test nublado	155	230	135	1040	254	177	222	133	249	2595
Test noche	168	215	168	1114	270	121	241	198	212	2707
Test soleado	123	187	109	793	213	102	191	180	216	2114
Modelo visual	44	46	31	238	46	26	57	30	38	556

Tabla 5-1: Tabla resumen de los conjuntos de datos de entrenamiento, validación y test.

tiene la misma cantidad de imágenes que el conjunto de entrenamiento, así como la misma proporción de imágenes por estancia y la misma condición de iluminación. El test de las redes entrenadas se ha realizado para cada condición de iluminación por separado, con el objetivo de evaluar la robustez de la red frente a cambios lumínicos. Los conjuntos de test nublado, noche y soleado están compuestos por 2595, 2707 y 2114 imágenes, respectivamente. Los conjuntos de entrenamiento, validación y test no poseen ninguna imagen en común, es decir, la validación y el test de la red se realizan con imágenes que la red no ha visto durante el entrenamiento.

5.2. LOCALIZACIÓN JERÁRQUICA

La localización jerárquica consiste en estimar la posición del robot en la que se ha tomado una imagen en dos etapas. En primer lugar, se realiza la localización gruesa, en la cual se determina en qué estancia se ha capturado la imagen, y en segundo lugar se realiza la localización fina, que consiste en obtener la posición exacta de la imagen conociendo previamente en qué habitación se encuentra el robot.

5.2.1. LOCALIZACIÓN GRUESA

En este apartado se ha realizado la evaluación del desempeño de las Redes Neuronales Tripletas para la tarea de identificación de la estancia en la que se han capturado las imágenes. Para ello, se ha comparado cada imagen de cada conjunto de test con la imagen representativa de cada habitación, obtenida como la imagen más cercana al centro geométrico de todas las imágenes pertenecientes a dicha habitación. La imagen representativa cuyo descriptor devuelve la menor distancia euclídea o la mayor similitud coseno con el correspondiente a la imagen de test indicará la habitación en la que se encuentra el robot. Si esta coincide con la habitación real, se considera un acierto de la red.

Para realizar la localización gruesa, se ha llevado a cabo un estudio exhaustivo de cómo influye la elección de la función de pérdida en el desempeño de la red.

Estudio de la función de pérdida de triplete escogida

Como se ha comentado en el apartado 3.6, la selección de una función de pérdida adecuada para la tarea a realizar es un factor clave en el desempeño de la red. Además, cada función de pérdida lleva asociados una serie de parámetros que se deben ajustar. Es por ello que se ha realizado un estudio de diferentes funciones de pérdida de triplete, así como de los valores que toman sus parámetros.

Las redes tripletas han sido entrenadas con combinaciones de tres imágenes seleccionadas de forma aleatoria, con la única restricción de que las imágenes ancla y positiva deben pertenecer a la misma estancia y la imagen negativa debe pertenecer a una estancia diferente. Se ha realizado un entrenamiento de tres épocas y en cada una de ellas se ha empleado un total de 50000 combinaciones de tripletas de imágenes.

En la Tabla **5-2** se muestran los resultados obtenidos para cada función de pérdida y para cada condición de iluminación, así como los parámetros que han devuelto mejores resultados:

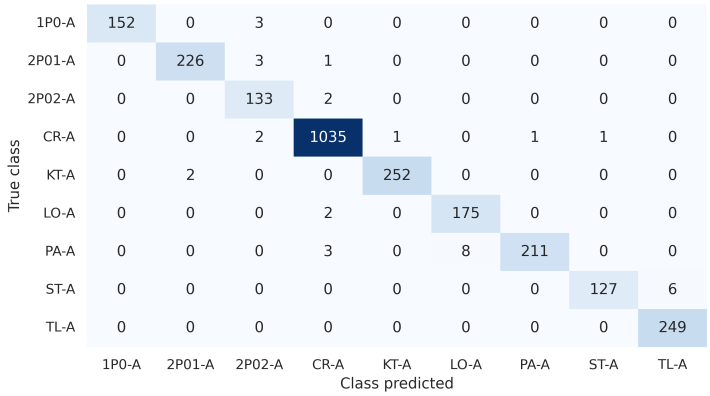
Función de pérdida	**Parámetros óptimos**	**Precisión test (%)**		
		Nublado	**Noche**	**Soleado**
Triplet margin loss	m=1	98,96	96,97	91,82
Lifted embedding loss	m=0	97,73	96,93	91,82
Circle loss	γ=1, m=0	97,38	97,23	91,20
Lazy triplet loss	m=1	98,65	97,16	92,53
Semi hard loss	m=0,5	98,69	97,27	91,06
Batch hard loss	m=0,5	99,19	97,41	89,12
Angular loss	α=20	84,55	82,67	79,99

Tabla 5-2: Resultados de la localización gruesa para cada función de pérdida en las 3 condiciones de iluminación.

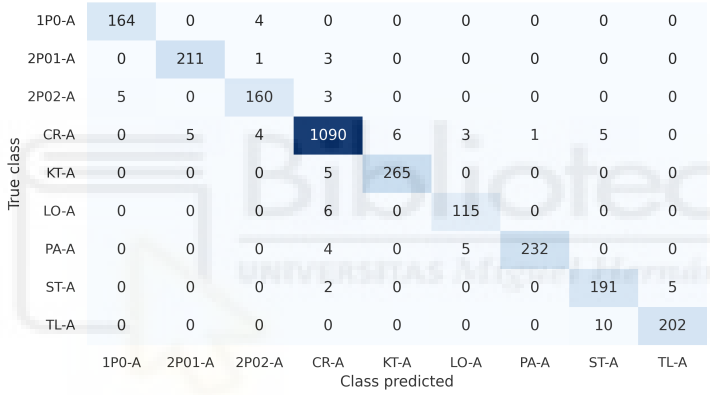
La función de pérdida que, en líneas generales, ha proporcionado mejores resultados ha sido la *Lazy Triplet Loss*. La red entrenada con la función de pérdida *Triplet Margin Loss* también ha tenido un desempeño ligeramente superior al resto. Exceptuando la *Angular Loss*, que ha tenido un rendimiento inferior en esta etapa, los resultados son muy similares para todas las funciones de pérdida. Además, los resultados son bastante elevados para las tres condiciones de iluminación, incluso en soleado, que es la iluminación que provoca un mayor cambio en la apariencia de las escenas respecto a nublado, que es la condición usada para el entrenamiento.

La matriz de confusión es una herramienta que nos permite ver las predicciones realizadas por la red, así como los aciertos y errores cometidos para cada habitación. En la Figura **5-1** se muestran las matrices de confusión obtenidas en el test de la red que ha proporcionado mejores resultados en la localización gruesa.

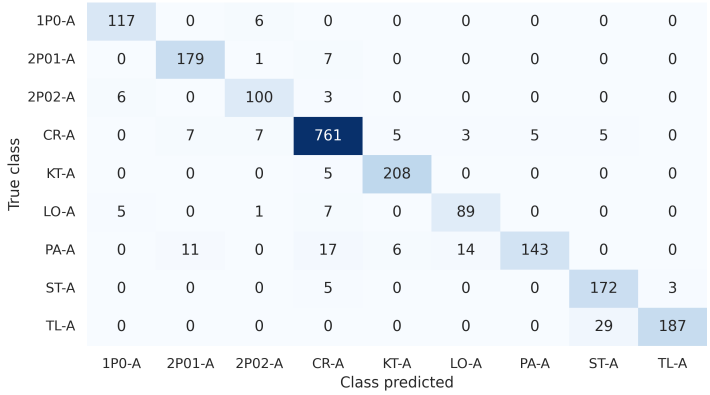
A partir de las matrices de confusión, se puede observar que la mayoría de predic-



(a)



(b)



(c)

Figura 5-1: Matrices de confusión obtenidas en el test de la red entrenada con la función de pérdida *Lazy Triplet Loss* para a) Nublado, b) Noche y c) Soleado

.

ciones erróneas se han producido en el pasillo (CR-A). Esto se debe a que presenta mayores dimensiones y por tanto se han tomado más imágenes, y a que está conectado a un mayor número de habitaciones. Además, las predicciones erróneas han sido casi en su totalidad entre habitaciones conectadas debido a que comparten parte de la información visual en las zonas de transición. Si únicamente se tienen en cuenta los errores entre habitaciones no conectadas, se alcanzaría una precisión del 99'50 % para nublado, 99'82 % para noche y 98'25 % para soleado para la función de pérdida *Lazy Triplet Loss*. Además, se puede observar que hay un número elevado de errores entre la zona de escaleras (ST-A) y los aseos (TL-A), así como entre la sala de la impresora (PA-A) y la oficina grande (LO-A). Esto se debe a que la red no ha sido entrenada de forma óptima para estas habitaciones, ya que en el conjunto de entrenamiento hay pocas imágenes de las estancias ST-A y LO-A, produciéndose un sesgo en los datos.

