

Generalized Contrastive Optimization of Siamese Networks for Place Recognition

María Leyva-Vallina, Nicola Strisciuglio, and Nicolai Petkov

Abstract—Visual place recognition is a challenging task in computer vision and a key component of camera-based localization and navigation systems. Recently, Convolutional Neural Networks (CNNs) achieved high results and good generalization capabilities. They are usually trained using pairs or triplets of images labeled as either *similar* or *dissimilar*, in a binary fashion. In practice, the similarity between two images is not binary, but rather continuous. Furthermore, training these CNNs is computationally complex and involves costly pair and triplet mining strategies.

We propose a Generalized Contrastive loss (GCL) function that relies on image similarity as a continuous measure, and use it to train a siamese CNN. Furthermore, we propose three techniques for automatic annotation of image pairs with labels indicating their degree of similarity, and deploy them to re-annotate the MSLS, TB-Places, and 7Scenes datasets.

We demonstrate that siamese CNNs trained using the GCL function and the improved annotations consistently outperform their binary counterparts. Our models trained on MSLS outperform the state-of-the-art methods, including NetVLAD, and generalize well on the Pittsburgh, TokyoTM and Tokyo 24/7 datasets. Furthermore, training a siamese network using the GCL function does not require complex pair mining. We release the source code at https://github.com/marialeyvallina/generalized_contrastive_loss.

Index Terms—contrastive learning, image retrieval, siamese networks, visual place recognition

1 INTRODUCTION

Visual place recognition has received large interest from researchers in computer vision, machine learning and information retrieval. It consists of, given a query image, seeking an image depicting a similar scene in a map or database. Possible instances of this problem are the retrieval of an image containing a specific distinctive landmark [1], [2] or the recognition of a previously visited place [3], [4]. Algorithms for image retrieval and place recognition are deployed in and are a key component of many visual localization [5] and Simultaneous Localization and Mapping (SLAM) systems [6]. In these applications, effectively retrieving similar images from a map set is of utmost importance to facilitate a reliable estimation of the camera pose in an environment and perform the subsequent 3D reconstruction tasks. Methods for place recognition have been tested on data recorded in different environments, each including different challenges. Indoor scenes can be subject to heavy viewpoint variations [7]. Outdoor environments include drastic changes in illumination [4], weather [6], seasonal [8] as well as long-term variations [9].

Recently, approaches based on Convolutional Neural Networks (CNNs) achieved very good results for place recognition [10], [11]. Their great generalization capabilities allow to deploy them as feature extractors, yielding acceptable results [12], [13], [14], [15]. It is, however, with end-to-end trained methods that the best performance results were achieved. These methods usually optimize a triplet loss function, using triplets made of a query image, a positive

and a negative match [16], or a query image, a set of similar images and a set of dissimilar images [4]. Other approaches rely on a contrastive loss function [17], [18], and are trained using as input similar or dissimilar image pairs.

All these methods are built upon considering image similarity as a binary option, i.e. two images either depict the same scene/object or not. In practice, image similarity is not a binary attribute, but rather a continuous one. A pair of images may be 100% similar if they are the same image, or not similar at all, if they do not share any common features. In between, one should consider a range where two images share some features, while not being exactly the same. We illustrate some examples in Fig. 1. Existing methods [4], [17], [19] are trained using binary similarity labels, ignoring the range of partial similarity that exists in practice. Since deep learning methods rely on the amount and quality of the training data, noise or errors in the labels may cause the models to not be effectively trained, lowering their performance. More complete and precise label information can contribute to increasing the performance of existing methods. This improvement can be achieved by re-annotating a dataset [20], by implementing a multi-task architecture [21], [22], [23], [24], or by also using semantic [25], [26], [27], [28], geometric [29], [30], [31], or sequential [6] information to train the models.

In this work, we build on the concept of partial image similarity for image retrieval and place recognition, and propose a generalized formulation of the Contrastive Loss function. We deploy it as objective function to train siamese network architectures. It enforces the representation of similar images to be closer in the latent space, while pushing apart the representations of dissimilar images proportionally to their annotated degree of similarity.

We propose three strategies to automatically re-annotate

• M. Leyva-Vallina and N. Petkov are with the Intelligent Systems Group of the Bernoulli Institute, at the University of Groningen, the Netherlands.
 • N. Strisciuglio is with the Faculty of Electrical Engineering, Mathematics and Computer Science of the University of Twente, the Netherlands.
 • E-mail: m.leyva.vallina@rug.nl, n.strisciuglio@utwente.nl

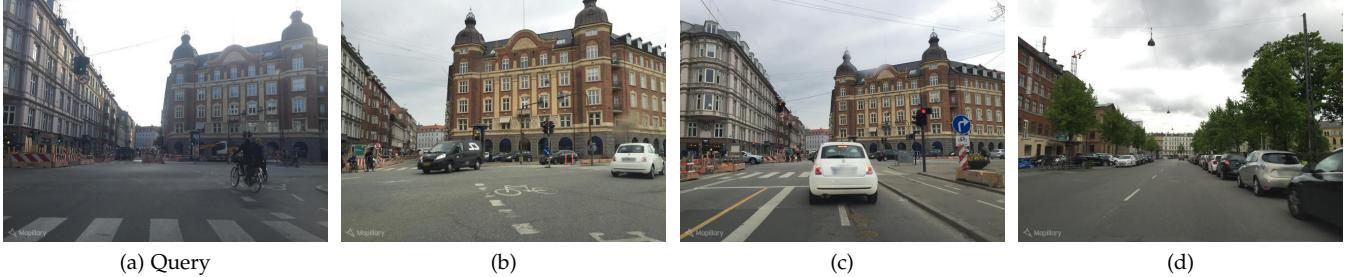


Fig. 1. Examples from MSLS dataset of a query image (a) and three matches with different degrees of similarity. A very close match is (b) with 86% of overlap, while (c) is a borderline case with 52% of commonality, and (d) is a negative match, with no features in common with the query.

existing datasets for place recognition with labels that indicate the degree of similarity of image pairs: Weak 2D Field-of-View overlap, Strong 2D Field-of-View overlap and 3D Field-of-View overlap. These strategies rely on different available information recorded in the original data sets. The Weak 2D Field-of-View overlap labeling strategy exploits the GPS position and compass angle information associated to the images to estimate the 2D Field-of-View (FoV) of the camera that has taken the concerned picture. The intersection area between the FoV of two cameras determines the ground truth degree of similarity of the image pair. We use this to re-label the Mapillary Street Level Sequences dataset [9], which contains images recorded in urban, suburban and countryside environments in 30 cities all over the world. The Strong 2D Field-of-View overlap also computes the intersection of the FoV of two cameras, but it is calculated based on the 6DOF camera pose information. We use it to re-annotate the TB-Places dataset, where the 6DOF ground truth camera pose was recorded using a laser tracker and an IMU sensor [32]. The 3D Field-of-View overlap strategy computes the intersection of the 3D FoV of two cameras, based on the 3D reconstruction of the scene. We apply it to re-label the 7Scenes dataset [7], which contains images and 3D models of indoor scenes.

