Hierarchical localization with panoramic views and triplet loss functions

Marcos Alfaro[a,*], Juan José Cabrera[a], María Flores[a], Óscar Reinoso[a,b], Luis Payá[a,b]

[a] Miguel Hernández University, Avenida de la Universidad s/n, Elche, 03202, Comunidad Valenciana, Spain
[b] Valencian Graduate School and Research Network of Artificial Intelligence, Camí de Vera, Building 3Q, Valencia, 46020, Comunidad Valenciana, Spain

Abstract

The main objective of this paper is to tackle visual localization, which is essential for the safe navigation of mobile robots. The solution we propose employs panoramic images and triplet convolutional neural networks. We seek to exploit the properties of such architectures to address both hierarchical and global localization in indoor environments, which are prone to visual aliasing and other phenomena. Considering their importance in these architectures, a complete comparative evaluation of different triplet loss functions is performed. The experimental section proves that triplet networks can be trained with a relatively low number of images captured under a specific lighting condition and even so, the resulting networks are a robust tool to perform visual localization under dynamic conditions. Our approach has been evaluated against some of these effects, such as changes in the lighting conditions, occlusions, noise and motion blurring. Furthermore, to explore the limits of our approach, triplet networks have been tested in different indoor environments simultaneously. In all the cases, these architectures have demonstrated a great capability to generalize to diverse and challenging scenarios. The code used in the experiments is available at https://github.com/MarcosAlfaro/TripletNetworksIndoorLocalization.git.

[*] Corresponding author: Marcos Alfaro

Email addresses: malfaro@umh.es (Marcos Alfaro), juan.cabreram@umh.es (Juan José Cabrera), m.flores@umh.es (María Flores), o.reinoso@umh.es (Óscar Reinoso), lpaya@umh.es (Luis Payá)

Keywords: robot localization, convolutional neural network, omnidirectional images, triplet loss

1. Introduction

Vision sensors are a suitable option to tackle robot localization, since they can capture a large amount of information from the environment at a low cost. Among these sensors, omnidirectional cameras stand out (Amorós *et al.* [1]). These cameras have a field of view up to 360º, so they capture complete information from the environment regardless of the robot orientation. Omnidirectional views can be obtained with different alternatives, such as multi-camera systems (Kneip *et al.* [2]), catadioptric systems (Lin *et al.* [3]) or the combination of a pair of fisheye cameras (Flores *et al.* [4]).

In order to describe the visual information from the scene, it can be conducted by means of global or local description. First, holistic or global description consists in working with the image information as a whole (Payá *et al.* [5]), whereas the description based on local features only focuses on those points or areas easily identifiable in an image, such as borders or corners (Murillo *et al.* [6]). In the present approach, global description is used.

Traditionally, analytical techniques have been used to create visual descriptors (Se *et al.* [7]). However, with the huge increase of computing power, the use of deep learning tools has increased during the past few years. In this context, Convolutional Neural Networks (CNNs) have revolutionized the field of image processing (Nilwong *et al.* [8], Cebollada *et al.* [9]). This type of neural networks apply filters to the image based on the convolution operation, and are able to extract features from the image with a high level of abstraction.

Concerning the training of CNNs, architectures composed of several branches of these networks have emerged in recent years, giving place to Siamese (Yin *et al.* [10]) and triplet networks (Liu and Huang [11]), among others. Siamese networks contain two identical neural networks, that is, they have the same architecture and share their weights, and work in parallel, in such a way that each of them receives a different input and provides a different output. Meanwhile, triplet networks receive three inputs, commonly called anchor, positive and negative, and provide three outputs. While siamese networks are typically used to learn if two inputs are similar or different, triplet networks are able to simultaneously learn similarities between the anchor and positive inputs and differences between the anchor and negative data. In the case of robot localization, triplet samples can be chosen in such a way that two of them are captured from similar positions and the other is captured from a different position. The fact of receiving three inputs permits the CNN to adjust both to positive and negative examples during the training process. Besides, since the number of possible combinations of three images is very large, a fairly small number of images captured by the robot can be enough to create a complete training set.

During the training process, the loss function compares the output provided by the CNN with the required output, and the optimization of this function leads to more accurate predictions. As a function of the loss value, the optimizer algorithm modifies the CNN weights to a greater or smaller extent. Triplet loss functions (Hermans *et al.* [12]) seek to minimize the difference between anchor and positive inputs and also seek to maximize the difference between anchor and negative inputs. This type of loss functions have some parameters that must be set before the training. The most relevant is the margin, which permits adjusting the required similarity and difference relationships between the data.

In this paper, a CNN model is used, which is adapted and retrained to tackle visual localization in indoor environments with panoramic images, by means of a triplet network. The experimental section shows the robustness of such architectures to address localization, with a direct comparison to siamese architectures. Thanks to it, a rapid training with a limited set of images captured under a specific lighting condition is enough to obtain a tool which is accurate and capable of adapting to adverse conditions without the need of a data augmentation process. In addition, an exhaustive comparative evaluation between several triplet losses has been performed in both localization methods: hierarchical and global. To evaluate the performance of triplet architectures in large indoor environments, which are prone to visual aliasing, three different indoor environments have been employed simultaneously to train and test the CNNs.

Therefore, the main contributions of this paper are:

- A hierarchical localization approach which exploits the advantages of triplet architectures in indoor environments is proposed.

- We conduct a complete comparative evaluation of the performance of different triplet loss functions in the global and the hierarchical localization.

- Triplet networks are trained with a limited set of images and thoroughly evaluated against defying visual phenomena that appear often in mobile robotics, such as changes in the lighting conditions, noise, occlusions or motion blur. Beside, the ability of the tool to generalize to different environments is assessed.

The rest of the manuscript is structured as follows. Section 2 reviews the state of the art on robot localization, holistic visual description and the use of deep learning to perform these tasks. Section 3 presents the CNN architecture and the loss functions used in this research. In Section 4, the two localization methods employed in this paper are detailed. Section 5 describes the experiments conducted. Finally, in Section 6 the conclusions and future works are outlined.

2. State of the art

Nowadays, the use of vision systems for mobile robotics applications is very common in the literature. Many researchers make use of cameras to solve the localization and mapping problems. Among this type of sensors, monocular cameras are the most extended option. For example, Xiao *et al.* [13] addressed the SLAM problem in dynamic environments with a monocular vision system. Other approaches make use of omnidirectional vision systems as they can capture complete information from the scenario regardless of the robot orientation. Flores *et al.* [4] perform localization with omnidirectional and fisheye cameras.

With respect to visual description, there are some authors, such as Payá *et al.* [5] or Cebollada *et al.* [14], that propose environment modeling techniques with global-appearance descriptors. Moreover, some researchers make use of such descriptors to tackle the loop closure problem, one of the most critical parts of SLAM algorithms (Zhang *et al.* [15]). Also, local descriptors are commonly used as well to perform localization (Kallasi *et al.* [16]). Furthermore, it is frequent to combine the two types of descriptors to address mapping and/or localization (Li *et al.* [17], Su *et al.* [18]).

The increase of computing power has led to the rise of CNNs in the past decade. When it comes to process visual information captured by a robot, this type of networks proved to be able to extract features from the image and therefore solve mobile robotics problems like visual localization. CNNs were first proposed in [19], and further developed in subsequent studies, which propose more complex architectures, such as VGG (Simonyan and Zisserman [20]), GoogLeNet (Szegedy *et al.* [21]) or AlexNet (Krizhevsky *et al.* [22]), all of them trained to classify a thousand different objects with the ImageNet database (Deng *et al.* [23]). Although CNNs are the most extended choice, lately other architectures have been proposed to process visual information. This is the case of Visual Transformers (Dosovitskiy *et al.* [24]), which are based on Transformers, commonly used in Natural Language Processing. Besides, other approaches propose different networks that are able to process 3D point clouds (Komorowski [25]).

Focusing on CNNs, many recent studies use them to address visual localization. For instance, Nilwong *et al.* [8] make use of local features obtained with a CNN from RGB images captured in outdoor environments, and Foroughi *et al.* [26] followed a similar procedure indoors. Others, such as Xu *et al.* [27], make use of feature descriptors extracted from different convolutional layers of the network. CNNs can also be trained to obtain global-appearance descriptors from the image (Cabrera *et al.* [28]). Moreover, Chen *et al.* [29] propose a two-step method by combining global and local features. First, an image retrieval phase takes place by comparing global image descriptors. Second, the robot pose is estimated by comparing the ORB keypoints of the captured image with the keypoints in the two most similar images. Rostkowska and Skrzypezynski [30], Ballesta *et al.* [31] and Cebollada *et al.* [9] also perform a hierarchical localization by identifying in first place the room where the robot has captured the image and later estimate the robot coordinates inside the room predicted in the first step. Besides, Wozniak *et al.* [32] train a CNN to classify images among 16 rooms.