5.2.2. LOCALIZACIÓN FINA

En este apartado se evaluarán las Redes Neuronales Tripletas para la identificación de las coordenadas en las que se ha capturado una imagen, conociendo previamente la estancia en la que encuentra el robot. Para ello, se ha comparado cada imagen de test con todas las imágenes de su misma habitación del conjunto de entrenamiento, utilizado como modelo visual. La imagen cuyo descriptor devuelva la menor distancia euclídea o mayor similitud coseno será la imagen más cercana a la imagen de test predicha por la red. Si la imagen predicha está dentro de las K imágenes más cercanas, se considera un acierto de la red. Para abordar el problema de localización fina, se ha realizado el entrenamiento de una red por cada una de las habitaciones por separado, partiendo de los pesos de la red guardada en la localización gruesa.

Las redes tripletas han sido entrenadas con combinaciones de tres imágenes seleccionadas de forma totalmente aleatoria, con las únicas restricciones de que las tres imágenes deben pertenecer a la misma estancia, la imagen positiva debe haber sido capturada a una distancia menor que una distancia umbral r de la imagen ancla, mientras que la imagen negativa debe haber sido capturada a una distancia mayor que r de la imagen ancla. Para cada estancia, se ha realizado un entrenamiento de cinco épocas y en cada una de ellas se ha empleado un total de 5000 combinaciones de tripletas de imágenes.

En este apartado, se ha estudiado cómo influyen la selección de la función de pérdida y la variación de la distancia r en el desempeño de la red.

Estudio de la función de pérdida de triplete escogida

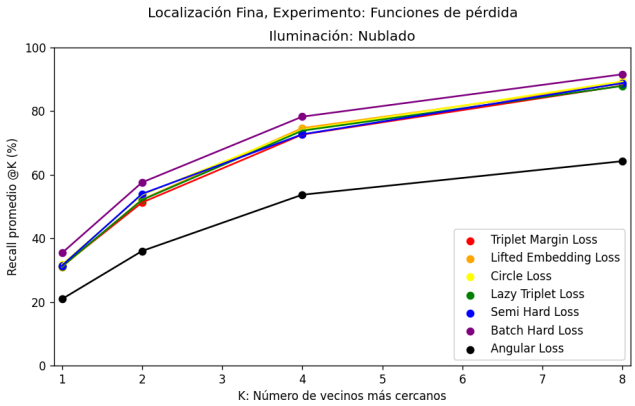
En este estudio se utilizarán las funciones de pérdida con los parámetros que han dado mejores resultados en la localización gruesa.

En la Figura **5-2** y en la Tabla **5-3** se muestran el acierto para K vecinos más cercanos y el error en metros cometido por la red, respectivamente, para cada función de pérdida y para cada condición de iluminación por separado. El recall se define como el porcentaje de predicciones correctas para cada valor de K. En el eje X se representan el número de vecinos más cercanos, mientras que en el eje Y se representa el recall medio para ese número de vecinos más cercanos. En la leyenda se muestran las funciones de pérdida para las cuales se ha realizado el experimento.

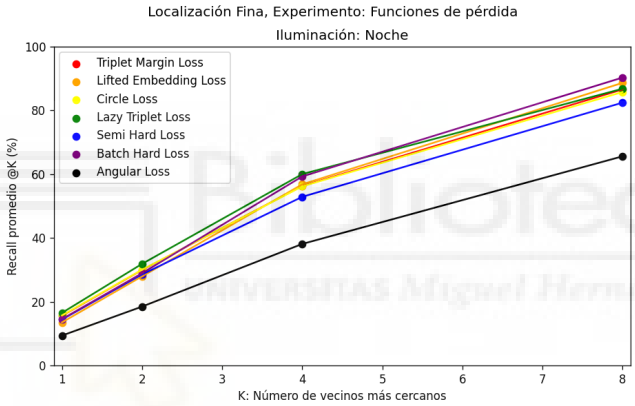
Función de pérdida	**Error distancia(m)**		
	Cloudy (Error mínimo: 0.128m)	**Cloudy (Error mínimo: 0.127m)**	**Cloudy (Error mínimo: 0.121m)**
Triplet margin loss(m=1)	0,251	0,287	0,478
Lifted embedding(m=0)	0,252	0,285	0,700
Circle loss(γ=1,m=0)	0,260	0,298	0,607
Lazy triplet(m=1)	0,223	0,277	0,441
Semi hard(m=0.5)	0,261	0,329	0,540
Batch hard(m=0.5)	0,228	0,276	0,524
Angular loss(α=20)	1,115	0,894	1,613

Tabla 5-3: Error medio cometido en metros en la localización fina para cada función de pérdida en las 3 condiciones de iluminación.

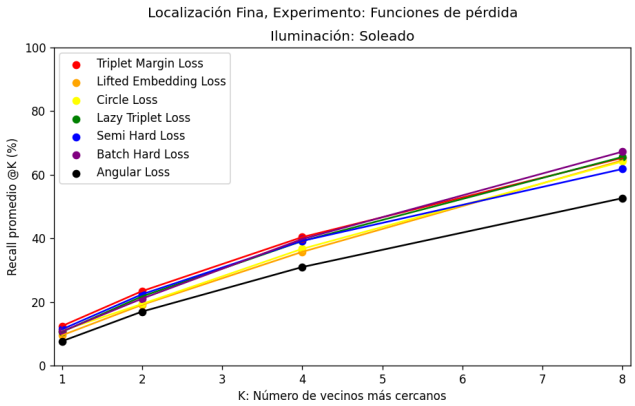
A partir de las gráficas mostradas en la Figura **5-2** y de la Tabla **5-3**, se puede comprobar que, en líneas generales, los errores cometidos son pequeños para las tres condiciones de iluminación, especialmente en nublado y noche. La función de pérdida *Lazy Triplet Loss* y sus variantes, la *Semi Hard Loss* y la *Batch Hard Loss*, son las que proporcionan mejores resultados para la etapa de la localización fina, así como la *Triplet Margin Loss*. De la misma forma que sucedía en la localización gruesa, la función de pérdida *Angular Loss* es la que ha tenido peores resultados, mientras que el resto ha mostrado un rendimiento similar.



(a)



(b)



(c)

Figura 5-2: Recall para K = 1, 2, 4 y 8 en la localización fina para cada función de pérdida en a) Nublado, b) Noche y c) Soleado.

Estudio de la variación de la distancia umbral r

En este apartado se utilizará la función de pérdida *Lazy Triplet Loss* con m=1 para el entrenamiento de la red, ya que ha sido la función de pérdida que ha proporcionado mejores resultados tanto en la localización gruesa como en la localización fina. Se ha partido de una distancia umbral de 0'3 metros, puesto que se trata de la distancia mínima para la cual todas las imágenes del conjunto de datos tienen como mínimo una imagen positiva, y se ha analizado como afecta al desempeño de la red el aumento de la distancia r.