We use the proposed automatic annotations to train a siamese CNN architecture by optimizing the Generalized Contrastive Loss and we compare it to existing approaches that rely on binary image similarity ground truth. The method that we propose consists of a fully convolutional backbone with a simple pooling operation on top (i.e. Global Average or GeM pooling). We do not incorporate complex pair mining strategies into the training process of our method. We only ensure that each batch contains approximately the same amount of positive and negative pairs, each composed of two images: a query and a positive or negative counterpart. We carry out experiments on the MSLS [9] (for which we report new state-of-the-art results), Pittsburgh [4], [5], TokyoTM [4], Tokyo 24/7 [33], TB-Places [32] and 7Scenes [7] benchmark data sets.

The contributions of this work are four-fold:

- 1) a novel Generalized Contrastive Loss (GCL) function, which relies on graded image similarity annotations to train siamese networks;
- 2) three automatic data annotation techniques, based on GPS coordinates and compass angle, 6DOF camera pose and 3D reconstruction information;

- 3) a naïve pair mining strategy that does not involve memory-intensive computations on the GPUs;
- 4) new state-of-the-art results on the MSLS data set.

The paper is organized as follows. We discuss related works in Section 2. We introduce the Generalized Contrastive Loss function, the model architecture and mining strategy in Section 3, and provide details about our automatic labeling techniques in Section 4. We explain the experimental framework and evaluation procedure in Section 5. We present and discuss the results that we achieved in Section 6. Finally, we draw conclusions in Section 7.

2 RELATED WORKS

2.1 Place recognition

Traditional place recognition algorithms rely on local image features and holistic representations, such as Fisher Vectors [34], [35], Bag of Words [1], [5], [36] and VLAD [37], [38], or on exploiting image sequences [6], [8] or panoramas [5].

Deep learning methods for place recognition achieved state-of-the-art results [10], [11], initially using pre-trained CNNs as feature extractors [12], [13], [14], [39], [40], and subsequently as end-to-end trainable models. Several methods optimize a contrastive loss function [17], with pairs of positive and negative samples. In [41], the authors demonstrate the benefits of using a ground truth based on soft assignments to positive and negative classes. Other methods are trained by optimizing a triplet loss function, with image tuples consisting of a query, a positive match and a negative match [16]. In the case of NetVLAD [4], the authors train a network using a triplet loss function with a query, a set of potential positive matches and a set of definite negative matches. They optimize a weak triplet loss function that enforces higher distances between the query and all the negative matches than between the query and any potential positive. PointNetVLAD [42] exploits 3D information by combining PointNet [43] and NetVLAD for large-scale place recognition. In [19], the authors use spatial pyramid pooling to encode spatial and structural information into the NetVLAD descriptors, and a weighted triplet loss to maximize the distances between the negative pairs while minimizing the distance between the positive pairs.

Several challenges of visual place recognition in different environments were studied and several benchmark datasets were publicly released. The KITTI [44], Oxford Robot-Car [45], Pittsburgh250K [33] and TokyoTM [4] datasets

contain images taken in urban environments. Tokyo 24/7 [4] and Aachen [46] also provide images taken at day and night that depict drastic changes in illumination conditions. The Alderley dataset also includes weather variations [6]. The Mapillary Street Level Sequences dataset includes images taken with different cameras in 30 cities in six continents [9]. It contains changes in viewpoint, weather, illumination and long-term variations, as well as urban, suburban and countryside environments. To the best of our knowledge, it is the largest dataset for long-term place recognition. The Norland dataset includes images taken in countryside environments and contains variations of weather conditions and season [8]. The TB-Places dataset was designed for place recognition in garden environments [18], [32]. It includes challenging long-term variation, and viewpoint changes, as well as very repetitive textures and scenes dominated by objects of green color appearance.

2.2 Data annotation

CNNs, which constitute the current state-of-the-art in most computer vision tasks, are trained on large-scale annotated datasets, such as ImageNet [47] and Places [48] for object and scene classification, COCO [49] and Pascal VOC [50] for object detection, and Pittsburgh250k [33] and Mapillary Street Level Sequences [9] for visual place recognition. The quality, diversity, and richness of the ground truth annotations have a direct impact on the performance of the trained models and contribute to the strength of their generalization capabilities [51]. For instance, in [52] it was shown that models trained using inaccurate ground truth data can lead to gender and racial biased predictions, as they fail to learn the actual distribution of the real-world problem.

Thus, a correct, informative and complete definition of the ground truth labels is of utmost importance to learn more accurate and unbiased models. Previous works on data curation include dataset expansion, e.g. Places2 [48], and manual re-annotations, e.g. Google Landmarks [20]. Another trend in the literature is Multi-Task learning, which consists of learning two (or more) related but yet diverse tasks simultaneously, and benefiting from their commonalities. This generally results in higher prediction accuracy for the individual tasks, compared to learning them separately [53]. Recent works include the combination of semantic and geometric tasks for pose regression [21] or joint learning of semantic segmentation and image intrinsic decomposition [23]. Although its efficacy, Multi-Task learning is limited by the scarcity of adequate datasets, which are required to provide samples with multiple labels, for each considered task.

Another option is to incorporate available complementary information in the training process, such as for SeqSLAM [6], [8] where image sequences were used for Simultaneous Localization and Mapping. In [25], the authors used semantic sub-category information to identify food-related environments in egocentric photo sequences, and [27] proposed a similar approach for semantic segmentation. The authors of [26] incorporated semantics to predict human poses and [28] they used semantics for 3D reconstruction. In [29], the authors presented PoseNet, an end-to-end pose regressor trained relying only on camera pose information.

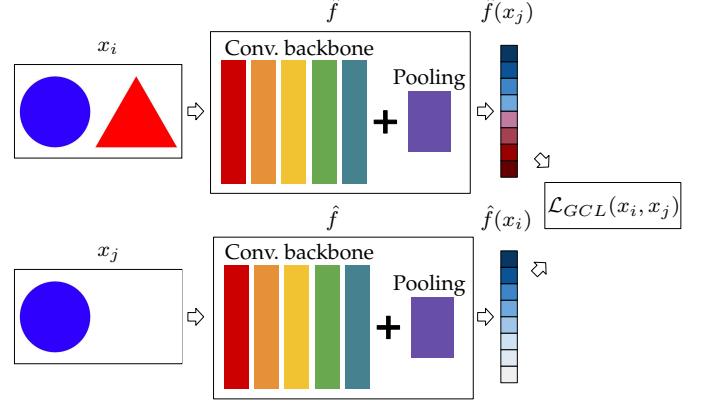


Fig. 2. Sketch of a siamese architecture where x_i and x_j are the input images, \hat{f} represents the convolutional backbone with a pooling layer and $\hat{f}(x_i)$ and $\hat{f}(x_j)$ are the representations of the input images. They are used as input for the optimization of the Generalized Contrastive Loss function $\mathcal{L}_{GCL}(x_i, x_j)$.