Due to the success of CNNs, other approaches have implemented advanced architectures composed of several CNNs. Siamese networks are composed of two identical CNNs that work in parallel and share their weights. Apart from being able to extract global features from the image, Siamese networks can include some additional layers to evaluate the similarity between the two inputs. This ability can be used in mobile robotics tasks such as place recognition (Leyva-Vallina *et al.* [33]), loop closure (Qiu *et al.* [34]) or visual localization (Oliveira *et al.* [35]). Other researchers have designed siamese architectures to process other kinds of sensory data. For example, Chen *et al.* [36] make use of a siamese network to evaluate LiDAR scan similarity. Each network receives a LiDAR 3D point cloud and embeds the representation into the Euclidean space to estimate their similarity.

Concerning triplet architectures, they have barely been used in visual localization tasks, and only few approaches can be found in recent years. Also, all of them used standard cameras or RGB-d cameras. Arandjelovic *et al.* [37] designed a triplet network that aggregates the extracted local features into a single descriptor using a VLAD layer. Yu *et al.* [38] also make use of a VLAD layer to address the same problem. López-Antequera *et al.* [39] proposed a triplet architecture to carry out a visual localization under seasonal changes. Likewise, Olid *et al.* [40] make a comparative evaluation of several CNN, siamese and triplet networks, obtaining the highest recall with triplet architectures. Comparing to these works, in the present work we propose a hierarchical localization approach, which exploits the advantages of the triplet networks in challenging indoor environments. Also, we explore the use of triplet networks along with panoramic images, obtained from a catadioptric system mounted on the robot both to train and test the architectures.

The development of triplet networks goes hand in hand with the design of triplet loss functions. Several studies have focused on creating a loss function that optimizes the training of their triplet architecture. Hermans *et al.* [12] compare different triplet loss functions used to train a CNN for people recognition. Cheng *et al.* [41] use a variant of the Triplet Margin Loss, proposed in [12], to solve the same problem. Nevertheless, there have been only few approaches that designed a triplet loss function to tackle visual localization. In this sense, Liu *et al.* [42] created a triplet loss function and compared it with other loss functions to solve a place recognition problem. Also, Kim *et al.* [43] developed a triplet loss function to undertake a room retrieval task. Notwithstanding that, triplet loss functions have not been thoroughly tested in visual localization tasks and in the present manuscript we perform a complete comparative evaluation of the performance of such loss functions and the influence of their parameters in the global and hierarchical localization with panoramic images.

3. Architecture of the CNN and triplet losses

Triplet networks consist of three identical CNNs that work in parallel and share their weights, but each of them receives a different input and therefore will provide a different output. In some applications, triplet networks present some advantages over siamese networks. First, they receive the same number of positive and negative inputs, which allows the CNN to adjust equally to similar and different data during the training process. This property can be especially useful in localization tasks, especially in those indoor environments which are prone to visual aliasing. Second, the number of possible input combinations in the training process increases substantially compared with siamese networks. This can be especially useful when just a scarce dataset is initially available, because a reasonably high number of triplet samples can be obtained to train the CNN even if no data augmentation is performed. For these reasons, triplet architectures can play a remarkable role to solve the visual localization of a mobile robot, and we address this problem in the present approach.

In order to carry out localization by using a triplet architecture, we make use of the VGG-16 model [20], since it has proven to have a great ability to extract features despite its small number of parameters, which can be especially useful to perform localization in real time. This model was originally designed to classify objects among 1000 different classes. In order to adapt it to the localization task, we have modified its backbone as shown in the Figure 1. In first place, given that the size of the panoramic images is 128x512x3 pixels, the first fully connected layer of the feature aggregation stage must be adapted to this size (its original size was 224x224x3 pixels). Additionally, we leave the convolutional layers intact, which correspond to the feature extraction phase, and modify the remaining fully connected layers to obtain a five-element global-appearance descriptor. With the aim of taking advantage of the knowledge already acquired by the VGG-16 model, the transfer learning technique is employed on the convolutional layers.

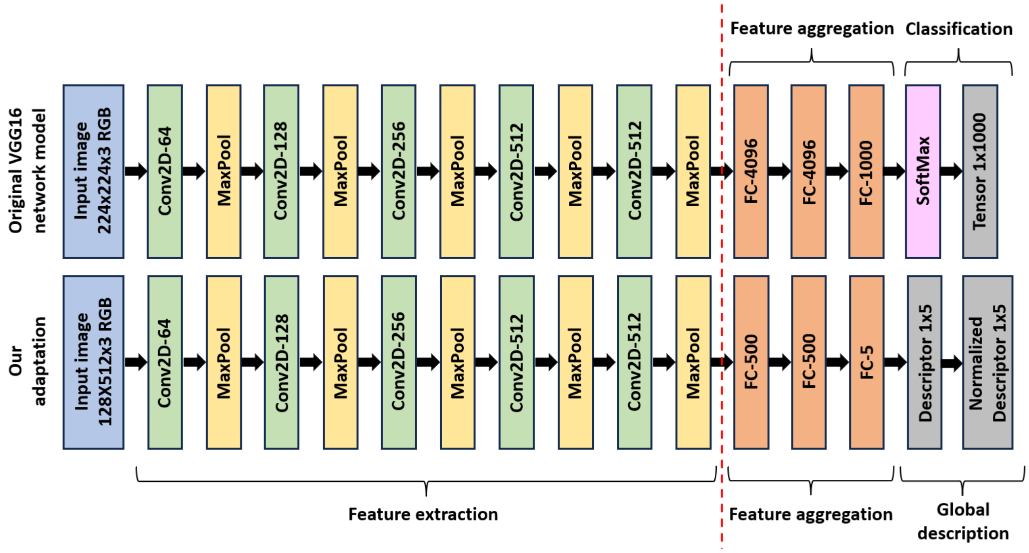


Figure 1. Original VGG-16 model (above) and our adaptation (below). Convolutional and max pooling layers have been left intact, whereas the fully connected layers have been modified in order to adapt the architecture to the size of the input images and obtain a five-element global descriptor. ReLU layers have not been included so as to simplify this figure.

During the training, the loss function compares the output provided by the CNN with the required output. Later, the optimizer algorithm will modify the network weights according to the committed error to optimize the value of the loss function and achieve a more accurate prediction. Therefore, triplet losses minimize their value when the anchor and positive inputs are predicted as similar and the negative input is predicted as different to the other two inputs. During the training process, the chosen loss function is expected to have an important influence on the performance of the trained network. In this paper, an exhaustive study is conducted to assess the influence of the loss function in the accuracy of the CNN when it is trained to solve the localization problem.

- **Triplet Margin Loss (TL)**: This is the most renowned triplet loss. It returns the average value of all the batch combinations:

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

where $D^i_{a,p}$ is the Euclidean distance between the anchor and positive descriptors in the i-th triplet, $D^i_{a,n}$ is the Euclidean distance between the anchor and negative descriptors, $[...]_+$ is the ReLU function, m is the margin and N is the batch size (number of triplet samples that are taken into account before updating the internal model parameters).

- **Lifted Embedding Loss (LE)**: This loss, described in [12], is characterized by not only taking into account the distance between the anchor and positive inputs and the distance between the anchor and negative inputs, but also trying to maximize the distance between the positive and negative inputs:

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

  where $D^i_{p,n}$ is the Euclidean distance between the positive and negative descriptors in the i-th triplet sample.

- **Lazy Triplet Loss (LT)**: This loss returns the hardest example of the batch for the network learning process:

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

  where $\vec{D}_{a,p} = (D^1_{a,p}, D^2_{a,p}, ..., D^N_{a,p})$ are the Euclidean distances between each anchor-positive pair and $\vec{D}_{a,n} = (D^1_{a,n}, D^2_{a,n}, ..., D^N_{a,n})$ are the Euclidean distances between each anchor-negative pair.

- **Semi Hard Loss (SH)**: This loss is a Lazy Triplet Loss variant. It calculates the average distance between the anchor and positive descriptors, and the minimum distance between the anchor and negative descriptors. In other words, it returns the hardest negative example of the batch:

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

- **Batch Hard Loss (BH)**: This loss is another variant of the Lazy Triplet Loss. It returns the maximum distance between the anchor and positive descriptors, and the minimum distance between the anchor and negative descriptors. Therefore, it returns the hardest positive and negative examples of the batch:

$$\mathcal{L} = \left[\max\left(\vec{D}_{a,p}\right) - \min\left(\vec{D}_{a,n}\right) + m\right]_+$$

- **Circle Loss (CL)**: This loss, proposed in [44], makes use of the cosine similarity metric instead of the Euclidean distance:

$$\mathcal{L} = \ln\left(1 + \sum_{j=1}^{N} e^{\gamma \alpha_n^j s_n^j} + \sum_{i=1}^{N} e^{-\gamma \alpha_p^i s_p^i}\right)$$

where,

$$\alpha_p^i = \left[O_p - s_p^i\right]_+ ; \alpha_n^j = \left[s_n^j - O_n\right]_+ ; O_p = 1 - m; O_n = m$$

where s_p^i is the cosine similarity between the anchor and positive descriptors, s_n^j is the cosine similarity between the anchor and negative descriptors and γ is a scale factor.