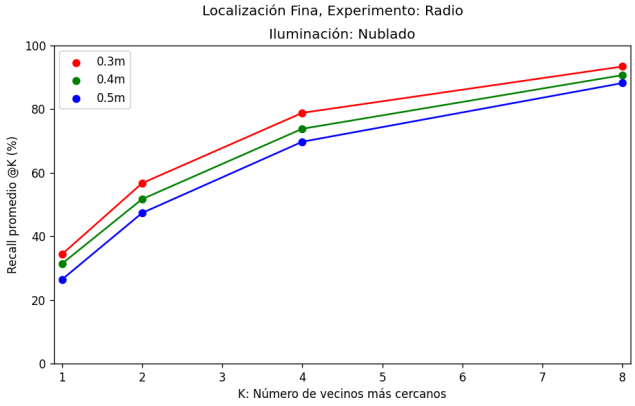
En la Figura **5-3** y en la Tabla **5-4** se muestran el acierto para K vecinos más cercanos y el error medio cometido en metros, respectivamente, con la función de pérdida *Lazy Triplet Loss* para cada valor de r y para cada condición de iluminación por separado. En el eje X se representan el número de vecinos más cercanos, mientras que en el eje Y se representa el recall medio para ese número de vecinos más cercanos. En la leyenda se muestran los valores de r utilizados.

r(m)	**Error distancia(m)**		
	Cloudy (Error mínimo: 0,128m)	**Night (Error mínimo: 0,127m)**	**Sunny (Error mínimo: 0,121m)**
0,3	0,223	0,277	0,441
0,4	0,238	0,295	0,538
0,5	0,260	0,296	0,589

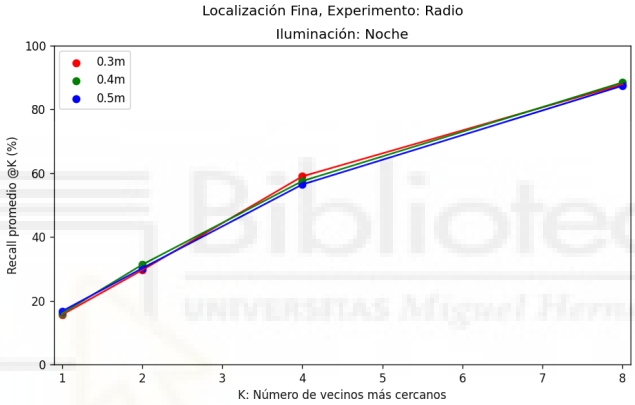
Tabla 5-4: Error medio cometido en metros en la localización fina para cada valor de r en las 3 condiciones de iluminación.

A partir de las gráficas mostradas en la Figura **5-3** y de la Tabla **5-4**, se puede comprobar que, al aumentar la distancia r, el error aumenta en las tres condiciones de iluminación. Esto se debe a que al aumentar r, cada imagen ancla tendrá más imágenes positivas y menos imágenes negativas, por lo que el entrenamiento es menos restrictivo y por tanto se producen errores mayores.

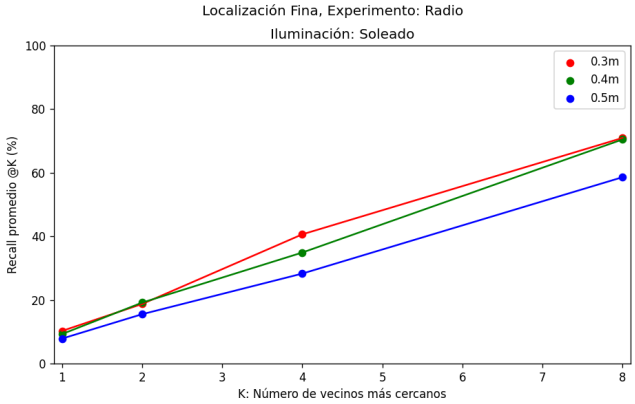
En la Tabla **5-5** se muestra el error en metros cometido para cada habitación con la función de pérdida y el valor de r que han proporcionado los mejores resultados en los experimentos anteriores.



(a)



(b)



(c)

Figura 5-3: Recall para K = 1, 2, 4 y 8 en la localización fina con la función de pérdida *Lazy Triplet Loss* para cada valor de r en a) Nublado, b) Noche y c) Soleado.

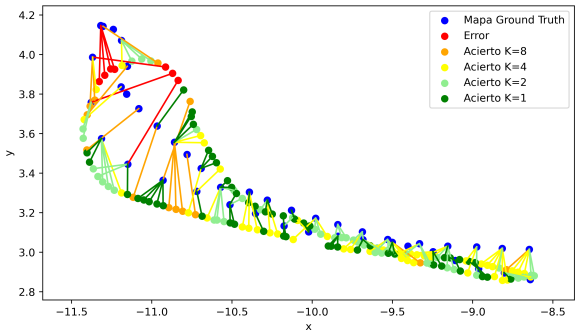
Estancia	Cloudy		Night		Sunny	
	Error(m)	Error mínimo(m)	Error(m)	Error mínimo(m)	Error(m)	Error mínimo(m)
1PO-A	0,147	0,087	0,243	0,128	0,321	0,103
2PO1-A	0,120	0,068	0,215	0,067	0,373	0,088
2PO2-A	0,102	0,063	0,283	0,169	0,338	0,084
CR-A	0,246	0,121	0,342	0,146	0,641	0,125
KT-A	0,250	0,148	0,222	0,106	0,288	0,164
LO-A	0,325	0,232	0,337	0,208	0,383	0,168
PA-A	0,322	0,201	0,222	0,095	0,307	0,131
ST-A	0,117	0,074	0,225	0,119	0,259	0,106
TL-A	0,197	0,146	0,171	0,073	0,338	0,105
Medio	**0,223**	**0,128**	**0,277**	**0,127**	**0,441**	**0,121**

Tabla 5-5: Error medio cometido en metros en la localización fina para la función de pérdida *Lazy Triplet Loss* con r=0'3 m en las 3 condiciones de iluminación y errores mínimos que se pueden cometer para cada habitación.

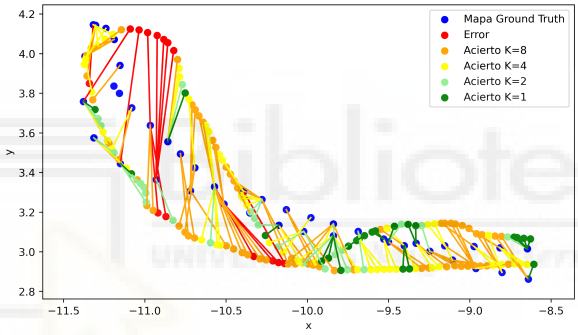
Biblioteca

Se puede comprobar que el error es muy variable dependiendo de la habitación para la que se realice la localización fina. Los mayores errores se obtienen en el pasillo (CR-A), ya que tiene unas mayores dimensiones que el resto y un mayor número de imágenes. Además, el error es muy dependiente de la diferencia entre las trayectorias seguidas por el robot en las secuencias de entrenamiento y test. Sin embargo, los errores cometidos en líneas generales son pequeños en comparación con las dimensiones de las estancias. El error nunca puede ser cero, ya que para ello las secuencias de entrenamiento y de test tendrían que ser iguales. El error mínimo es el que se cometería si se produjera un acierto del 100 % para K=1, esto es, la imagen predicha como la más cercana a la imagen de test coincide siempre con la imagen más cercana.

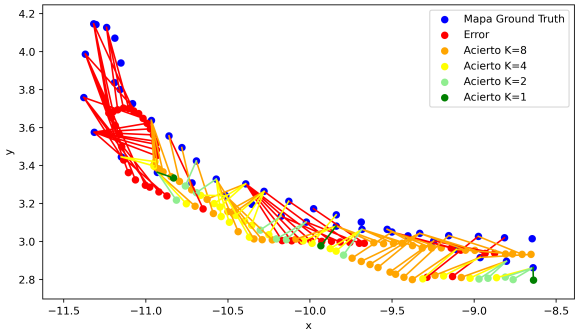
En la Figura **5-4** se muestra la trayectoria del robot en las secuencias de test y del modelo visual con las predicciones de la red para cada imagen de test en una de las habitaciones para las tres condiciones de iluminación. Los puntos azules conforman el modelo visual de la habitación, mientras que el resto representan las imágenes de test. Si se produce un acierto con K=1, el punto se representa de color verde oscuro, si se acierta con K=2, el punto tomará un color verde claro, y así sucesivamente hasta K=8. Si no se acierta para K=8, el punto se representa de color rojo. Las líneas unen la imagen de test con la imagen del modelo visual predicha como la más cercana.



(a)



(b)



(c)

Figura 5-4: Predicciones de la red para la localización fina con la función de pérdida *Lazy Triplet Loss* con r=0'3 m en la habitación 1PO-A para a) Nublado, b) Noche y c) Soleado.