They expanded their work in [30] by incorporating further geometric information into the loss function. In [31], the authors successfully incorporated pose information to learn object segmentation, which contributed to an improvement of the camera localization accuracy.

3 METHODOLOGY

In this section, we propose the Generalized Contrastive Loss function and show how to use it to train a siamese network architecture. Furthermore, we describe the place recognition pipeline and how we apply it to search and retrieve similar images from a reference map.

3.1 Fully convolutional backbone and pooling

We deploy a siamese architecture with a fully convolutional backbone. Given an input image $x \in R^{w_n \times h_n \times d_n}$, we consider a convolutional network that computes a representation $f(x) \in R^{w_m \times h_m \times d_m}$, where w_m and h_m are the width and height of the last convolutional activation map, and d_m corresponds to the number of kernels of the last convolutional layer. The output tensor of the last convolutional layer of the network is fed to a global pooling layer, which computes the image representation $\hat{f}(x) \in R^{d_m}$. In this work, we experiment with versions of the network with a Global Average Pooling and a GeM pooling layer [17]. However, one can explore the use of other global pooling strategies, in accordance with the application at hand. Given a pair of images (x_i, x_j) , their representation is computed as $(\hat{f}(x_i), \hat{f}(x_j))$. We illustrate the architecture in Fig. 2. Our methodology can be applied using any convolutional architecture as backbone: in this work we use DenseNet [54] and ResNet [55].

3.2 Generalized Contrastive Loss

Siamese architectures are used to learn representations that disentangle the (dis-)similarity of pairs of input images. They consist of two identical networks (i.e. they share their

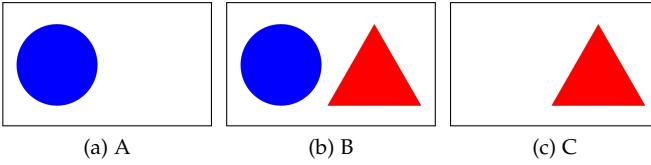


Fig. 3. Example dataset consisting of three images. (A, B) and (B, C) are considered similar ($y = 1$) because they share part of their content, while (A, C) are dissimilar ($y = 0$).

weights) that process input image pairs. Applications include signature verification [56], face identification [57], visual place recognition [4], [16], [18] and image retrieval [17]. The training of a siamese architecture is generally carried out by optimizing a Contrastive Loss function.

3.2.1 Contrastive Loss function

Let us consider two input images x_i and x_j , and their representations $\hat{f}(x_i)$ and $\hat{f}(x_j)$. We define the distance between the representations of the input images x_i and x_j in the latent space as $d(x_i, x_j) = \|\hat{f}(x_i) - \hat{f}(x_j)\|_2$. The Contrastive Loss function \mathcal{L}_{CL} is defined as:

$$\mathcal{L}_{CL}(x_i, x_j) = \begin{cases} \frac{1}{2}d(x_i, x_j)^2, & \text{if } y = 1 \\ \frac{1}{2} \max(\tau - d(x_i, x_j), 0)^2, & \text{if } y = 0 \end{cases} \quad (1)$$

where τ is the margin, i.e. a threshold for the descriptor distance above which a pair of images is not considered as depicting the same place. The margin τ is an hyperparameter defined by the user. The ground truth label y is such that 1 indicates a pair of similar images, and 0 a not-similar pair of images. Similarity, however, is not a binary attribute, and defining it as such may cause the trained models to produce unreliable predictions.

Let us consider an example dataset consisting of three images, as depicted in Fig. 3. The pairs (A, B) and (B, C) are labeled as similar, while the pair (A, C) is labeled as not-similar. The aim of a siamese network is to learn a function \hat{f} that maps similar images to points in a latent space that are close together, and non-similar image pairs to points with larger distance in the latent space.

The optimization of the Contrastive Loss function aims at minimizing the distance between the representations of the similar image pairs (A, B) and (B, C), such that:

$$\|\hat{f}(A) - \hat{f}(B)\|_2 \approx 0 \wedge \|\hat{f}(B) - \hat{f}(C)\|_2 \approx 0$$

that corresponds to:

$$\hat{f}(A) \approx \hat{f}(B) \wedge \hat{f}(B) \approx \hat{f}(C)$$

Therefore, by the transitive property of equality, the Euclidean distance of the representation of A and C is also ensured to be close in the latent space:

$$[(\hat{f}(A) \approx \hat{f}(B)) \wedge (\hat{f}(B) \approx \hat{f}(C))] \rightarrow \hat{f}(C) \approx \hat{f}(A) \quad (2)$$

In the example we considered, however, (A, C) is labeled as dissimilar. The result of Eq. 2 is in contrast with the optimization ensured by Eq. 1, where the distance of the representation of dissimilar images is maximized. This inconsistency in the training process is due to the fact that

the Contrastive Loss function does not take into account the partial similarity between the input pairs.

3.2.2 Generalized Contrastive Loss function

We propose a generalized formulation of the Contrastive Loss that relies on a definition of continuous, rather than binary, similarity. We define the Generalized Contrastive Loss function \mathcal{L}_{GCL} as:

$$\mathcal{L}_{GCL}(x_i, x_j) = \psi_{i,j} \cdot \frac{1}{2}d(x_i, x_j)^2 + (1 - \psi_{i,j}) \cdot \frac{1}{2} \max(\tau - d(x_i, x_j), 0)^2 \quad (3)$$

where x_i and x_j denote the two input images, $\hat{f}(x)$ is the representation of the input x and $\psi_{i,j}$ is the ground truth degree of similarity of x_i and x_j , a continuous value ranging from 0 (completely dissimilar) to 1 (exactly similar). By minimising a Generalized Contrastive Loss function, the distance of each image pair in the latent space is optimized proportionally to the corresponding degree of similarity.

3.2.3 Gradient of the Generalized Contrastive Loss

In the training phase, the loss function is minimized by gradient descent optimization and the weights of the network are updated by backpropagation. In the case of the Contrastive Loss function, the gradient is:

$$\nabla \mathcal{L}_{CL}(x_i, x_j) = \begin{cases} d(x_i, x_j), & \text{if } y = 1 \\ \min(d(x_i, x_j) - \tau, 0), & \text{if } y = 0 \end{cases} \quad (4)$$

It is worth noting that the min function ensures that the gradient is computed for the negative pairs whose distance in the latent space is lower than the margin τ (see the supplementary materials for the derivation of the gradient). The weights of the model are updated in order to minimize the distance between similar images, and to maximize the distance between dissimilar ones. A partial similarity is therefore not considered, and the training process may incur in learning inconsistent representations.