- **Angular Loss (AL)**: This loss, introduced in [45], seeks to minimize the angle formed by the vector that connects the anchor and the negative descriptors and the vector that connects the positive and the negative descriptors. Thus, it minimizes the distance between the anchor and positive inputs:

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

where,

$$f_{a,p,n}^i = 4\tan^2\alpha\left(x_a^i + x_p^i\right)^T x_n^i - 2\left(1 + tan^2\alpha\right)(x_a^i)^T x_p^i$$

where x_a^i is the anchor descriptor of the i-th triplet sample, x_p^i is the positive descriptor of the i-th triplet sample, x_n^i is the negative descriptor of the i-th triplet sample and α is an angular margin.

Moreover, the Contrastive Loss has been used to train a siamese architecture with the aim of comparing directly siamese and triplet architectures. Its equation is defined below:

$$\mathcal{L} = \frac{1}{N} \sum \left[(1-l) * D_i^2 + l * max(0, (m - D_i^2)\right]$$

where l is the label and D_i is the Euclidean distance between the i-th pair of descriptors.

4. Visual Localization

With the aim of addressing the localization problem, the present approach makes use of omnidirectional images captured in indoor environments by a catadioptric system mounted on a mobile robot. Subsequently, RGB images are converted to a panoramic format with 128x512x3 pixels and split into training, validation and test sets. Additionally, a visual model is generated with the images used during the training process. For every image, the coordinates of the capture points are known (ground truth), which allows us to conduct a supervised training.

Afterwards, we conduct the training, validation and test of the CNN proposed in Section 3. In every stage, a triplet architecture will be used to train the model, in such a way that the model is trained with combinations of three images I_a, I_p, I_n, where each of the branches that compose the network receives an input image and outputs a descriptor of that image.

In order to perform the validation and test, the CNN model will be used to embed each test image into a global-appearance descriptor $\vec{d}_{test} \in \mathbb{R}^{5x1}$ that will be compared with the rest of the image descriptors that constitute the visual map, composed of the images used during the training process. These descriptors are normalized and then compared using Euclidean distance or cosine similarity. The nearest neighbor among the images of the visual model will allow us to estimate the position of the robot when it captured the test image. The next subsections describe the two localization approaches: hierarchical localization and global localization.

4.1. Hierarchical localization

Hierarchical localization involves estimating the coordinates where the robot has captured an image in two steps (see Figure 2). First, we carry out a coarse localization, in which the CNN identifies the room where the robot is located. Second, a fine localization is performed, in which the CNN determines the robot coordinates in the room that has been retrieved in the first stage.

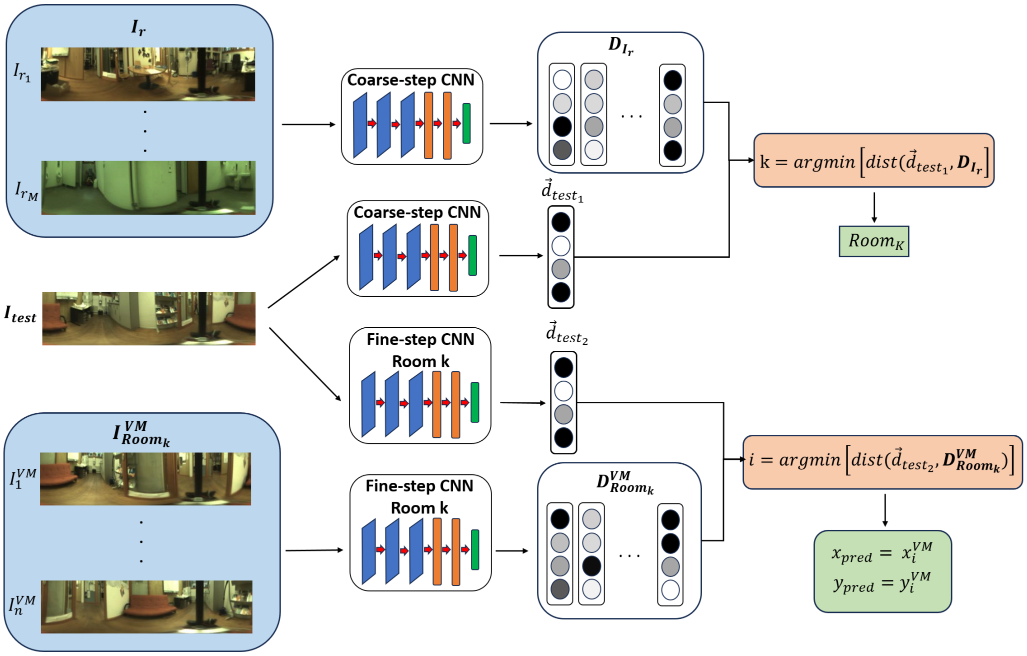


Figure 2. Hierarchical localization process performed in two steps: first, coarse localization (room retrieval); second, fine localization (estimate the robot coordinates inside the retrieved room).

- **Coarse localization**: in this stage, the CNN must determine in which room the test image has been taken. To do that, the triplet network is trained with combinations of three images I_a, I_p, I_n chosen randomly, in such a way that the anchor and positive images belong to the same room and the negative image must have been captured in a different room.

The CNN is trained to output a global descriptor per input image, as shown in Figure 1. Once trained, to test the CNN, the descriptor of each test image $\vec{d}_{test_1}$ is compared with a set of descriptors that contain a representative descriptor of every room $\boldsymbol{D_{I_r}} = \left[\vec{d}_{I_{r_1}}, \vec{d}_{I_{r_2}}, ..., \vec{d}_{I_{r_M}}\right]$. The representative image of every room is the image captured from the position which is the closest to the geometrical centre of the room, where M is the number of rooms. If the predicted room matches the actual room, it will be considered as a network success.

- **Fine localization**: Once a room has been retrieved, the CNN must estimate the robot position inside the room. To do this part, an independent triplet network is trained for each one of the rooms, starting from the weights of the coarse-step model. In this case, all the training images belong to the same room and a distance threshold is defined to consider positive or negative pairs. In this stage, the distance between anchor and positive images must be smaller than 0.3 m and the distance between anchor and negative images must be larger than 0.3 m. This threshold has not been chosen arbitrarily, since it is the minimum distance that permits every image to have at least one possible positive pair in the training dataset. To conduct the test, the descriptor of every test image $\vec{d}_{test_2}$ is compared with the descriptor of every image that belongs to the visual model (VM) of the room that has been retrieved during the coarse localization $\boldsymbol{D^{VM}_{Room_k}} = \left[\vec{d}_1^{\,VM}, \vec{d}_2^{\,VM}, ..., \vec{d}_n^{\,VM}\right]$, where n is the number of images in the visual model of the predicted room. The coordinates of the nearest neighbor are considered an estimation of the position of the robot when capturing the test image.

4.2. Global localization

Global localization consists in determining the robot position in the entire map in one step (see Figure 3). A unique CNN is trained for the whole environment, including images captured in all the rooms with random combinations. As in the fine localization, a distance threshold is set to create the positive and negative pairs: the distance between anchor and positive images must be smaller than 0.3 m and the distance between anchor and negative images must be larger than 0.3 m. In order to test the CNN, the descriptor of every test image $\vec{d}_{test}$ is compared with the descriptors of the visual model of the whole map $\boldsymbol{D^{VM}} = \left[\vec{d}_1^{\,VM}, \vec{d}_2^{\,VM}, ..., \vec{d}_n^{\,VM}\right]$, where n is the number of images in the complete visual model. Likewise, the coordinates of the nearest neighbor are considered an estimation of the position of the robot $(x_{pred}, y_{pred}) = \left(x_i^{VM}, y_i^{VM}\right)$.

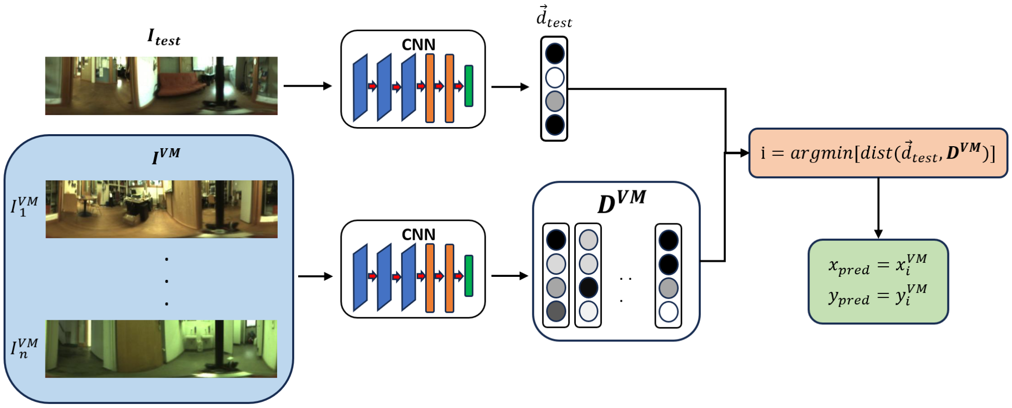


Figure 3. Global localization process performed in a unique step, which consists in estimating the coordinates of an image by retrieving the most similar image from the visual model.

The training of siamese architectures differs with respect to triplet networks. Therefore, to compare the proposed framework, which employs triplet architectures, with siamese networks, the selection of the training samples and the labeling method used in [46] have been followed to train the siamese network, since an exhaustive evaluation of siamese architectures is conducted in their study. Nevertheless, the rest of the training and test conditions have been set the same with the aim to perform a fair comparison.