Se puede observar que, en condiciones nubladas, prácticamente todas las imágenes son localizadas dentro de los 8 vecinos más cercanos, mientras que en condiciones nocturnas los resultados son menos precisos y en condiciones soleadas se producen los mayores errores. Sin embargo, se puede comprobar que generalmente no se producen errores de predicción excesivamente grandes. Como es lógico, la red entrenada tiene mayor dificultad para localizar las imágenes en aquellas zonas en las que las trayectorias seguidas en las secuencias de entrenamiento y de test son más diferentes.

5.3. LOCALIZACIÓN GLOBAL

La localización global consiste en determinar las coordenadas exactas en las que se ha capturado una imagen en una sola etapa. Para ello, se ha comparado cada imagen del conjunto de test con todas las imágenes del conjunto de entrenamiento, empleado como modelo visual. La imagen cuyo descriptor devuelva la menor distancia euclídea o mayor similitud coseno será la imagen más cercana predicha por la red. Si la imagen predicha está dentro de las K imágenes más cercanas, se considera un acierto de la red.

Las redes tripletas han sido entrenadas para esta tarea con combinaciones de tres imágenes escogidas aleatoriamente, con la única restricción de que la imagen ancla y la imagen positiva deben estar a una distancia menor que una distancia umbral r, mientras que la imagen negativa debe estar a una distancia mayor que r de la imagen ancla. Para este apartado, se ha utilizado una distancia r igual a 0'3 metros, ya que se ha demostrado previamente que es la distancia óptima para el entrenamiento de la red. Se ha realizado un entrenamiento de cinco épocas y en cada una de ellas se ha empleado un total de 50000 combinaciones de tripletas de imágenes.

En este apartado se ha estudiado cómo influye la selección de la función de pérdida en el desempeño de la red para esta tarea.

Estudio de la función de pérdida de triplete escogida

En este estudio se emplearán las funciones de pérdida utilizadas en la localización jerárquica con los parámetros que han dado mejores resultados en la localización gruesa.

En la Figura **5-5** y en la Tabla **5-6** se muestran el acierto para K vecinos más cercanos y el error en metros cometido por la red, respectivamente, para cada función de pérdida y para cada condición de iluminación por separado. En el eje X se repre-

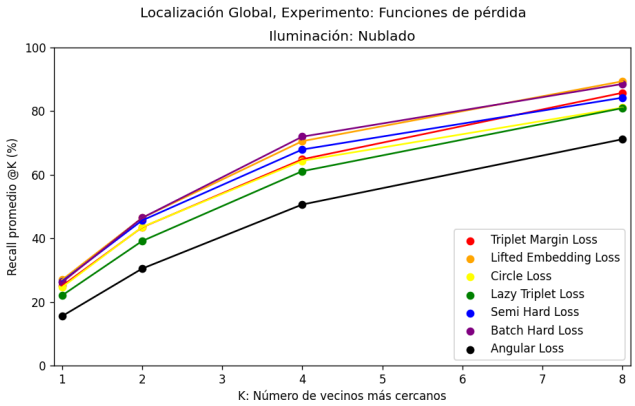
Función de pérdida	**Error distancia(m)**		
	Cloudy (Error mínimo: 0.128m)	**Night (Error mínimo: 0.127m)**	**Sunny (Error mínimo: 0.121m)**
Triplet margin loss(m=1)	0,303	0,324	0,633
Lifted embedding loss(m=0)	0,295	0,438	1,247
Circle loss(γ=1,m=0)	0,428	0,547	1,219
Lazy triplet loss(m=1)	0,361	0,350	1,412
Semi hard loss(m=0.5)	0,284	0,296	1,250
Batch hard loss(m=0.5)	0,346	0,274	2,362
Angular loss(α=20)	0,581	0,620	1,433

Tabla 5-6: Error medio cometido en metros en la localización global para cada función de pérdida en las 3 condiciones de iluminación.

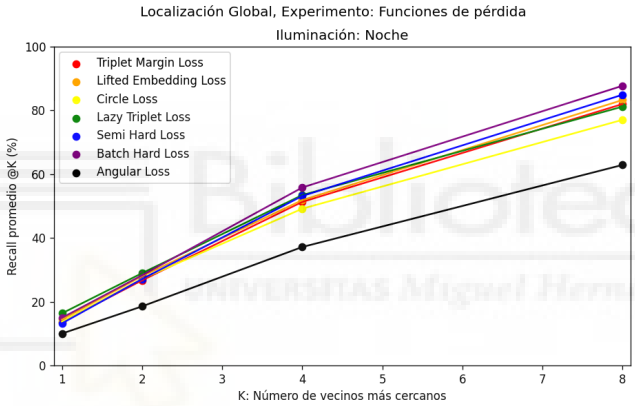
sentan el número de vecinos más cercanos, mientras que en el eje Y se representa el recall medio para ese número de vecinos más cercanos. En la leyenda se muestran las funciones de pérdida utilizadas.

A partir de las gráficas mostradas en la Figura **5-5** y de la Tabla **5-6**, se puede comprobar que, en líneas generales, los errores son mayores que en la localización jerárquica, especialmente en soleado. Esto es lógico, ya que en este caso se ha tratado de determinar las coordenadas en las que ha capturado una imagen dentro de todo el mapa. La función de pérdida que ha tenido el mejor resultado ha sido la *Triplet Margin Loss*, especialmente en soleado en comparación con el resto de funciones de pérdida. Esto se puede ver de forma más clara con la función de pérdida *Lazy Triplet Loss* y sus variantes, que en condiciones nubladas y de noche han tenido resultados similares e incluso mejores que la *Triplet Margin Loss*, pero en soleado han tenido errores mucho mayores. Esto se debe principalmente a que se ha producido un sobreajuste de la red al realizar el entrenamiento con imágenes tomadas únicamente bajo condiciones de iluminación nubladas.

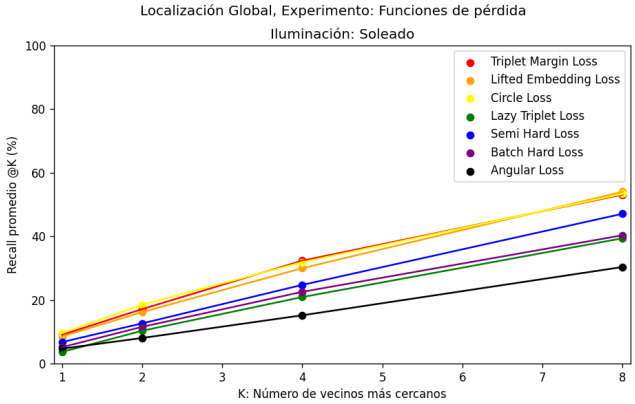
En la Figura **5-6** se muestran los mapas con las predicciones de la red para cada imagen de test para las tres condiciones de iluminación con la función de pérdida que ha tenido los mejores resultados en la localización global. Los puntos azules conforman el modelo visual del edificio, mientras que el resto representan las imágenes de test. Si se produce un acierto con K=1, el punto se representa de color verde oscuro, si se acierta con K=2, el punto tomará un color verde claro, y así sucesivamente hasta K=8. Si no se acierta para K=8, el punto se representa de color rojo. Las líneas unen la imagen de test con la imagen del modelo visual predicha como la más cercana.