In contrast, the proposed Generalized Contrastive loss function takes into account the degree of similarity between the input images during the training process. Its gradient is:

$$\nabla \mathcal{L}_{GCL}(x_i, x_j) = \begin{cases} d(x_i, x_j) + \tau(\psi_{i,j} - 1), & \text{if } d(x_i, x_j) < \tau \\ d(x_i, x_j) \cdot \psi_{i,j}, & \text{if } d(x_i, x_j) \geq \tau \end{cases} \quad (5)$$

It is worth pointing out that the magnitude of the gradient of the Generalized Contrastive Loss function is modulated by the ground truth degree of similarity of the input image pairs, $\psi_{i,j}$. In the case the distance $d(x_i, x_j)$ is larger than the margin τ , the weights of the network are updated proportionally to $\psi_{i,j}$. More interestingly, when the distance $d(x_i, x_j) < \tau$, the Generalized Contrastive Loss function has an intrinsic regularization effect of the learned latent space. The network weights are indeed updated so that the vector representations of the training images x_i and x_j are moved closer in the latent space if their ground truth degree of similarity is $\psi_{i,j} > 1 - \frac{d(x_i, x_j)}{\tau}$, otherwise they are pushed away. For the boundary cases $\psi_{i,j} = 0$ (completely dissimilar input images) and $\psi_{i,j} = 1$ (same exact input images), the gradient is the same as in Eq. 4.

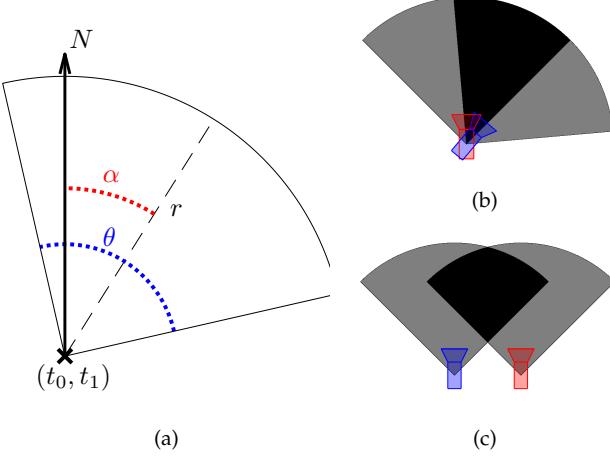


Fig. 4. (a) 2D Field-of-View representation with angle θ and radius r . The point (t_0, t_1) is the camera location in the environment, and α is the camera orientation in the form of a compass angle with respect to the north N . A (b) soft positive match: two cameras (with $\theta = 90^\circ$ and $r = 50m$) in the same position but with orientations 40° apart. A (c) soft negative example: two cameras cameras (with $\theta = 90^\circ$ and $r = 50m$) located 25m apart but with the same orientation.

3.3 Image search and place recognition

Let us consider a set X of map images, for which the location in the environment of the camera that took them is known. We are presented with a set Y of query images taken from unknown positions. In order to effectively localize the camera that took the query images in the environment, we are required to retrieve similar images to the query images from the map set. We compute a feature representation of the map images $\hat{f}(x) \forall x \in X$, and of the query images $\hat{f}(y) \forall y \in Y$ using our network. For a given query representation $\hat{f}(y)$, image retrieval is performed by an exhaustive nearest neighbor search among the representations of the map set $\hat{f}(x) \forall x \in X$, retrieving the map images with the closest representations in the latent space.

It is worth pointing out that in this paper we do not address the camera localization and pose estimation problems. We focus on training networks that are able to compute an effective representation for better retrieval results, which can further benefit the localization task.

4 TRAINING DATA AND AUTOMATIC LABELLING

The optimization of the Generalized Contrastive Loss function relies on the ground truth similarity of image pairs defined in the range $[0, 1]$. In this section we present three approaches that we designed to automatically label the degree of similarity of image pairs. The use of each of the proposed techniques depends on the data available together with the images in the concerned datasets. We introduce two measures of image similarity based on the 2D Field-of-View overlap: weakly labeled, relying on GPS data, and strongly labeled, relying on the 6DOF camera pose information associated to the images. Additionally, we present the 3D Field-of-View overlap, estimated from 3D reconstruction and 6DOF camera pose.

4.1 2D Field-of-View overlap

We estimate the similarity of two images by approximating a measure of the overlap of their two-dimensional Field-of-View (FoV) in the horizontal plane. Let us consider a camera with a FoV defined by the angle θ and radius r , positioned in an environment according to a 2D translation vector (t_0, t_1) with respect to the origin of the reference system. The camera is oriented at an angle α with respect to the north direction of the reference system. We define the 2D FoV as the sector of the circle denoted by the center (t_0, t_1) and radius r enclosed in the angle range delimited by $[\alpha - \frac{\theta}{2}, \alpha + \frac{\theta}{2}]$ (see Fig. 4a). The 2D Field-of-View overlap between two images corresponds to the intersection-over-union (IoU) of their FoVs. We consider as positives the image pairs with determined ground truth FoV overlap higher than 50%, and the rest as negatives. More specifically, the negative pairs with a similarity higher than 0% are soft negatives, while the pairs with a similarity degree equal to 0% are hard negatives.

4.1.1 Weak 2D Field-of-View overlap

We propose the *weak 2D Field-of-View overlap* to estimate the similarity of pairs of images for which GPS position labels (UTM format) and compass angle information are available. We consider the UTM data as the translation vector (t_0, t_1) and the compass angle as the orientation α necessary to estimate the 2D FoV of the cameras.

We use the weak 2D Field-of-View overlap to re-annotate the Mapillary Street Level Sequences (MSLS) dataset [9], a large-scale outdoor visual place recognition dataset. It contains images of 30 cities across 6 continents that depict urban, suburban and countryside environments, and are annotated with the UTM position and the compass angle. The FoV angle of the cameras and the intrinsics are not provided. We thus estimate the value of the FoV angle θ according to the following reasoning. The authors of MSLS define a positive match when the retrieved map image falls within 25m and 40° from the query. We define a similarity measure that satisfies those constraints. Image pairs taken at locations that are closer than 25m and with orientation differences lower than 40° are expected to have a similarity higher than 50%. Moreover, the borderline cases with distances close to 25m and/or orientation difference near to 40° should have a similarity close to 50%. Hence, we define $r = 25m \times 2 = 50m$ and we estimate a θ that gives approximately a 50% FoV overlap for the borderline cases, i.e. 0m and 40° and 25m and 0° . For the first case, the optimal θ corresponds to 80° , and for the second case to 102° . We settle for a value in the middle and define $\theta = 90^\circ$, which gives 55.63% and 45.01% FoV overlaps for each case, as shown in Fig. 4b and 4c. We display some examples of the similarity ground truth for MSLS in Fig. 5.