5. Experiments

This section describes the dataset and the results of the experimental evaluation. In this manuscript, three experiments have been performed. Experiment 1 addresses a comparative evaluation of the influence of the triplet loss function in the performance of the CNN in a specific environment under different lighting conditions. Moreover, two localization approaches have been tackled: hierarchical and global, as described in Section 4. Experiment 2 analyzes the robustness of the trained triplet network against challenging effects. Finally, Experiment 3 evaluates the performance of triplet architectures when multiple environments are considered at the same time.

5.1. Dataset

The images used in this paper belong to COLD database (Pronobis and Caputo [47]). This dataset contains omnidirectional images captured by a mobile robot that makes use of a catadioptric vision system with a hyperbolic mirror. The robot follows different paths inside several buildings and goes through different rooms, taking a picture every 0.08 s, with a gap of roughly 20 cm between them. Various types of rooms can be found inside the buildings, such as offices, kitchens, toilets or corridors that connect the different rooms. In this dataset, images captured under three illumination conditions can be found: cloudy, night and sunny. Besides, some images include people moving or changes in the position of some pieces of furniture.

First, the Freiburg Part A environment (FR-A) from this dataset has been used in Experiments 1 and 2 to assess the performance of the tool in hierarchical and global localization, the influence of the triplet loss and the robustness against different conditions. Second, in order to analyze the capability of generalization of triplet architectures, we have made use of three different environments in Experiment 3: Freiburg Part A (FR-A), Saarbrücken Part A (SA-A) and Part B (SA-B). Despite the fact that two sets of images have been captured in the Saarbrücken building, they do not share any room, so they can be considered as two different environments.

Figure 4 shows some examples of images under each lighting condition and some examples of images that belong to each environment. These Figures illustrate some challenging cases that the network can find, such as changes of appearance caused by lighting variations or visual aliasing due to similar rooms that belong to different environments.

According to this philosophy, only cloudy images have been employed to conduct the training and validation, since it is the most standard illumination and it presents the lower contrast between the pixels corresponding to

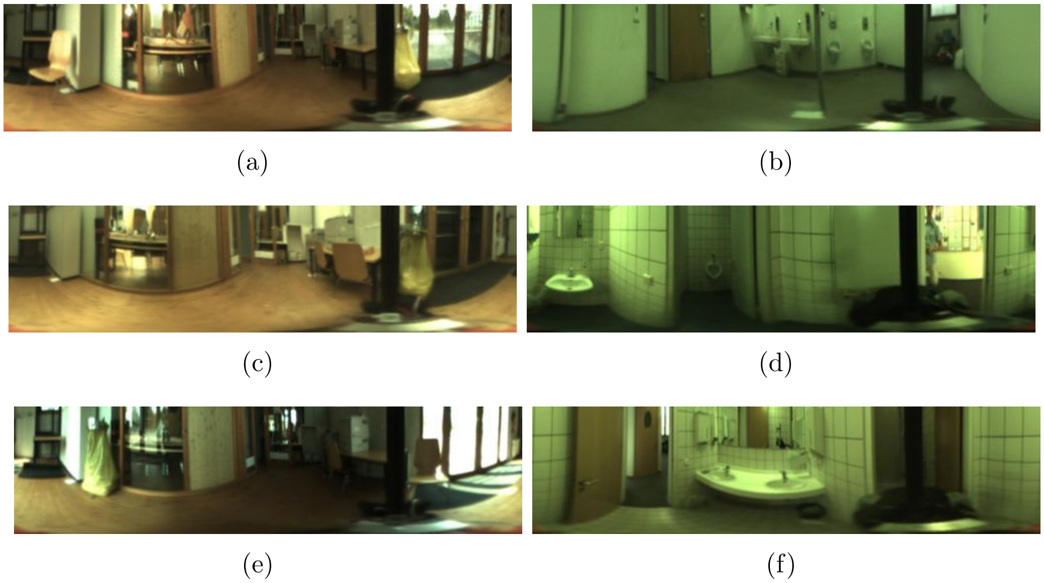

(a) (b) (c) (d) (e) (f)

Figure 4. Examples of images captured under different lighting conditions (a) Cloudy, c) Night, e) Sunny) and examples of images captured in different environments (b) Freiburg, d) Saarbrücken A, f) Saarbrücken B).

information indoors and outdoors. This training set has also been employed as a visual map during the validation and the test. Meanwhile, all the illumination conditions are used for the test, so as to prove the robustness of the trained model against changes in the lighting conditions. The training, validation and test sets do not share any of their images, that is, the validation and the test are carried out with images that the CNN has not seen during the training process.

Table 1 shows the number of images from each image set used in Experiment 1. For Experiment 2, the same image sets have been used, but in this case the studied effects (noise, occlusions and motion blur) are applied to the image sets.

Besides, Table 2 includes the number of images of each set employed in Experiment 3, according to the environment where they have been captured. The procedure has been the same as in Experiment 1, but two different validation sets have been employed. Validation 1 set has been employed for

Image set	Illumination	Freiburg
Train/Visual Model	Cloudy	556
Validation	Cloudy	586
Test 1	Cloudy	2595
Test 2	Night	2707
Test 3	Sunny	2114

Table 1. Size and lighting conditions of the training, validation and test sets used in Experiment 1.

the coarse-step training, whereas validation 2 set has been used for the fine step.

Image set	Illumination	FR-A	SA-A	SA-B	TOTAL
Train/Visual Model	Cloudy	556	586	321	1463
Validation 1	Cloudy	199	198	112	509
Validation 2	Cloudy	586	582	301	1469
Test 1	Cloudy	867	758	281	1906
Test 2	Night	905	759	292	1956
Test 3	Sunny	707	X	291	998

Table 2. Size and lighting conditions of the training, validation and test sets used in Experiment 3 (X indicates that the original COLD dataset contains no image in this set).

5.2. Experiment 1. Influence of the loss function.

In this experiment, a comparative evaluation has been conducted among different triplet loss functions (described in Section 3). For all localization stages, a network has been trained with each triplet loss, giving different values to the parameters of the loss function with the purpose of finding their optimal value for each task. Moreover, a comparison with siamese architectures is conducted in this experiment.

5.2.1. Hierarchical localization

a) Coarse localization

To train the model for the room retrieval task, the training process consists of 5 epochs and 50000 triplet samples per epoch. In Table 3 the best results obtained with each loss function are shown.

Room Retrieval Accuracy (%)				
Loss Function	**Cloudy**	**Night**	**Sunny**	**Average**
TL (m=1.25)	99.23	97.04	95.08	97.12
LE (m=0.25)	99.23	97.23	93.42	96.63
LT (m=1.25)	99.27	97.52	95.13	97.31
SH (m=1)	99.27	97.19	**95.55**	97.34
BH (m=0.75)	99.27	97.56	95.27	**97.37**
CL (γ=1, m=0)	**99.38**	**97.64**	93.05	96.69
AL (α=30º)	99.23	97.19	95.41	97.28
SNN (m=2)	99.11	97.30	94.18	96.86

Table 3. Test accuracy for each loss function in the coarse localization.

Table 3 reveals that the loss function that has output the best results is the Batch Hard (97.37% average accuracy), and it arrives to a good balance under cloudy, night and sunny conditions. The other variant of the Lazy Triplet, i.e. the Semi Hard, and the Lazy Triplet itself have output similar results as well. The Circle loss has obtained the best results under cloudy and night conditions, but it has had a worst performance under the sunny condition, which differs the most from the condition used during the training of the CNNs, i.e. cloudy. In this sense, the Semi Hard loss has proved to have less overfitting to the training condition than the rest of triplet losses. If the results output by the CNNs trained with a triplet architecture and the ones obtained with a siamese CNN are compared, in general terms, triplet loss functions have lead to a better performance under every lighting condition, especially with sunny images.

b) Fine localization

In this phase, a CNN is trained in order to estimate the robot position inside the room retrieved in the previous stage. For every room, a model has been trained with each loss function and its optimal parameters obtained in the coarse localization stage, with a training length of 5 epochs and 10000 triplet samples per epoch. Table 4 reveals the average geometric error made by the CNN. The error cannot be zero, because in order to happen that, the training and test sequences should be exactly the same. The minimum reachable error is the one that would be obtained if the image retrieved by the CNN always matches the actual closest image. In this experiment, the minimum reachable error is around 0.12 m under every lighting condition.

Geometric error (m)				
Loss Function	**Cloudy**	**Night**	**Sunny**	**Average**
TL (m=1.25)	0.257	0.281	0.468	0.335
LE (m=0.25)	0.255	0.275	0.562	0.364
LT (m=1.25)	0.240	**0.274**	0.513	0.342
SH (m=1)	**0.239**	0.275	**0.395**	**0.303**
BH (m=0.75)	0.245	0.279	0.417	0.314
CL (γ=1, m=0)	0.256	0.312	0.644	0.404
AL (α=30º)	0.260	0.300	0.471	0.344
SNN (m=2)	0.460	0.448	1.048	0.652

Table 4. Average geometric error (m) for each loss function in the hierarchical localization.