(a)

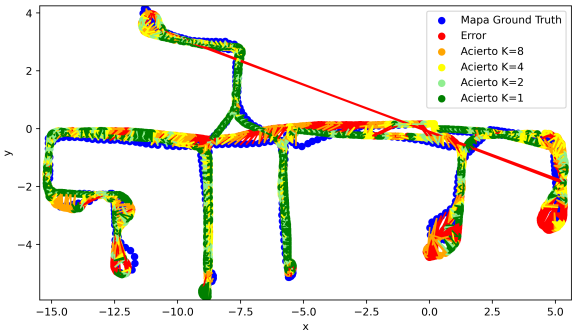


(b)

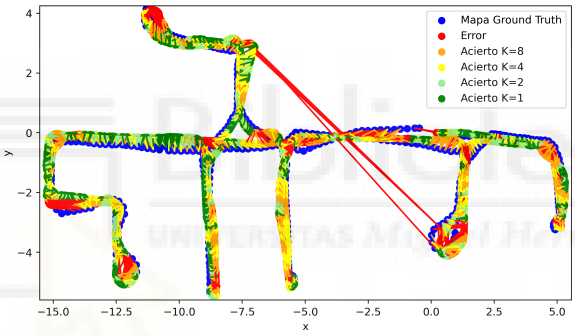


(c)

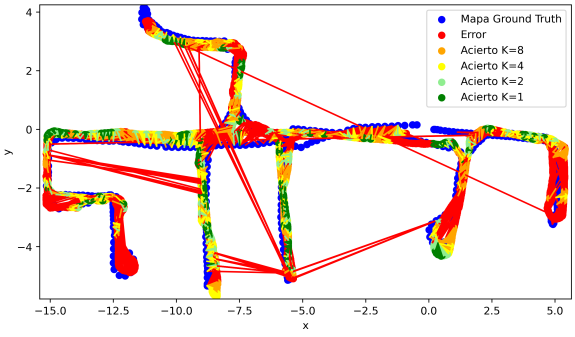
Figura 5-5: Recall para K = 1, 2, 4 y 8 en la localización global para cada función de pérdida en a) Nublado, b) Noche y c) Soleado.



(a)



(b)



(c)

Figura 5-6: Predicciones de la red para la localización global con la función de pérdida *Triplet Margin Loss* para a) Nublado, b) Noche y c) Soleado.

Se puede observar que un elevado porcentaje de predicciones son correctas para K=8, especialmente en nublado y de noche. Además, se producen muy pocos errores al predecir la habitación en la que se encuentra el robot. Estos errores aumentan significativamente en soleado, muchos de ellos entre habitaciones no conectadas, debido al sobreajuste de la red comentado previamente.

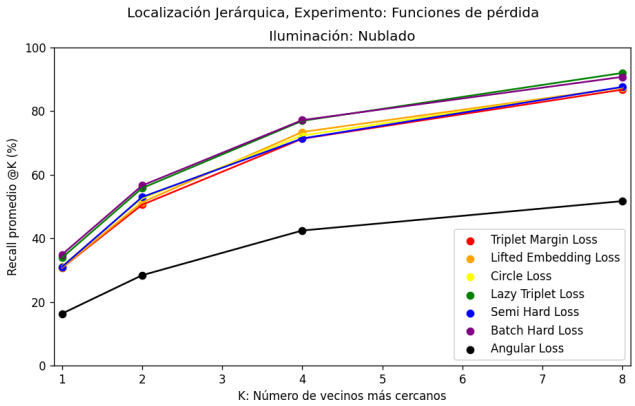
5.4. EXPERIMENTOS ADICIONALES

En el apartado 5.2 se ha llevado a cabo la localización jerárquica en dos pasos. En primer lugar, se ha determinado en qué estancia se encuentra el robot y, en segundo lugar, se han determinado las coordenadas en las que el robot ha capturado la imagen de test comparándola con las imágenes del modelo visual de dicha habitación, suponiendo un acierto del 100 % en la primera fase. En este apartado, se ha abordado el mismo problema realizando el test de manera más estricta.

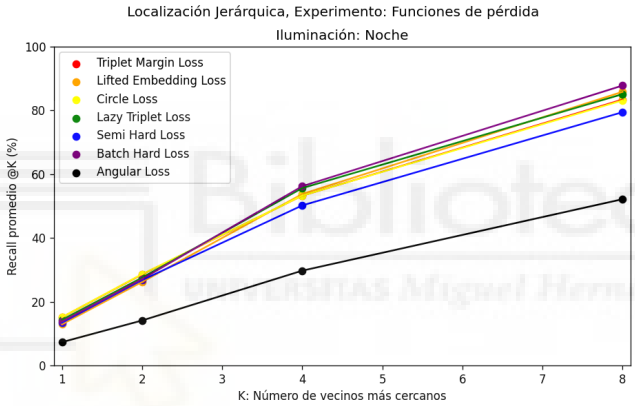
En primer lugar, se ha comparado cada imagen de test con la imagen representativa de cada habitación. Como en el caso anterior, la imagen representativa cuyo descriptor devuelve la menor distancia euclídea o la mayor similitud coseno con el descriptor de la imagen de test indicará la habitación en la que se encuentra el robot. Para la localización fina, se ha comparado la imagen de test con las imágenes del modelo visual de la habitación predicha en la primera etapa, aunque la predicción sea errónea. De esta forma, los errores cometidos en la localización gruesa se trasladan a la etapa de localización fina.

Estudio de la función de pérdida de triplete escogida

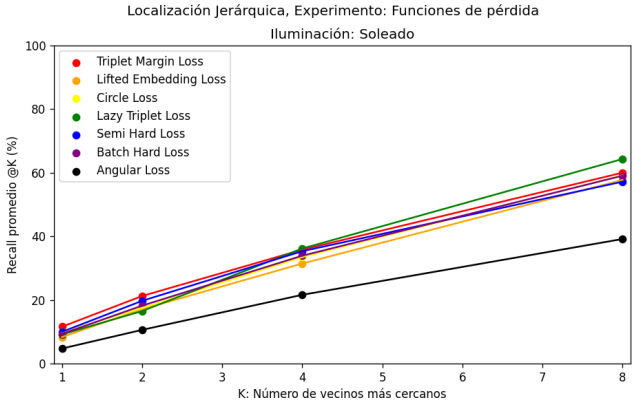
En la Figura **5-7** y en la Tabla **5-7** se muestran el acierto para K vecinos más cercanos y el error en metros cometido por la red, respectivamente, para cada función de pérdida y para cada condición de iluminación por separado.



(a)



(b)



(c)

Figura 5-7: Recall para K = 1, 2, 4 y 8 en la localización jerárquica para cada función de pérdida en a) Nublado, b) Noche y c) Soleado.

Función de pérdida	**Error distancia(m)**		
	Cloudy (Error mínimo: 0.128m)	**Cloudy (Error mínimo: 0.126m)**	**Cloudy (Error mínimo: 0.120m)**
Triplet margin loss(m=1)	0,273	0,300	0,899
Lifted embedding loss(m=0)	0,325	0,306	0,996
Circle loss(γ=1,m=0)	0,540	0,294	1,041
Lazy triplet loss(m=1)	0,240	0,287	0,658
Semi hard loss(m=0.5)	0,281	0,343	0,938
Batch hard loss(m=0.5)	0,231	0,280	0,884
Angular loss(α=20)	1,674	1,396	2,290

Tabla 5-7: Error medio cometido en metros en la localización jerárquica para cada función de pérdida en las 3 condiciones de iluminación.

Se puede observar en la Figura **5-7** y en la Tabla **5-7** que la función de pérdida *Lazy Triplet Loss* ha proporcionado los mejores resultados. Sus variantes, la *Semi Hard Loss* y la *Batch Hard Loss*, y la *Triplet Margin Loss* también han tenido resultados superiores al resto. Excepto la *Angular Loss*, todas las funciones de pérdida han tenido un rendimiento similar. Los errores para nublado y noche son pequeños pero en soleado aumentan considerablemente, ya que al producirse más errores en la localización gruesa, la influencia en el error medio es mayor.

Como es lógico, el error cometido ha aumentado en todos los casos en comparación con los experimentos realizados en el apartado 5.2.2. En este caso, las predicciones erróneas realizadas por la red entrenada para la localización gruesa provocan que la red entrenada para la localización fina compare la imagen de test con las imágenes del modelo visual de una habitación distinta y no sepa localizarla correctamente. En las zonas de transición, la red puede ser capaz de realizar predicciones más acertadas, pero cuando se produce una confusión entre estancias no conectadas, el error cometido en la predicción es muy elevado. Esto se puede observar de forma más clara en el test en condiciones soleadas. Al producirse más predicciones erróneas en la localización gruesa, el rendimiento de la red entrenada para la localización fina empeora considerablemente.

Estudio de la variación de la distancia umbral r

En la Figura **5-8** y en la Tabla **5-8** se muestran el acierto para K vecinos más cercanos y el error en metros cometido por la red, respectivamente, con la función de pérdida *Lazy Triplet Loss* para cada valor de r y para cada condición de iluminación por separado.

r(m)	Error distancia(m)		
	Cloudy (Error mínimo: 0,128m)	**Night (Error mínimo: 0,126m)**	**Sunny (Error mínimo: 0,120m)**
0,3	0,240	0,287	0,658
0,4	0,254	0,285	0,775
0,5	0,278	0,294	0,782

Tabla 5-8: Error medio cometido en metros en la localización jerárquica para cada valor de r en las 3 condiciones de iluminación.