4.1.2 Strong 2D Field-of-View overlap

We propose the *strong 2D Field-of-View overlap* to estimate the similarity of image pairs in the case 6DOF camera pose information is available together with the recorded images in the dataset. The translation vector (t_0, t_1) and orientation angle α are extracted from the pose vector.

This is the case of the TB-Places dataset [32], for visual place recognition in garden environments, created for the

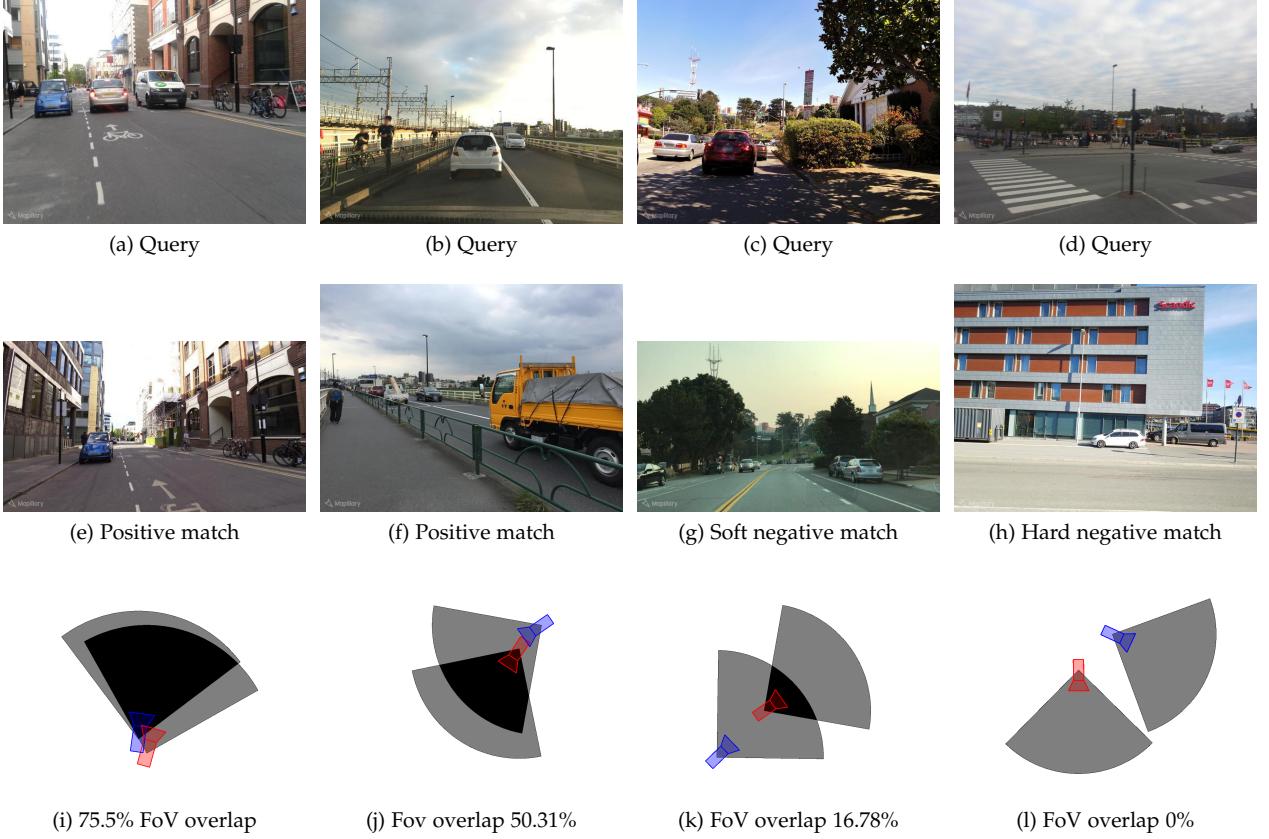


Fig. 5. Example image pairs from the MSLS dataset. The first row shows the query images, the second row shows the corresponding matches from the map set, and the third row shows the estimated 2D FoV overlap. The query image is associated with the red camera, while the map image with the blue camera. The first column shows a positive match with 75.5% FoV overlap, and many visual features in common. The second column shows a borderline pair: the two images have FoV overlap of 50.31% and some features in common. The third column shows a soft negative match, where the two images have FoV overlap of 16.78%. The fourth column shows a hard negative match, where the two images are taken by cameras looking in opposite directions, and the FoV overlap is 0%.

Trimbot2020 project [58]. It contains images taken in an experimental garden over three years and includes variations in illumination, season and viewpoint. Each image comes with a 6DOF camera pose, which allows us to estimate a very precise 2D FoV. According to the original paper of the TB-places dataset, we set the FoV angle of the cameras as $\theta = 90^\circ$ and the radius as $r = 3.5m$. We thus estimate the 2D Field-of-View overlap and use it to re-label the pairs of images contained in the dataset. We show some examples of image pairs and their weak 2D FoV overlap in Fig. 6.

4.2 3D Field-of-View overlap

When a 3D reconstruction of a concerned environment is available, we propose to estimate the degree of similarity of image pairs by computing the *3D Field-of-View overlap*. We project a given image with an associated 6DOF camera pose onto the reconstructed pointcloud of the environment. We select the subset of 3D points that falls within the boundaries of the image as the image 3D FoV. For an image pair, we compute their 3D FoV overlap as the intersection-over-union (IoU) of the sets of 3D points associated with the two images. We consider the computed 3D FoV overlap as a measure of the degree of similarity of a pair of images.

We use the *3D Field-of-View overlap* to re-annotate the 7Scenes dataset [7], an indoor localization benchmark, that

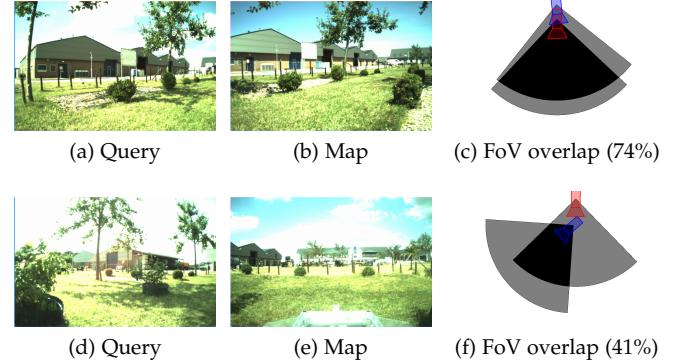


Fig. 6. Example image pairs from the TB-Places dataset. The first row shows a positive pair with FoV overlap of 74%. The second row depicts a soft negative pair with FoV overlap equal to 41%. The red camera corresponds to the query image, while the blue one to the map image.

contains RGBD images taken in seven environments. Each image has an associated 6DOF pose, and a 3D reconstruction of each scene is provided. We show some examples of the 3D FoV overlap in Fig. 7. We display the 3D FoV associated to the query image in red, the 3D FoV of the map image in blue and their overlap in magenta.