Table 4 reveals that the Semi Hard loss has output the best results in the fine step. Likewise, a similar error is obtained with the Batch Hard. The errors obtained with every triplet loss are fairly small for every lighting condition considering the size of the building, especially under cloudy and night conditions. The errors obtained under sunny conditions are larger because the mistakes committed during the coarse localization penalize the network performance in this stage. In this case, the siamese CNN has had a fairly worse performance compared to the triplet CNNs, despite its good results in the coarse step.

5.2.2. Global localization

To address the global localization problem, an exhaustive study of the influence of the loss function and its parameters in the performance of the CNN has been performed, as in the case of the coarse step of the hierarchical localization. A model has been trained for each loss function and parameters setting, with a training length of 5 epochs and 50000 triplet samples per epoch. Table 5 reveals the average geometric error for each loss function.

From Table 5 it can be noticed that the loss function that has obtained the best results is the Batch Hard, followed by the Angular and the Semi Hard losses. The performance of the siamese CNN has dropped substantially in the

Geometric error (m)				
Loss Function	**Cloudy**	**Night**	**Sunny**	**Average**
TL (m=1)	0.303	0.324	0.633	0.420
LE (m=0.25)	0.304	0.313	0.642	0.420
LT (m=1)	0.298	0.292	0.543	0.378
SH (m=1.25)	0.286	0.305	0.497	0.363
BH (m=1)	**0.250**	0.282	**0.492**	**0.341**
CL (γ=1, m=0.25)	0.344	0.379	0.825	0.516
AL (α=30º)	0.262	**0.275**	0.513	0.350
SNN (m=2)	0.899	0.817	2.387	1.368

Table 5. Average geometric error (m) for each loss function in the global localization and optimal parameters.

global method. This happens because this architecture has struggled more in the image retrieval task than in the room classification. The difference with the hierarchical localization is that, in the hierarchical method, the good performance of the model trained for the coarse step has helped to reduce the mistakes between rooms and therefore to reduce the geometric error.

In order to compare both localization approaches directly, Table 6 shows the average geometric error committed with each loss function in the two methods. Besides, Figure 5 contains maps with the predictions of the CNN that had the best results: Semi Hard (m=1), for the hierarchical localization and Batch Hard (m=1), for the global localization. The blue points represent the visual map, whilst the rest of points represent the test images, which are colored differently depending on the quality of the prediction. If the test image is located correctly among the K=1 nearest neighbors, the point is colored green, whereas if the image is not located among the K=20 nearest neighbors, the point is colored red. Meanwhile, if a mistake in the room prediction is made, the color will be brown. Intermediate values will take yellow or orange colors. The lines connect every test image with the retrieved image from the visual model.

Table 6 shows that the average errors tend to increase in the global localization with every loss function, comparing to the hierarchical localization. This is logical, since in this case the CNN tries to locate each image inside the entire map in a single step, and this environment is prone to visual aliasing,

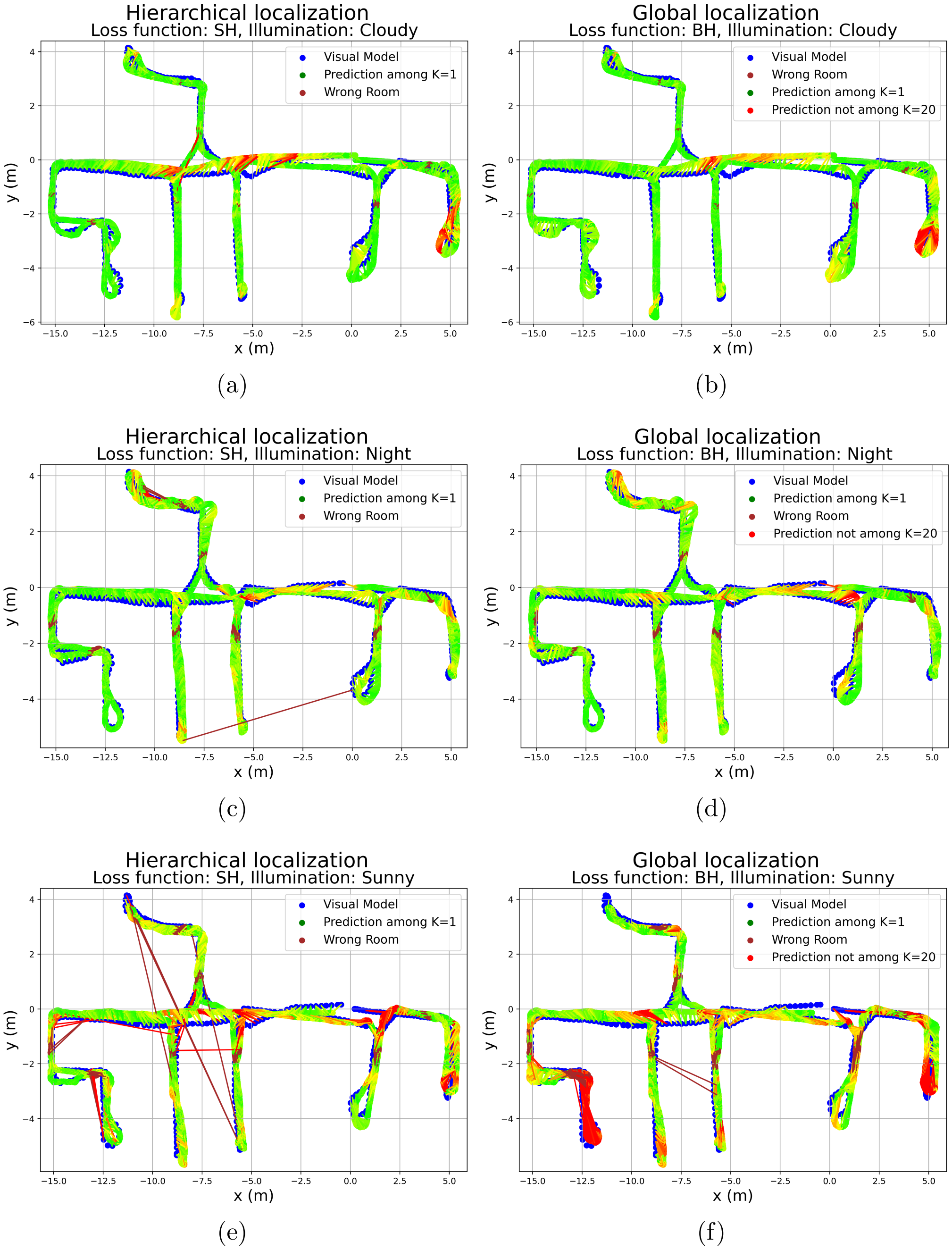
Geometric Error (m)		
Loss Function	**Hierarchical Loc.**	**Global Loc.**
TL	0.335	0.420
LE	0.364	0.420
LT	0.342	0.378
SH	**0.303**	0.363
BH	0.314	**0.341**
CL	0.404	0.516
AL	0.344	0.350
SNN	0.652	1.368

Table 6. Average geometric error (m) for each loss function in the hierarchical localization and in the global localization.

so the hierarchical process is able to better retain the features that characterize and distinguish every room. In general terms, comparing to hierarchical localization, the performance is slightly worse for cloudy and night, but the error is larger for sunny (Tables 4 and 5).

By comparing the maps in Figure 5, it can be appreciated that the number of errors between non-connected rooms is very small in both approaches (less than 0.1% of all the test images). However, in the global localization, the number of errors increases substantially in some rooms under sunny conditions, concretely the toilet and the printer area. In both methods, the errors take place more frequently in the transition zones or in junctions, where the images contain visual information from different rooms.

If an overall comparison of all the triplet loss functions is made, the Lazy Triplet and its variants, i.e. the Semi Hard and the Batch Hard, had a great performance in the localization task. This can be explained because of the fact that these losses penalize the largest errors of the CNN of every training batch, which has allowed to perform a more challenging training process. Also, the Angular loss has output good results in both localization approaches.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 5. Maps with the predictions of the trained models for every test image in the hierarchical localization (a) Cloudy, c) Night, e) Sunny) and in the global localization (b) Cloudy, d) Night, f) Sunny).

In order to study the feasibility of the implementation of the proposed algorithms in real time, Table 7 includes the localization time of each method. The localization time can be defined as the time gap since an image is captured until the coordinates of the image are obtained. All the experiments have been carried out by means of an NVIDIA GeForce GTX 3090 GPU with 24GB of RAM.

	Hierarchical localization	**Global Localization**
Localization time (ms)	6.83	3.82

Table 7. Localization time (ms) for each localization method.

This table reveals that the hierarchical localization time is larger than global localization time. This difference is due to the fact that in the global localization, the image coordinates are retrieved in a single step by one network, whereas in the hierarchical localization two steps are needed. In this case, a single network is used during the coarse localization step and one network for every room is used during the fine localization step. However, in both cases the time is sufficiently low as to enable the robot to perform localization with a reasonable frequency.