A partir de la Figura **5-8** y la Tabla **5-8** se puede obtener la misma conclusión que en el apartado 5.2: al aumentar el valor de r, el desempeño de la red empeora. Esto se debe a que al aumentar r, para cada imagen ancla hay más imágenes positivas posibles, por lo tanto el entrenamiento de la red es menos exigente.

En la Tabla **5-9** se muestra el error cometido en metros para cada habitación con la función de pérdida y el valor de r que han proporcionado los mejores resultados en los experimentos anteriores.

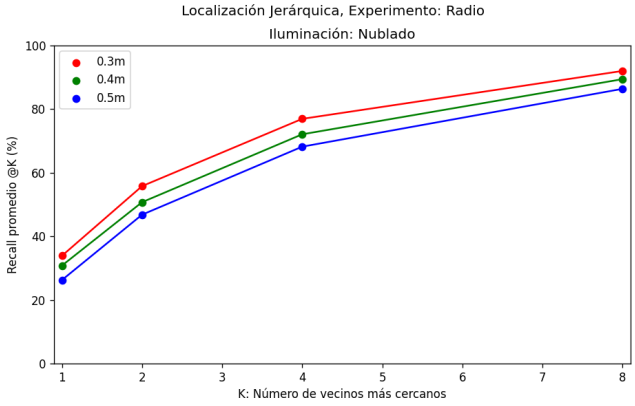
Estancia	Cloudy		Night		Sunny	
	Error(m)	Error mínimo(m)	Error(m)	Error mínimo(m)	Error(m)	Error mínimo(m)
1PO-A	0,148	0,087	0,251	0,128	0,353	0,103
2PO1-A	0,172	0,067	0,235	0,067	0,381	0,087
2PO2-A	0,105	0,062	0,289	0,168	0,286	0,081
CR-A	0,246	0,121	0,342	0,145	0,591	0,124
KT-A	0,280	0,148	0,223	0,106	0,290	0,163
LO-A	0,326	0,231	0,342	0,207	1,175	0,164
PA-A	0,431	0,201	0,306	0,095	2,429	0,130
ST-A	0,120	0,074	0,218	0,119	0,260	0,105
TL-A	0,197	0,146	0,177	0,073	0,393	0,104
Medio	**0,240**	**0,128**	**0,287**	**0,126**	**0,658**	**0,120**

Tabla 5-9: Error medio cometido en metros en la localización jerárquica para la función de pérdida *Lazy Triplet Loss* con r=0'3 m en las 3 condiciones de iluminación y errores mínimos que se pueden cometer para cada habitación.

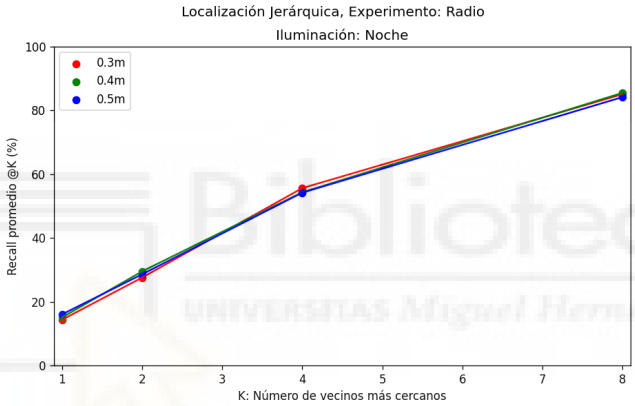
A partir de la Tabla **5-9** se puede observar que el error obtenido es muy variable en función de la estancia, ya que para cada una de ellas se ha entrenado una red distinta. El error también depende de la diferencia entre las trayectorias seguidas

por el robot en las secuencias de entrenamiento y test. Podemos observar que los mayores errores se obtienen en las estancias en las que se ha producido un mayor número de predicciones erróneas en la localización gruesa. Esto se puede observar de forma más clara en los resultados obtenidos para la *Lazy Triplet Loss* en la estancia PA-A, especialmente en soleado. En esta estancia se ha obtenido un error de 2'429 m cuando el error medio para soleado ha sido de 0'658 m. A partir de la matriz de confusión correspondiente a este experimento se puede obtener que el recall para la estancia PA-A en la localización gruesa es igual al 74'87 %, un valor muy inferior a la precisión media para soleado (92'53 %).

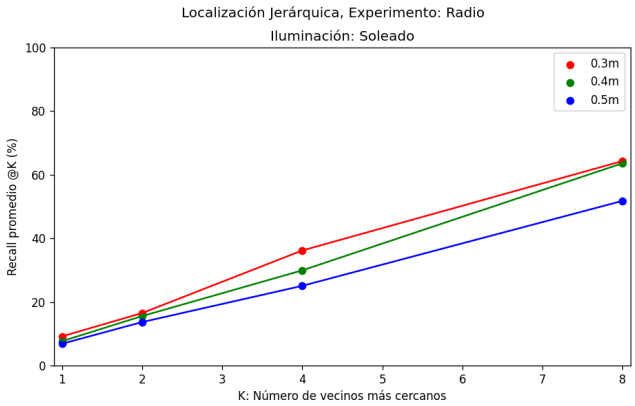
En la Figura **5-9** se muestran la trayectoria del robot en las secuencias de test y del modelo visual con las predicciones de la red para cada imagen de test en una de las habitaciones para las tres condiciones de iluminación. Estos mapas muestran los resultados anteriores de una forma más visual. En primer lugar, se puede observar que para nublado y noche prácticamente no se producen errores de confusión entre habitaciones, mientras que para soleado la cantidad aumenta considerablemente. También se puede comprobar que en las estancias LO-A y PA-A (las que se encuentran en la derecha del mapa) es donde se ha producido un mayor error. Además, en las zonas de transición entre habitaciones el error también es elevado, ya que la red encuentra una mayor dificultad para predecir la habitación en la que se encuentra el robot.



(a)

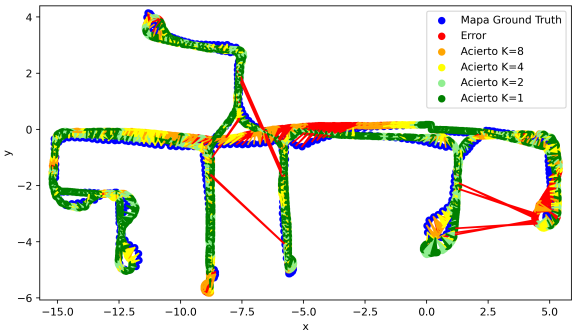


(b)

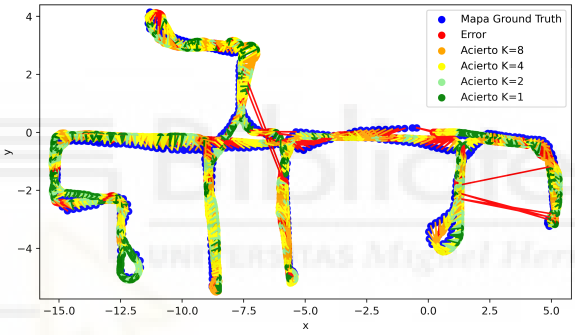


(c)

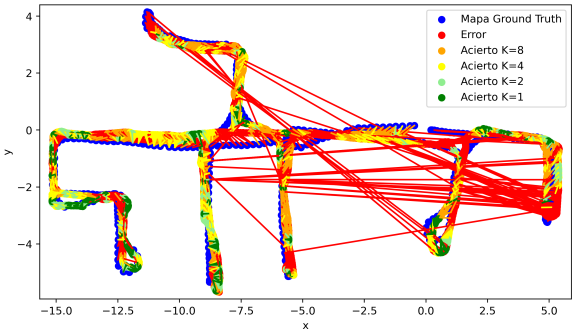
Figura 5-8: Recall para K = 1, 2, 4 y 8 en la localización jerárquica con la función de pérdida *Lazy Triplet Loss* para cada valor de r en a) Nublado, b) Noche y c) Soleado.