4.3 Selection of training pairs

Training a model for visual place recognition is usually formulated as a binary classification problem. The aim is to determine whether two images depict a similar (class 1) or dissimilar (class 2) place. To train binary classifiers, it is generally desirable to have balanced datasets, to ensure that the two classes are equally weighted. This is often not the case for visual place recognition, where the dissimilar image pairs significantly outnumber the positive pairs. Moreover, when training a siamese architecture, it is necessary to form batches with meaningful image pairs or triplets. For instance, if the training process consists of pairs that are too easy, the learning process might stall, and the weights of the network not be updated. If the pairs are too difficult, the training dynamics can become unstable [59]. Hence, the selection of image pairs is a crucial element of the training.

Existing approaches use complex image pair mining strategies. In the case of NetVLAD [4], the authors use a Hard Negative Mining strategy. For each query image, they select a set of potential positive map images and a set with its 10 hardest negative map images. The best combination of positive and negative samples is selected by means of their contribution to a Triplet Ranking Loss function. This implies that all the images need to be forwarded to the network to compute the value of the loss function, although only three contribute to the learning. In [9], the authors deploy a similar strategy, but they select only the top-5 hardest negatives. This mining technique requires that, for each triplet, many image latent representations are computed. Even with a caching strategy, bigger backbones like VGG cannot be trained in their entirety [4] and large training batches do not fit in the memory of a regular GPU. In [9], indeed, the authors use a batch size of 4 triplets.

We argue that a better curated and more informative ground truth can replace these complicated mining techniques. We use our graded similarity annotations, presented in Sections 4.1 and 4.2 to select the image pairs that compose a training batch. For our training process, we ensure that each training batch contains a balanced amount of positive and negative pairs. In the case of the negative pairs, we also ensure that half of them are soft negatives (i.e. their annotated similarity is higher than 0). This means that each selected image pair consist of a query and a match image, and we do not require complicated mining strategies or to compute any latent representations to form the pairs. Hence, we are able to train our models with a batch size of 64.

5 EXPERIMENTAL FRAMEWORK

5.1 Datasets

5.1.1 Mapillary Street Level Sequences

Mapillary Street Level Sequences (MSLS) [9] is a large scale place recognition dataset that contains images taken in 30 different cities across six continents. It includes challenging variations of camera viewpoint, season, time and illumination. Moreover, the images have been taken with different cameras. The training set contains over 500k query images and 900k map images, taken in 22 cities. The validation set consists of 19k map images and 11k query images, taken in two cities, and the test set has 39k map images and 27k query images, taken in six different cities.

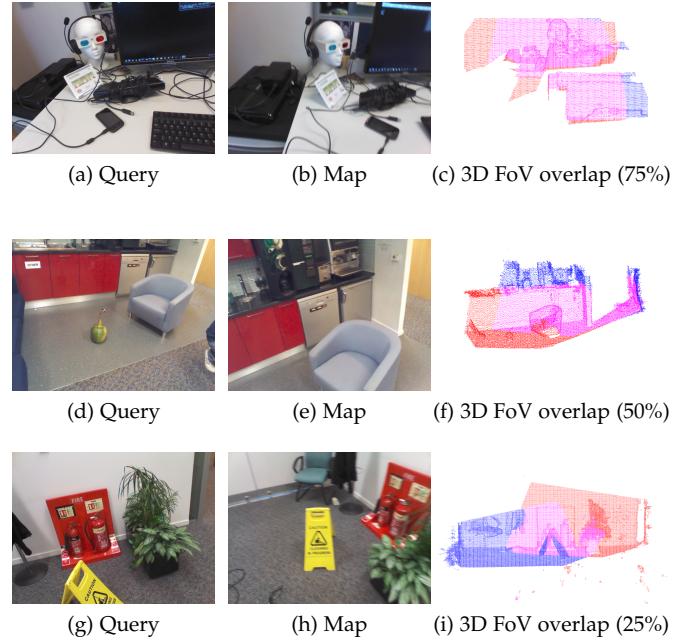


Fig. 7. 3D Field-of-View overlap examples from the 7Scenes dataset. The first row shows a positive pair with 3D FoV overlap of 75%. The second row depicts a borderline pair with 50% 3D FoV overlap. The last row shows a soft negative image pair with 25% 3D FoV overlap. The red pointcloud corresponds to the query 3D FoV, the blue one is the map, and the magenta represents the overlap between them.

We evaluate our models on the MSLS validation set and on the MSLS test set. For the former, we use the evaluation script published by the authors of the dataset. For the latter, since the ground truth is not publicly available, we submit our predictions to the official evaluation server. Following the protocol established by the authors of the dataset [9], two images are considered as similar if they are taken by cameras located within 25m of distance, and with less than 40° of viewpoint variation.

5.1.2 Pittsburgh

The Pittsburgh dataset contains images of urban environments gathered from Google Street View in the city of Pittsburgh, Pennsylvania, USA [5], taken in the span of several years. We test our models on the test set of the Pitts250k and Pitts30k sets. The former consists of 83k map and 8k query images, and the latter is a subset of 10k map images and 7k query images. We use it to evaluate the generalization of our models trained on MSLS.

5.1.3 Tokyo 24/7 and Tokyo Time Machine

The Tokyo 24/7 dataset consists of images taken in Tokyo, Japan, containing large variations of illumination, as they are taken during day and night [33]. The query set consists of 315 images, and the map contains 76k photos.

The TokyoTM dataset consists of images collected using the Time Machine tool on Google Street View [4] in the city of Tokyo over several years. Its validation set is divided into a map set of 49k images and a query set of 7k images. We use Tokyo 24/7 and the validation set of TokyoTM to evaluate the generalization of our models trained on the MSLS dataset.

5.1.4 TB-Places

TB-Places [18], [32] is a place recognition dataset for garden environments. It contains images taken by a rover robot in a garden at the University of Wageningen, the Netherlands, over the course of three years. The dataset was collected for the TrimBot2020 project [58]. It includes drastic viewpoint variations, as well as illumination changes. The garden is a very challenging small environment with repetitive textures.

The dataset consists of three subsets, i.e. W16, that contains 41k images taken in 2016; W17, that includes 11k images taken in 2017; and W18, that has 23k images taken in 2018. As in [18] we use the W17 subset to train our models.