5.3. Experiment 2. Analysis of the robustness against dynamic effects.

In Experiment 1, two localization approaches have been compared under different lighting conditions. Moreover, other challenging effects such as the presence of people, changes in the position of objects or occlusions, e.g. the structure that supports the mirror, are implicit in the images. Now, the hierarchical method, which is the approach that has output the best results, is tested against certain effects that appear frequently in images captured by a mobile robot: occlusions, noise and motion blur. Figure 6 shows examples of such effects applied on a panoramic image. In this experiment, no training is conducted, since the models trained for Experiment 1 are directly evaluated with the same test sets (cloudy, night and sunny), but with the different effects applied on the images. To do that, the model of the loss function that has output the best accuracy in the coarse step (Batch Hard) is employed along with the models of the best loss function in the fine step (Semi Hard). As in Experiment 1, a comparison with siamese architectures is performed.

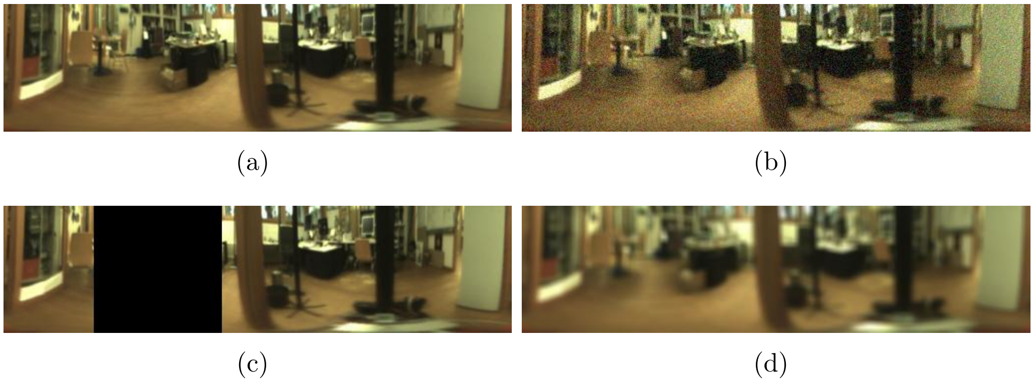


Figure 6. Example of a panoramic image (a) with no effects applied on it, (b) with a Gaussian noise of $\sigma = 20$, (c) with 128 columns occluded and (d) with motion blur with a mask size of 7 pixels.

5.3.1. Noise effect

Frequently, when the robot captures an image, it contains a certain noise. In order to apply this effect on the images, a Gaussian noise of different magnitude (σ) has been added to each image of the test sequences and also to the ones that compose the visual model. Table 8 shows the average error committed with siamese and triplet networks for different values of σ.

Noise	SNN (Contrastive Loss)		Triplet Network (BH + SH)	
Sigma	Coarse Loc. Accuracy (%)	Fine Loc. Error (m)	Coarse Loc. Accuracy (%)	Fine Loc. Error (m)
0	96.86	0.652	97.37	0.325
5	96.65	0.707	97.06	0.328
10	95.51	0.896	96.64	0.393
15	93.69	1.024	95.29	0.499
20	91.92	1.176	88.27	1.104

Table 8. Average geometric error committed by each localization approach with Gaussian noise applied to the images.

5.3.2. Occlusions effect

Another common situation that a mobile robot can face is that some parts of the scene are occluded by other objects of the scene. In this experiment, these occlusions are generated by setting some columns of the panoramic image to black. Table 9 contains the average error committed when the number of occluded columns is modified.

Occlusions	SNN (Contrastive Loss)		Triplet Network (BH + SH)	
N. columns	Coarse Loc. Accuracy (%)	Fine Loc. Error (m)	Coarse Loc. Accuracy (%)	Fine Loc. Error (m)
0	96.86	0.652	97.37	0.325
16	96.16	0.777	96.98	0.373
32	95.80	0.859	96.50	0.432
64	95.28	0.940	96.15	0.501
128	92.76	1.256	93.68	0.806

Table 9. Average geometric error committed by each localization approach with occlusions applied to the images.

5.3.3. Motion blur effect

This effects happens when images are captured while the robot is moving at a certain speed or when it is turning. When this happens, the objects of the scene appear blurred. To implement this effect, a convolution mask is applied to the image, so that each pixel value is affected by the values of the neighboring pixels along the horizontal axis. To increase the magnitude of the effect, the size of the mask is modified. Table 10 shows the influence of this effect on the performance of the CNN.

Blur	SNN (Contrastive Loss)		Triplet Network (BH + SH)	
Mask size	Coarse Loc. Accuracy (%)	Fine Loc. Error (m)	Coarse Loc. Accuracy (%)	Fine Loc. Error (m)
0	96.86	0.652	97.37	0.325
1	96.87	0.651	97.38	0.325
3	97.01	0.666	97.38	0.350
5	95.61	0.788	96.46	0.462
7	87.25	1.383	92.62	1.028

Table 10. Average geometric error committed by each localization approach with motion blur applied to the images.

Tables 8, 9 and 10 reveal that, as expected, the performance of the proposed hierarchical method is affected by the visual effects studied in this experiment. However, the errors do not increase substantially until the magnitude of such effects is very pronounced. It must be stated that the magnitude of the effects applied in this experiment is much larger that some variable effects that already appear in the images (see Figure 6). The occlusion caused by the structure that supports the mirror always appears in the images and it is fairly noticeable, but the results obtained in Experiment 1 demonstrate that it does not have a big negative impact on the performance of the localization approach.

If the results obtained with siamese and triplet networks are compared, it can be clearly appreciated that the triplet approach is more robust against these effects. This fact supports the evidence that triplet architectures are more suitable to tackle visual localization in adverse conditions.

5.4. Experiment 3. Hierarchical localization in different environments simultaneously

The objective of this experiment is to prove the ability of triplet networks to address localization in larger and different environments, and to explore the limits of the proposal. To do that, the same procedure has been followed than in Experiment 1, with the difference that in this case, three image sets corresponding to different environments have been jointly used: Freiburg, Saarbrücken A and Saarbrücken B. Therefore, the CNNs to be trained are facing a more challenging task, and the ability of the approach to generalize to different environments is assessed. A triplet architecture is retrained for the four triplet loss functions that had the best performance in Experiment 1 (Lazy Triplet, Semi Hard, Batch Hard and Angular loss) with the optimal parameters obtained in Experiment 1. Besides, only the localization method that had the best performance in Experiment 1 was tackled, i.e. the hierarchical approach.

a) Coarse localization

In this stage, the ability of the CNN to retrieve the correct room is studied. The procedure followed in this step has been the same as in the coarse step in Experiment 1, but with the difference that in this case, more rooms are taken into account, and many of them are of the same nature, which can aggravate the visual aliasing problem. In Table 11 the accuracy obtained with each loss function is shown.

Room Retrieval Accuracy (%)				
Loss Function	**Cloudy**	**Night**	**Sunny**	**Average**
LT (m=1.25)	**97.90**	92.84	**93.89**	**94.88**
SH (m=1)	97.85	**95.30**	91.48	**94.88**
BH (m=0.75)	**97.90**	91.10	92.79	93.93
AL (α=30º)	92.08	89.93	88.18	90.06

Table 11. Room retrieval accuracy with each loss function in the coarse localization.

As observed in Table 11, the accuracy is lower than in Experiment 1, since now the CNN must distinguish among 22 rooms instead of 9, and the visual aliasing problem is more present. It should be noted that, in certain cases, the accuracy obtained under sunny conditions is higher than under night, because the sunny test set only contains images captured in two different environments (Freiburg and Saarbrücken B) and only 14 rooms are considered. This is due to the fact that the dataset does not contain any images captured under sunny conditions in Saarbrücken A. In this stage, the Semi Hard and the Lazy Triplet losses have output the best results. The Batch Hard has had a worse performance under night conditions. Besides, Angular loss has suffered more overfitting than the rest of losses.

b) Fine localization

In this stage, a network per room is trained in order to determine the robot coordinates inside the room retrieved in the coarse localization, as in Experiment 1. For every room, a network has been trained with each loss function. Table 12 shows the average geometric error made by the network.

From these graphics we can observe that the error committed is larger than in Experiment 1. This is logical, since the CNNs have had a worse performance in the room retrieval as the number of rooms increased. In this case, the error committed under cloudy conditions is substantially lower than under night or sunny. However, the errors are reasonable, given the difficulty

Geometric error (m)				
Loss Function	**Cloudy**	**Night**	**Sunny**	**Average**
LT (m=1.25)	0.379	1.431	0.517	0.775
SH (m=1)	0.379	0.848	**0.504**	**0.577**
BH (m=0.75)	**0.328**	1.306	0.771	0.802
AL (α=30º)	0.461	**0.766**	0.841	0.689

Table 12. Average geometric error made with each loss function in the fine localization.

of the task. Despite the fact that the number of errors between rooms has increased, each of the fine-step models have fairly maintained their precision when the room of the image is retrieved correctly. Consequently, the hierarchical method has permitted that the errors do not increase substantially.

In this case, the Semi Hard loss has had the best overall performance. On the one hand, the smallest error under cloudy conditions have been obtained with the Batch Hard, but the CNNs trained with this loss have struggled more under night and sunny conditions. On the other hand, the Angular loss has output the highest error under cloudy conditions, but it has had less overfitting to the training conditions that the rest of losses.