(a)



(b)



(c)

Figura 5-9: Predicciones de la red para la localización jerárquica con la función de pérdida *Lazy Triplet Loss* con r=0'3 m para a) Nublado, b) Noche y c) Soleado.

6 CONCLUSIONES Y LÍNEAS DE INVESTIGACIÓN FUTURAS

En este trabajo se ha llevado a cabo la tarea de localización mediante Redes Neuronales Tripletas. Para ello, se han propuesto dos métodos distintos: la localización jerárquica y la localización global. Se ha empleado el modelo de red VGG16 y se han mantenido sus capas convolucionales, correspondientes a la extracción de características. Únicamente se han modificado sus capas totalmente conectadas para obtener como salida un vector descriptor de 5 elementos. Además, se ha empleado la técnica de *Transfer Learning*, para aprovechar los conocimientos ya adquiridos por la red en la tarea de clasificación de imágenes. El rendimiento de la red se ha evaluado bajo 3 condiciones de iluminación distintas: nublado, noche y soleado, con el objetivo de analizar la robustez de las Redes Tripletas frente a cambios de iluminación.

Para cada uno de los métodos de localización se ha realizado un estudio del efecto de la función de pérdida escogida en el desempeño de la red para esta tarea. También se ha estudiado la variación de la distancia umbral r que separa los ejemplos positivos y negativos en la localización fina y en la localización global.

En la fase de la localización gruesa, se ha obtenido un acierto de la estancia en la que se ha capturado la imagen del 98'65 % para nublado, 97'16 % para noche y 92'53 % para soleado. La función de pérdida *Lazy Triplet Loss* con $m = 1$ han tenido los mejores resultados y la *Triplet Margin Loss*, con m=1, también ha tenido resultados ligeramente superiores al resto. A excepción de la *Angular Loss*, que ha tenido un rendimiento inferior, los resultados obtenidos para el resto de funciones de pérdida han sido muy similares. También cabe destacar que los parámetros que han devuelto mejores resultados para las funciones de pérdida *Circle Loss* y *Angular Loss* no se corresponden con los valores comúnmente utilizados en experimentos de *Image Retrieval* que han utilizado estas funciones de pérdida ([34], [39]).

En la fase de la localización fina, se ha obtenido un error mínimo de 0'223 m para nublado, 0'277 m para noche y 0'441 m para soleado, suponiendo un acierto del 100 % en la etapa de localización gruesa. Para esta tarea, la función de pérdida *Lazy Triplet Loss* ha tenido los mejores resultados y sus variantes, la *Semi Hard Loss* y la *Batch*

Hard Loss, así como la función de pérdida *Triplet Margin Loss*, también han tenido mejores resultados que el resto. En cuanto al estudio de la variación del umbral r, al aumentar esta distancia los resultados empeoran. Esto concuerda con lo esperado: si se aumenta la distancia r, cada imagen ancla tendrá más ejemplos positivos posibles, por tanto el entrenamiento de la red es menos exigente.

En líneas generales, los errores cometidos son pequeños y la diferencia de las trayectorias seguidas por el robot en las secuencias de entrenamiento y de test es un factor determinante en el error cometido. Además, el desempeño de la red es muy variable según la habitación en la que se realice el test. Los mayores errores se han cometido en el pasillo, ya que sus dimensiones son considerablemente superiores al resto y tiene el mayor número de habitaciones colindantes.

Al realizar la localización jerárquica de forma más estricta, como es lógico, los resultados han empeorado, ya que los errores cometidos en la etapa de localización gruesa se trasladan a la localización fina. En este caso, la función de pérdida *Lazy Triplet Loss* ha tenido los mejores resultados (0'240 m para nublado, 0'287 m para noche y 0'658 m para soleado). Sus variantes y la *Triplet Margin Loss* también han tenido resultados superiores al resto, mientras que la *Angular Loss* ha tenido errores muy elevados. En lo que respecta al estudio de la distancia umbral r, se ha llegado a la misma conclusión que en el caso anterior.

Los errores obtenidos en cada habitación presentan una elevada dispersión. En las estancias donde la red ha tenido un mejor desempeño en la localización gruesa, el error cometido en la localización fina ha sido pequeño. Por otro lado, en las estancias donde se han cometido más predicciones erróneas en la primera etapa, se han obtenido errores muy elevados en la segunda etapa.

En la localización global, se ha obtenido un error mínimo de 0'303 m para nublado, 0'324 m para noche y 0'633 m para soleado con la función de pérdida *Triplet Margin Loss*. El resto de funciones de pérdida han tenido resultados peores, especialmente en condiciones soleadas. Esto se debe a que se ha producido un sobreajuste de la red a la condición de iluminación de entrenamiento, empeorando los resultados para la condición de iluminación más distinta. Este sobreajuste se puede observar de forma más clara en los resultados obtenidos con la función de pérdida *Lazy Triplet* y sus variantes, ya que han tenido resultados similares e incluso mejores que la *Triplet Margin* para nublado y noche, pero en soleado han duplicado el error, incluso triplicado en el caso de la *Batch Hard*.

Si se realiza una comparación entre los dos métodos, se puede observar que la localización jerárquica presenta mejores resultados en líneas generales. Sin embargo, el rendimiento obtenido en la localización global no disminuye demasiado en comparación con la localización jerárquica, y presenta la ventaja de ser capaz de determinar las coordenadas en las que se ha capturado una imagen en un solo paso. Por lo tanto, ambos métodos permiten llevar a cabo la tarea de localización de un robot móvil, aunque el método de localización jerárquica presenta una mejor precisión.

En conclusión, las Redes Neuronales Tripletas permiten realizar la tarea de *Image Retrieval* con una gran precisión y robustez frente a cambios de iluminación. Las Redes Tripletas solucionan los problemas de las Redes Siamesas como el sesgo introducido al variar la proporción de imágenes similares y diferentes o la escasez de datos. Además, se ha demostrado que la elección de la función de pérdida y de la distancia umbral r utilizada en la generación de ejemplos positivos y negativos de entrenamiento influyen significativamente en el desempeño de la red en todas las etapas del proceso de localización. La función de pérdida *Lazy Triplet Loss* y sus variantes, así como la *Triplet Margin Loss*, han demostrado ser una elección adecuada para abordar el problema de *Image Retrieval*. Por otro lado, también se ha demostrado que realizar un entrenamiento de la red más exigente proporciona un mejor rendimiento de la red.

En cuanto a trabajos futuros, se pueden utilizar otras redes múltiples como las Redes Neuronales Cuadrupletas, caracterizadas por estar formadas por cuatro subredes simples idénticas que comparten sus pesos internos, o emplear otro tipo de redes neuronales como los Transformers Visuales. Además, se podría estudiar cómo afecta al desempeño de la red el uso de ejemplos con mayor o menor dificultad, o incluso dejar que sea la propia red la que selecciona los ejemplos con los que es entrenada. Este proceso es conocido como *Data Mining*.

Bibliografía

[1] H. Andreasson and T. Duckett. Topological localization for mobile robots using omni-directional vision and local features. *IFAC Proceedings Volumes*, 37(8):36–41, 2004. ISSN 1474-6670. doi: https://doi.org/10.1016/S1474-6670(17)31947-X. URL `https://www.sciencedirect.com/science/article/pii/S147466701731947X`. IFAC/EURON Symposium on Intelligent Autonomous Vehicles, Lisbon, Portugal, 5-7 July 2004.

[2] R. Arroyo, P. F. Alcantarilla, L. M. Bergasa, and E. Romera. Fusion and binarization of cnn features for robust topological localization across seasons. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4656–4663, 2016. doi: 10.1109/IROS.2016.7759685.

[3] G. Bae, M. de La Gorce, T. Baltrusaitis, C. Hewitt, D. Chen, J. Valentin, R. Cipolla, and J. Shen. Digiface-1m: 1 million digital face images for face recognition, 2022.

[4] M. Ballesta, L. Payá, S. Cebollada, O. Reinoso, and F. Murcia. A cnn regression approach to mobile robot localization using omnidirectional images. *Applied Sciences*, 11(16), 2021. ISSN 2076-3417. doi: 10.3390/app11167521. URL `https://www.mdpi.com/2076-3417/11/16/7521`.