We design two experiments to evaluate our models. For the first one we establish W17 as map set (11k images) and W18 as query (23k images). With this configuration we aim to test the robustness of our models w.r.t. changes between the map and the query sets. For the second experiment, we divide W18 into query (17k images) and map (6k images) to test the generalization capabilities of our models in the case both map and query sets were not used for training. In Fig. 8, we show a sketch of the trajectory that the robot covered in the TrimBot2020 garden for the recording of the reference map (blue trajectory) and query (orange trajectory) images. It is worth pointing out that the query images were taken from locations not covered by map images, thus including substantial viewpoint variations.

5.1.5 7Scenes

7Scenes is a benchmark dataset for indoor camera localization algorithms [7]. It includes 26k training images and 17k test images, taken in seven different environments. Each image has an associated ground truth 6DOF camera pose. Additionally, a 3D reconstruction of each scene is available. We use it to test our models for visual place recognition in indoor environments. For evaluation purposes, we define an image pair as a positive match if their annotated degree of similarity is higher than 50%. We use The training set as map, and the test set as query.

Additionally, we use the 7Scenes data set to test the effectiveness of the learned representations for visual place recognition to perform the image retrieval step in the visual localization pipeline of the InLoc algorithm [60].

5.2 Implementation details

We implemented our models using the PyTorch framework and trained them with an Nvidia V100 GPU card. We optimized the weights of our models with Stochastic Gradient Descent. To train the networks that use the binary Contrastive Loss function, we set an initial learning rate $l_0 = 0.01$. For those that deploy the Generalized Contrastive Loss function, we used as initial learning rate $l_0 = 0.1$. In both cases the learning rate is divided by 10 after every 25 epochs for MSLS, or every 15 epochs for TB-Places and 7Scenes. We set the margin value $\tau = 0.5$ and we use a batch size of 64. We trained the two last convolutional blocks of the backbone networks for all experiments. The necessary code to replicate the experiments is publicly available¹.

¹ github.com/marialeyvallina/generalized_contrastive_loss

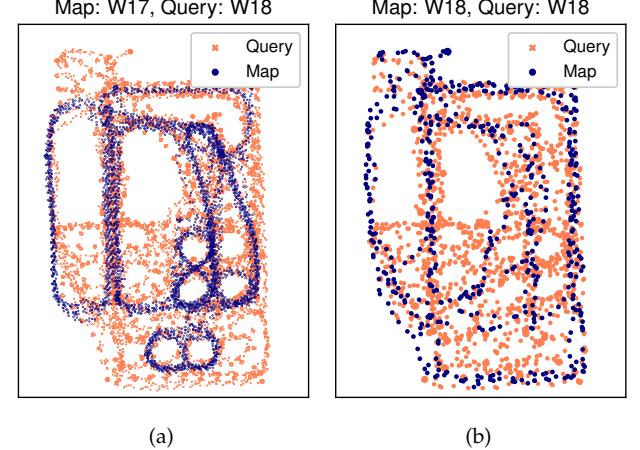


Fig. 8. Configurations of the experiments on the TB-Places dataset. (a) W17 subset is the map set, and W18 is the query. (b) We divide W18 into map and query. For visualization purposes, the trajectories have been downsampled.

5.3 Evaluation metrics

We followed a widely used place recognition evaluation protocol and considered a query as correctly identified if any of the top-k retrieved images are annotated as a positive match [4], [9], [46]. We computed the following metrics.

Top-k recall (R@k) is the percentage of queries for which we retrieved at least a correct map image among their k nearest neighbors.

Top-k Mean Average Precision (mAP@k) measures the relative amount of positive images retrieved among the k nearest neighbors of each query.

Average Precision (AP), an approximation of the area under the precision-recall curve for the classification of all the possible pairs of images. We use this measure for the 7Scenes dataset [7].

6 RESULTS AND DISCUSSION

6.1 Large scale outdoor place recognition

We trained several models on the MSLS training set and report results on the MSLS validation and test set. We evaluate the generalization capabilities of our models by also testing them on the Pittsburgh, TokyoTM and Tokyo 24/7 datasets. In the following, we use the naming *backbone-pooling-loss* for the considered models. For example, ResNet50-GeM-GCL indicates a ResNet50 backbone with GeM pooling trained using the Generalized Contrastive Loss function.

Mapillary Street Level Sequences. We carried out experiments using two backbones, i.e. ResNet50 and ResNet152, and two global pooling layers, i.e. Global Average Pooling (avg) and GeM [17]. For each possible combination of backbone network and pooling layer, we optimize the models with the binary Contrastive Loss and with our Generalized Contrastive Loss. As shown in Table 1 and in Fig. 9, the models trained with the GCL function consistently achieve better performance than their counterparts trained with the CL. For two models, namely ResNet50-GeM-GCL and ResNet152-GeM-GCL, we also

TABLE 1

Results achieved on the MSLS test set. All models are trained on the MSLS training set, unless marked with the symbols \dagger and \ddagger , which indicates models trained on the Pitts250k and SfM-120k datasets, respectively. TL stands for triplet loss, GCL corresponds to the binary Contrastive Loss and GCL is the Generalized Contrastive Loss. The $*$ symbol indicates models for which PCA whitening has been applied.

Backbone	Input size	Pooling	Dimensions	Loss	R@1	R@5	R@10	mAP@1	mAP@5	mAP@10
VGG	480x640	GeM \ddagger [9]	512	TL	28	35	49	-	-	-
		NetVLAD \dagger [9]	32768	TL	30	40	44	-	-	-
		NetVLAD [9]		TL	48	58	64	-	-	-
ResNet50	480x640	avg	2048	CL GCL	24.9 35.8	39.0 52.0	44.6 59.0	24.9 35.8	16.8 24.5	14.8 21.8
		GeM	2048	CL GCL	29.7 43.3	44.0 59.1	50.7 65.0	29.7 43.3	20.6 30	18.1 26.8
ResNet152	480x640	avg	2048	CL GCL	29.7 43.5	44.2 59.2	51.3 65.2	29.7 43.5	19.4 29.5	17.2 26.4
		GeM	2048	CL GCL	34.1 45.7	50.8 62.3	56.8 67.9	34.1 45.7	23.6 31.4	20.8 28.3
ResNet50*	480x640	GeM	2048	GCL	52.9	65.7	71.9	52.9	37.3	33.4
ResNet152*	480x640	GeM	2048	GCL	57.9	70.7	75.7	57.9	40.7	36.6