5.5. Comparison with the state of the art

Finally, the proposed method is compared with similar approaches that used global-appearance descriptors obtained with analytical techniques, such as gist or HOG ([5]), and with CNNs models that have been adapted and retrained in order to tackle hierarchical localization in the COLD-Freiburg indoor environment. All the experiments have been conducted under the same conditions: all of them used a training set composed of images captured under cloudy conditions and tested their models under three different lighting conditions (cloudy, night and sunny), and no data augmentation is performed. Besides, some of them have employed the sequence of the visual model also as a test set. This causes that the trajectory of the robot is exactly the same and in both cases the exact same lighting condition is present. Therefore, the error that they have obtained under cloudy conditions is smaller than the minimum error that can be reached with our method (0.12 m), and these values have not been included in Table 13 because they cannot be comparable to our experimental setup in which the visual model and the test images are in all cases different.

Table 13 shows the geometric error made in the hierarchical localization with each method. In the case of our method, the Semi Hard loss is considered, as it is the loss function that has output the best results in the hierarchical approach.

Geometric Error (m)				
Approaches	Cloudy	Night	Sunny	Average
HOG [9]	-	1.065	0.884	-
Gist [9]	-	0.451	0.820	-
AlexNet [9]	-	0.321	0.517	-
SVM + K-NN [14]	-	0.527	0.773	-
AlexNet [28]	0.293	0.288	0.690	0.424
EfficientNet [30]	0.240	0.330	0.440	0.337
ConvNext-L [48]	**0.220**	**0.260**	0.830	0.437
Triplet VGG-16 (ours)	0.239	0.275	**0.395**	**0.303**

Table 13. Comparison with other methods in the complete hierarchical localization.

Table 13 shows that the error obtained with our approach is similar to the smallest error obtained under cloudy and night conditions, obtained by means of the ConvNext-Large architecture, which was already very small considering the dimensions of the environment. Besides, this model [48] has a substantially heavier architecture than VGG-16 model, which can be a real impediment for a real-time implementation.

However, under the sunny condition, which differs the most from the training condition (cloudy), our method has obtained the smallest error by far. This means that the model trained with a triplet architecture has suffered less overfitting than the models trained with a single CNN or a siamese architecture. Consequently, the smallest average error in the hierarchical approach has been obtained with our approach, as it has arrived to a good balance between the three lighting conditions.

6. Conclusions

Throughout this manuscript, two different localization approaches have been tackled (hierarchical and global) in indoor environments, with the use of triplet neural networks along with panoramic images. The VGG-16 model has been adapted and retrained to embed the panoramic images into global-appearance descriptors. As the experimental section has proved, one of the main advantages of using triplet networks in visual localization is that they can be trained with a reduced set of images captured under a specific lighting condition (cloudy in this case) and with no need of data augmentation. In general terms, the results show that the networks can be configured and trained to present a good ability to generalize to different lighting conditions (which were not seen during the training), proving to have a balanced behavior under different conditions.

Moreover, Experiment 1 has addressed an exhaustive comparative evaluation of several triplet loss functions for every localization stage. In general terms, all the loss functions tend to present a high accuracy under cloudy conditions. However, the results obtained with each loss function differ when the network is facing a more challenging task such as global localization (due to the fact that the environment used in the tests is prone to visual aliasing) or lighting conditions that the network has never seen during the training process. The loss functions that have shown the best performance are the variants of the Lazy Triplet loss, i.e. the Semi Hard loss and the Batch Hard loss. This can be explained as this loss functions penalize the biggest errors of the batch, which has permitted to conduct a more demanding training process, and subsequently has enhanced the performance of the trained models. The Angular loss, which employs the cosine similarity metric, has also performed well both in the hierarchical and in the global localization.

Besides, Experiment 2 analyzes the robustness of the trained triplet networks against certain effects that appear frequently in images captured by mobile robots, which can compromise the performance in localization. The effects studied in this experiment are Gaussian noise, occlusions and motion blur. In every case, the performance of the trained models has been quite stable when the effects presents low to medium magnitude. The geometrical error exceeds 1 m only when the magnitude of the effects is very pronounced, which is not usual in real operation conditions.

Throughout both experiments, the results obtained with triplet networks have been compared to a siamese architecture, by following the same training and test conditions. As expected, triplet networks have outperformed the siamese network in every experiment. The siamese architecture has performed considerably well in the coarse step of the hierarchical localization, but its performance has decreased substantially in the fine step and in the global localization. Therefore, triplet networks prove to be able to cope with visual aliasing and dynamic conditions.

Furthermore, in Experiment 3 triplet architectures have been evaluated in multiple indoor environments simultaneously, which are especially prone to visual aliasing. The experiments demonstrate that, despite the difficulty of the task, the hierarchical approach has prevented that the errors increase substantially. Therefore, the proposed approach is capable of generalizing to diverse and challenging environments, keeping a good performance.

Finally, our method has been compared with similar approaches that have addressed a hierarchical localization. Under cloudy and night conditions, our method has led to low errors, similar to those obtained with other approaches. However, the error made by our approach is significantly lower under sunny conditions, which are the conditions that differ the most from the training conditions. Consequently, our method has obtained the smallest average error in the hierarchical localization.

All in all, triplet networks have proved to be a precise tool to address visual localization in challenging, repetitive and dynamic indoor environments. Furthermore, they have demonstrated a great robustness against lighting variations and other visual effects that are common in cameras mounted on mobile robots. In future experiments, the proposed architecture will be extended to outdoor environments, which are more challenging and show bigger changes of appearance. Furthermore, we will explore the use of more complex architectures to tackle visual localization in larger indoor environments.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The code used in the experiments is available at https://github.com/MarcosAlfaro/TripletNetworksIndoorLocalization.git. The images of the COLD database can be downloaded from their official website https://www.cas.kth.se/COLD/.

Acknowledgments

This research work is part of the project TED2021-130901B-I00 funded by MCIN/AEI/10.13039/501100011033 and by the European Union "NextGenerationEU"/PRTR, and of the project PROMETEO/2021/075 funded by Generalitat Valenciana.

References

[1] Amorós, F., Payá, L., Mayol-Cuevas, W., Jiménez, L. M., & Reinoso, O. (2020). Holistic descriptors of omnidirectional color images and their performance in estimation of position and orientation. IEEE Access, 8, 81822-81848, https://doi.org/10.1109/ACCESS.2020.2990996.

[2] Kneip, L., Furgale, P., & Siegwart, R. (2013, May). Using multi-camera systems in robotics: Efficient solutions to the NPNP problem. In 2013 IEEE International Conference on Robotics and Automation (pp. 3770-3776). IEEE, https://doi.org/10.1109/ICRA.2013.6631107.

[3] Lin, H. Y., Chung, Y. C., & Wang, M. L. (2021). Self-localization of mobile robots using a single catadioptric camera with line feature extraction. Sensors, 21(14), 4719, https://doi.org/10.3390/s21144719.

[4] Flores, M., Valiente, D., Gil, A., Reinoso, O., & Payá, L. (2022). Efficient probability-oriented feature matching using wide field-of-view imaging. Engineering Applications of Artificial Intelligence, 107, 104539, https://doi.org/https://doi.org/10.1016/j.engappai.2021.104539.

[5] Payá, L., Peidró, A., Amorós, F., Valiente, D., & Reinoso, O. (2018). Modeling environments hierarchically with omnidirectional imaging and global-appearance descriptors. Remote sensing, 10(4), 522, https://doi.org/10.3390/rs10040522.

[6] Murillo, A. C., Guerrero, J. J., & Sagues, C. (2007, April). Surf features for efficient robot localization with omnidirectional images. In Proceedings 2007 IEEE International Conference on Robotics and Automation (pp. 3901-3907). IEEE, https://doi.org/10.1109/ROBOT.2007.364077.

[7] Se, S., Lowe, D. G., & Little, J. J. (2005). Vision-based global localization and mapping for mobile robots. IEEE Transactions on robotics, 21(3), 364-375, https://doi.org/10.1109/TRO.2004.839228.

[8] Nilwong, S., Hossain, D., Kaneko, S. I., & Capi, G. (2019). Deep learning-based landmark detection for mobile robot outdoor localization. Machines, 7(2), 25, https://doi.org/10.3390/machines7020025.

[9] Cebollada, S., Payá, L., Jiang, X., & Reinoso, O. (2022). Development and use of a convolutional neural network for hierarchical appearance-based localization. Artificial Intelligence Review, 1-28, https://doi.org/10.1007/s10462-021-10076-2.

[10] Yin, H., Wang, Y., Ding, X., Tang, L., Huang, S., & Xiong, R. (2019). 3D LiDAR-based global localization using siamese neural network. IEEE Transactions on Intelligent Transportation Systems, 21(4), 1380-1392, https://doi.org/10.1109/TITS.2019.2905046.

[11] Liu, Y., & Huang, C. (2017). Scene classification via triplet networks. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 11(1), 220-237, https://doi.org/10.1109/JSTARS.2017.2761800.