[5] M. Betke and L. Gurvits. Mobile robot localization using landmarks. *IEEE Transactions on Robotics and Automation*, 13(2):251–263, 1997. doi: 10.1109/70.563647.

[6] J. J. Cabrera, S. Cebollada, M. Flores, Óscar Reinoso, and L. Payá. Training, optimization and validation of a cnn for room retrieval and description of omnidirectional images, 2022. URL `https://doi.org/10.1007/s42979-022-01127-8`.

[7] J. J. Cabrera, L. Payá, and A. Gil. Estudio de redes neuronales siamesas para la creación de modelos y localización de robots móviles. *(Trabajo de Fin de Máster). Universidad Miguel Hernández de Elche, España.*, 2022.

[8] S. Cebollada, L. Payá, X. Jiang, and Reinoso. Development and use of a convolutional neural network for hierarchical appearance-based localization. *Artificial Intelligence Review*, 55, 04 2022. doi: 10.1007/s10462-021-10076-2.

[9] W. Chen, X. Chen, J. Zhang, and K. Huang. Beyond triplet loss: A deep quadruplet network for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[10] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

[11] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *CoRR*, abs/2010.11929, 2020. URL `https://arxiv.org/abs/2010.11929`.

[12] J. Gaspar, N. Winters, and J. Santos-Victor. Vision-based navigation and environmental representations with an omnidirectional camera. *IEEE Transactions on Robotics and Automation*, 16(6):890–898, 2000. doi: 10.1109/70.897802.

[13] A. Gordo, J. Almazán, J. Revaud, and D. Larlus. Deep image retrieval: Learning global representations for image search. In B. Leibe, J. Matas, N. Sebe, and M. Welling, editors, *Computer Vision – ECCV 2016*, pages 241–257, Cham, 2016. Springer International Publishing. ISBN 978-3-319-46466-4.

[14] D. Hahnel, R. Triebel, W. Burgard, and S. Thrun. Map building with mobile robots in dynamic environments. In *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*, volume 2, pages 1557–1563 vol.2, 2003. doi: 10.1109/ROBOT.2003.1241816.

[15] A. Hermans, L. Beyer, and B. Leibe. In defense of the triplet loss for person re-identification, 2017.

[16] S. Ingle and M. Phute. *International Research Journal of Engineering and Technology (IRJET)*, 3:369–372, 09 2016.

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Commun. ACM*, 60(6):84–90, may 2017. ISSN 0001-0782. doi: 10.1145/3065386. URL `https://doi.org/10.1145/3065386`.

[18] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791.

[19] J. Leonard and H. Durrant-Whyte. Mobile robot localization by tracking geometric beacons. *IEEE Transactions on Robotics and Automation*, 7(3):376–382, June 1991. ISSN 1042296X. doi: 10.1109/70.88147. URL `http://ieeexplore.ieee.org/document/88147/`.

[20] Q. Li, W. Cai, X. Wang, Y. Zhou, D. D. Feng, and M. Chen. Medical image classification with convolutional neural network. In *2014 13th International Conference on Control Automation Robotics Vision (ICARCV)*, pages 844–848, 2014. doi: 10.1109/ICARCV.2014.7064414.

[21] W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 5(4):115–133, Dec. 1943. ISSN 0007-4985, 1522-9602. doi: 10.1007/BF02478259. URL `http://link.springer.com/10.1007/BF02478259`.

[22] E. Menegatti, A. Pretto, A. Scarpa, and E. Pagello. Omnidirectional vision scan matching for robot localization in dynamic environments. *IEEE Transactions on Robotics*, 22(3):523–535, 2006. doi: 10.1109/TRO.2006.875495.

[23] A. C. Murillo, J. J. Guerrero, and C. Sagues. Surf features for efficient robot localization with omnidirectional images. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 3901–3907, 2007. doi: 10.1109/ROBOT.2007.364077.

[24] A. C. Murillo, G. Singh, J. Kosecká, and J. J. Guerrero. Localization in urban environments using a panoramic gist descriptor. *IEEE Transactions on Robotics*, 29(1):146–160, 2013. doi: 10.1109/TRO.2012.2220211.

[25] D. Murray and C. Jennings. Stereo vision based mapping and navigation for mobile robots. In *Proceedings of International Conference on Robotics and Automation*, volume 2, pages 1694–1699 vol.2, 1997. doi: 10.1109/ROBOT.1997.614387.

[26] 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), 2018. ISSN 2072-4292. doi: 10.3390/rs10040522. URL `https://www.mdpi.com/2072-4292/10/4/522`.

[27] A. Pronobis and B. Caputo. COLD: COsy Localization Database. *The International Journal of Robotics Research (IJRR)*, 28

(5):588–594, May 2009. doi: 10.1177/0278364909103912. URL http://www.pronobis.pro/publications/pronobis2009ijrr.

[28] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408, 1958. ISSN 1939-1471, 0033-295X. doi: 10.1037/h0042519. URL http://doi.apa.org/getdoi.cfm?doi=10.1037/h0042519.

[29] E. Royer, M. Lhuillier, M. Dhome, and J.-M. Lavest. Monocular vision for mobile robot localization and autonomous navigation. *International Journal of Computer Vision*, 74(3):237–260, July 2007. ISSN 0920-5691, 1573-1405. doi: 10.1007/s11263-006-0023-y. URL http://link.springer.com/10.1007/s11263-006-0023-y.

[30] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, Oct. 1986. ISSN 0028-0836, 1476-4687. doi: 10.1038/323533a0. URL https://www.nature.com/articles/323533a0.

[31] P.-E. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[32] S. Se, D. Lowe, and J. Little. Vision-based global localization and mapping for mobile robots. *IEEE Transactions on Robotics*, 21(3):364–375, 2005. doi: 10.1109/TRO.2004.839228.

[33] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition, 2015.

[34] Y. Sun, C. Cheng, Y. Zhang, C. Zhang, L. Zheng, Z. Wang, and Y. Wei. Circle loss: A unified perspective of pair similarity optimization, 2020. URL https://arxiv.org/abs/2002.10857.

[35] 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.

[36] L. Tai and M. Liu. Mobile robots exploration through cnn-based reinforcement learning. *Robotics and Biomimetics*, 3(1):24, Dec. 2016. ISSN 2197-3768. doi: 10.1186/s40638-016-0055-x. URL http://jrobio.springeropen.com/articles/10.1186/s40638-016-0055-x.

[37] S. Thrun, W. Burgard, and D. Fox. A probabilistic approach to concurrent mapping and localization for mobile robots. *Autonomous Robots*, 5 (3/4):253–271, 1998. ISSN 09295593. doi: 10.1023/A:1008806205438. URL http://link.springer.com/10.1023/A:1008806205438.

[38] M. A. Uy and G. H. Lee. Pointnetvlad: Deep point cloud based retrieval for large-scale place recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[39] J. Wang, F. Zhou, S. Wen, X. Liu, and Y. Lin. Deep metric learning with angular loss. 08 2017.

[40] T.-H. Wang, H.-J. Huang, J.-T. Lin, C.-W. Hu, K.-H. Zeng, and M. Sun. Omnidirectional cnn for visual place recognition and navigation. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2341–2348, 2018. doi: 10.1109/ICRA.2018.8463173.

[41] D. F. Wolf and G. S. Sukhatme. Mobile robot simultaneous localization and mapping in dynamic environments. *Autonomous Robots*, 19(1):53–65, July 2005. ISSN 0929-5593, 1573-7527. doi: 10.1007/s10514-005-0606-4. URL http://link.springer.com/10.1007/s10514-005-0606-4.

[42] D. F. Wolf and G. S. Sukhatme. Semantic mapping using mobile robots. *IEEE Transactions on Robotics*, 24(2):245–258, 2008. doi: 10.1109/TRO.2008.917001.

[43] 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.

[44] 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. ISSN 1424-8220. doi: 10.3390/s19020249. URL https://www.mdpi.com/1424-8220/19/2/249.

[45] H. Yin, Y. Wang, X. Ding, L. Tang, S. Huang, and R. Xiong. 3d lidar-based global localization using siamese neural network. *IEEE Transactions on Intelligent Transportation Systems*, 21(4):1380–1392, 2020. doi: 10.1109/TITS.2019.2905046.

[46] Z. Zhang and H. Peng. Deeper and wider siamese networks for real-time visual tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.