[12] Hermans, A., Beyer, L., & Leibe, B. (2017). In defense of the triplet loss for person re-identification. arXiv preprint arXiv:1703.07737, https://doi.org/10.48550/arXiv.1703.07737.

[13] Xiao, L., Wang, J., Qiu, X., Rong, Z., and Zou, X. (2019). Dynamic-SLAM: Semantic monocular visual localization and mapping based on deep learning in dynamic environment. Robotics and Autonomous Systems, 117, 1-16, https://doi.org/https://doi.org/10.1016/j.robot.2019.03.012.

[14] Cebollada, S., Payá, L., Peidró, A., Mayol, W., & Reinoso, O. (2023). Environment modeling and localization from datasets of omnidirectional

scenes using machine learning techniques. Neural Computing and Applications, 1-22, https://doi.org/10.1007/s00521-023-08515-y.

[15] Zhang, X., Su, Y., & Zhu, X. (2017, September). Loop closure detection for visual SLAM systems using convolutional neural network. In 2017 23rd International Conference on Automation and Computing (ICAC) (pp. 1-6). IEEE, https://doi.org/10.23919/IConAC.2017.8082072.

[16] Kallasi, F., Rizzini, D. L., & Caselli, S. (2016). Fast keypoint features from laser scanner for robot localization and mapping. IEEE Robotics and Automation Letters, 1(1), 176-183, https://doi.org/10.1109/LRA.2016.2517210.

[17] Li, D., Shi, X., Long, Q., Liu, S., Yang, W., Wang, F., ... & Qiao, F. (2020, October). DXSLAM: A robust and efficient visual SLAM system with deep features. In 2020 IEEE/RSJ International conference on intelligent robots and systems (IROS) (pp. 4958-4965). IEEE, https://doi.org/10.1109/IROS45743.2020.9340907.

[18] Su, Z., Zhou, X., Cheng, T., Zhang, H., Xu, B., & Chen, W. (2017, December). Global localization of a mobile robot using LiDAR and visual features. In 2017 IEEE international conference on robotics and biomimetics (ROBIO) (pp. 2377-2383). IEEE, https://doi.org/10.1109/ROBIO.2017.8324775.

[19] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278-2324, https://doi.org/10.1109/5.726791.

[20] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, https://doi.org/10.48550/arXiv.1409.1556.

[21] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1-9), https://doi.org/10.48550/arXiv.1409.4842.

[22] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. Communications of the ACM, 60(6), 84-90, https://doi.org/10.1145/3065386.

[23] Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009, June). ImageNet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition (pp. 248-255). Ieee, https://doi.org/10.1109/CVPR.2009.5206848.

[24] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... & Houlsby, N. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929, https://doi.org/10.48550/arXiv.2010.11929.

[25] Komorowski, J. (2021). Minkloc3d: Point cloud based large-scale place recognition. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (pp. 1790-1799), https://doi.org/10.48550/arXiv.2011.04530.

[26] Foroughi, F., Chen, Z., & Wang, J. (2021). A CNN-based system for mobile robot navigation in indoor environments via visual localization with a small dataset. World Electric Vehicle Journal, 12(3), 134, https://doi.org/10.3390/wevj12030134.

[27] Xu, S., Chou, W., & Dong, H. (2019). A robust indoor localization system integrating visual localization aided by CNN-based image retrieval with Monte Carlo localization. Sensors, 19(2), 249, https://doi.org/10.3390/s19020249.

[28] Cabrera, J. J., Cebollada, S., Flores, M., Reinoso, Ó., & Payá, L. (2022). Training, optimization and validation of a CNN for room retrieval and description of omnidirectional images. SN Computer Science, 3(4), 271, https://doi.org/10.1007/s42979-022-01127-8.

[29] Chen, Y., Chen, R., Liu, M., Xiao, A., Wu, D., & Zhao, S. (2018). Indoor visual positioning aided by CNN-based image retrieval: training-free, 3D modeling-free. Sensors, 18(8), 2692, https://doi.org/10.3390/s18082692.

[30] Rostkowska, M., & Skrzypczyński, P. (2023). Optimizing Appearance-Based Localization with Catadioptric Cameras: Small-Footprint Models for Real-Time Inference on Edge Devices. Sensors, 23(14), 6485, https://doi.org/10.3390/s23146485.

[31] Ballesta, M., Paya, L., Cebollada, S., Reinoso, O., & Murcia, F. (2021). A CNN regression approach to mobile robot localization using omnidirectional images. Applied Sciences, 11(16), 7521, https://doi.org/10.3390/app11167521.

[32] Wozniak, P., Afrisal, H., Esparza, R. G., & Kwolek, B. (2018). Scene recognition for indoor localization of mobile robots using deep CNN. In Computer Vision and Graphics: International Conference, ICCVG 2018, Warsaw, Poland, September 17-19, 2018, Proceedings (pp. 137-147). Springer International Publishing, https://doi.org/10.1007/978-3-030-00692-1_13.

[33] Leyva-Vallina, M., Strisciuglio, N., & Petkov, N. (2021). Generalized contrastive optimization of siamese networks for place recognition. arXiv preprint arXiv:2103.06638, https://doi.org/10.48550/arXiv.2103.06638.

[34] Qiu, K., Ai, Y., Tian, B., Wang, B., & Cao, D. (2018, June). Siamese-ResNet: Implementing loop closure detection based on Siamese network. In 2018 IEEE Intelligent Vehicles Symposium (IV) (pp. 716-721). IEEE, https://doi.org/10.1109/IVS.2018.8500465.

[35] Oliveira, G. L., Radwan, N., Burgard, W., & Brox, T. (2020). Topometric localization with deep learning. In Robotics Research: The 18th International Symposium ISRR (pp. 505-520). Springer International Publishing, https://doi.org/10.48550/arXiv.1706.08775.

[36] Chen, X., Läbe, T., Milioto, A., Röhling, T., Behley, J., & Stachniss, C. (2022). OverlapNet: A siamese network for computing LiDAR scan similarity with applications to loop closing and localization. Autonomous Robots, 1-21, https://doi.org/10.1007/s10514-021-09999-0.

[37] Arandjelovic, R., Gronat, P., Torii, A., Pajdla, T., & Sivic, J. (2016). NetVLAD: CNN architecture for weakly supervised place recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 5297-5307), https://doi.org/10.48550/arXiv.1511.07247.

[38] Yu, J., Zhu, C., Zhang, J., Huang, Q., & Tao, D. (2019). Spatial pyramid-enhanced NetVLAD with weighted triplet loss for place recog-

nition. IEEE transactions on neural networks and learning systems, 31(2), 661-674, https://doi.org/10.1109/TNNLS.2019.2908982.

[39] Lopez-Antequera, M., Gomez-Ojeda, R., Petkov, N., & Gonzalez-Jimenez, J. (2017). Appearance-invariant place recognition by discriminatively training a convolutional neural network. Pattern Recognition Letters, 92, 89-95, https://doi.org/10.1016/j.patrec.2017.04.017.

[40] Olid, D., Fácil, J. M., & Civera, J. (2018). Single-view place recognition under seasonal changes. arXiv preprint arXiv:1808.06516, https://doi.org/10.48550/arXiv.1808.06516.

[41] Cheng, D., Gong, Y., Zhou, S., Wang, J., & Zheng, N. (2016). Person re-identification by multi-channel parts-based CNN with improved triplet loss function. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1335-1344), https://doi.org/10.1109/CVPR.2016.149.

[42] Liu, L., Li, H., & Dai, Y. (2019). Stochastic attraction-repulsion embedding for large scale image localization. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 2570-2579), https://doi.org/10.48550/arXiv.1808.08779.

[43] Kim, S., Seo, M., Laptev, I., Cho, M., & Kwak, S. (2019). Deep metric learning beyond binary supervision. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 2288-2297), https://doi.org/10.48550/arXiv.1904.09626.

[44] Sun, Y., Cheng, C., Zhang, Y., Zhang, C., Zheng, L., Wang, Z., & Wei, Y. (2020). Circle loss: A unified perspective of pair similarity optimization. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 6398-6407), https://doi.org/10.48550/arXiv.2002.10857.

[45] Wang, J., Zhou, F., Wen, S., Liu, X., & Lin, Y. (2017). Deep metric learning with angular loss. In Proceedings of the IEEE international conference on computer vision (pp. 2593-2601), https://doi.org/10.48550/arXiv.1708.01682.

[46] Cabrera, J. J., Román, V., Gil, A., Reinoso, O., & Payá, L. (2024). An experimental evaluation of Siamese Neural Networks for robot localization using omnidirectional imaging in indoor environments. Artificial Intelligence Review, 57(8), 198, https://doi.org/10.1007/s10462-024-10840-0.

[47] Pronobis, A., & Caputo, B. (2009). COLD: The CoSy localization database. The International Journal of Robotics Research, 28(5), 588-594, https://doi.org/10.1177/0278364909103912.

[48] Cabrera, J. J., Cebollada, S., Céspedes, O., Cebollada, S., Reinoso, O. & Payá, L. (2024). An evaluation of CNN models and data augmentation techniques in hierarchical localization of mobile robots. Evolving Systems, 1–13, Springer, https://doi.org/10.1007/s12530-024-09604-6.