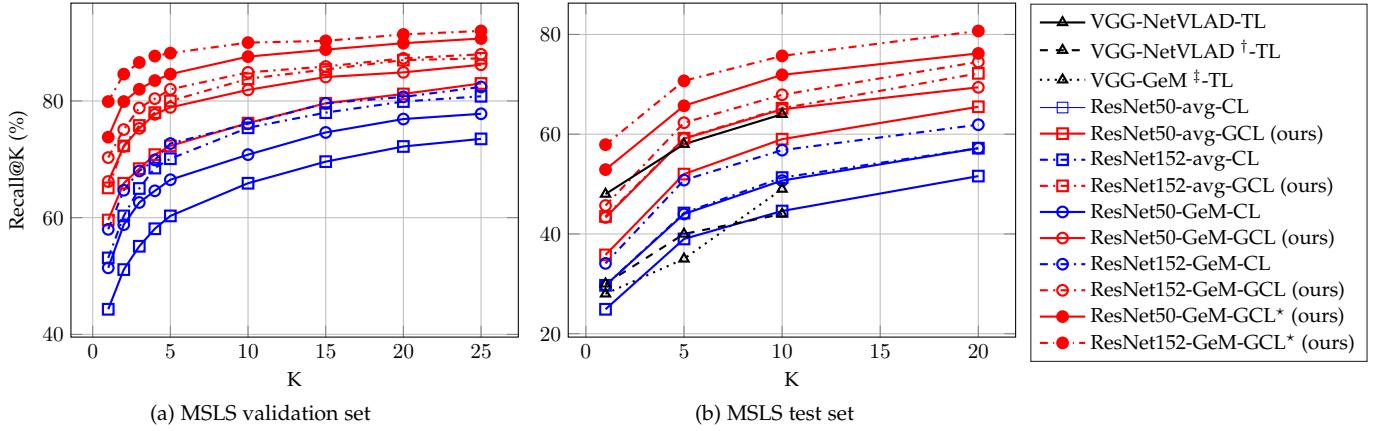


Fig. 9. Comparison of the results achieved by our methods with those of state-of-the-art methods and models trained with a binary contrastive loss for (a) MSLS validation and (b) MSLS test. The $*$ indicates the models for which whitening has been applied. TL stands for models trained with the Triplet Loss, CL for Contrastive Loss and GCL for Generalized Contrastive Loss. The results achieved by our models trained with GCL are displayed in red. The results of models trained with a binary CL are shown in blue, and the state-of-the-art results are in black. All models are trained on the MSLS training set, unless marked with the symbols \dagger and \ddagger , which indicates models trained on the Pitts250k and SfM-120k datasets, respectively.

perform a postprocessing step consisting of PCA whitening and observe that it further boosts the performance.

We compared the results of our models with those obtained by state-of-the-art methods. We achieved better results than existing approaches on the MSLS test set. We demonstrate that a simple architecture trained using the proposed GCL loss function, which can exploit a more informative ground truth, outperforms more complex architectures, like NetVLAD, which achieved a top-5 recall equal to 58% on the MSLS test set [9]. For ResNet50-GeM-GCL, we obtained a top-5 recall equal to 59.2%, similarly to the ResNet152-avg-GCL (59.1% top-5 recall). Our ResNet152-GeM-GCL model achieved a top-5 recall equal to 62.3%, outperforming NetVLAD trained on the MSLS dataset using a Triplet Loss. When whitening is applied, the performance of our models further improved, reaching a top-5 recall of 65.7% for ResNet50-GeM-GCL and 70.7% for ResNet152-GeM-GCL. It is worth pointing out that our learned representations have a size of 2048, while the

NetVLAD vectors consist of 32768 features.

Pittsburgh, Tokyo 24/7 and TokyoTM. We tested our models on the test set of Tokyo 24/7, Pittsburgh250k and Pittsburgh30k and on the validation set of TokyoTM, and report the results that we obtained in Table 2. We compare our generalization results with those achieved by the NetVLAD models trained on MSLS and Pittsburgh250k [9]. We observe that the models trained with the GCL function on the MSLS dataset generalize well to unseen datasets, in some cases better than existing methods. We recorded low generalization performance only on Tokyo 24/7, a very challenging dataset that contains drastic variations in illumination, with images taken at day and at night. Such conditions are underrepresented in the MSLS training set: only 4k query images are taken at night, out of 500k query images that compose the dataset. However, we observe that applying a whitening transform greatly enhances the generalization capabilities of our models, and it boosts the performance for all datasets. On Tokyo 24/7, we achieved

TABLE 2

Top-5 recall on the test set of Pitts30k, Pitts250k, Tokyo 24/7 and MSLS and on the validation set of TokyoTM. The † indicates models trained on Pitts250k dataset, and the * indicates the use of PCA whitening.

Model	TokyoTM	Tokyo 24/7	Pitts250k	Pitts30k	MSLS
VGG-NetVLAD [†] [9]	98	72	91	91	35
VGG-NetVLAD [9]	98	75	87	89	58
ResNet50-GeM-GCL	88.8	51.7	84.6	87.2	59.1
ResNet50-GeM-GCL*	95.9	74.0	92.5	89.7	65.7
ResNet152-GeM-GCL	90.3	51.8	84.9	87.9	62.3
ResNet152-GeM-GCL*	96.3	80.6	93.7	91.5	70.7

an improvement of 28.8% on the top-5 recall.

6.2 Small scale outdoor place recognition

We trained our method on the TB-Places dataset and show the results that we achieved in Fig. 10. We considered three different backbones, namely ResNet18, ResNet34, and DenseNet161, which learn representations of size 512, 512 and 2208, respectively. We trained them with the binary Contrastive Loss function and with our Generalized Contrastive Loss function. Furthermore, we compare them with the NetVLAD off-the-shelf model. For these experiments we use only a Global Average Pooling, so this information is omitted from the names of the models.

We report results of two experiments. For the first one, we used the training set (W17) as map, and the W18 set as query. With this experiment we tested the strength of our learned representations against significant variations between the query and the map set. For this experiment, we show the results that we obtained in Fig. 10a. For the second experiment, we divided the W18 set into map and query, to test the generalization capabilities when both map and query sets are unknown to the place recognition model (i.e. not seen during training). For this experiment, the results are displayed in Fig. 10b. For both experiments the models trained with our GCL function and the proposed similarity ground truth consistently achieve better recall than the ones trained with the binary Contrastive Loss, with the exception of the ResNet18 model. In some cases the models trained with the binary ground truth do not outperform the NetVLAD model, even though this one is not trained on the TB-Places dataset.

6.3 Indoor place recognition and localization

We reports the results that we achieved on the seven sets of the 7scenes dataset in Fig. 11. We used the ResNet18 and ResNet34 architectures as backbones with a Global Average Pooling layer and we compare them to NetVLAD off-the-shelf. Since we only use one type of pooling for this experiments this information is omitted from the model names. We achieved generally higher Recall@K results for the models trained using the GCL function, for all scenes. The cases of the *stairs* (Fig. 11b), *chess* (Fig. 11f) and *office* (Fig. 11g) scenes are particularly interesting, since with the GCL descriptors we are able to retrieve positive matches for nearly all the query images, with a top-5 recall for ResNet34-GCL of 98.1%, 98.2% and 99.7%, respectively.

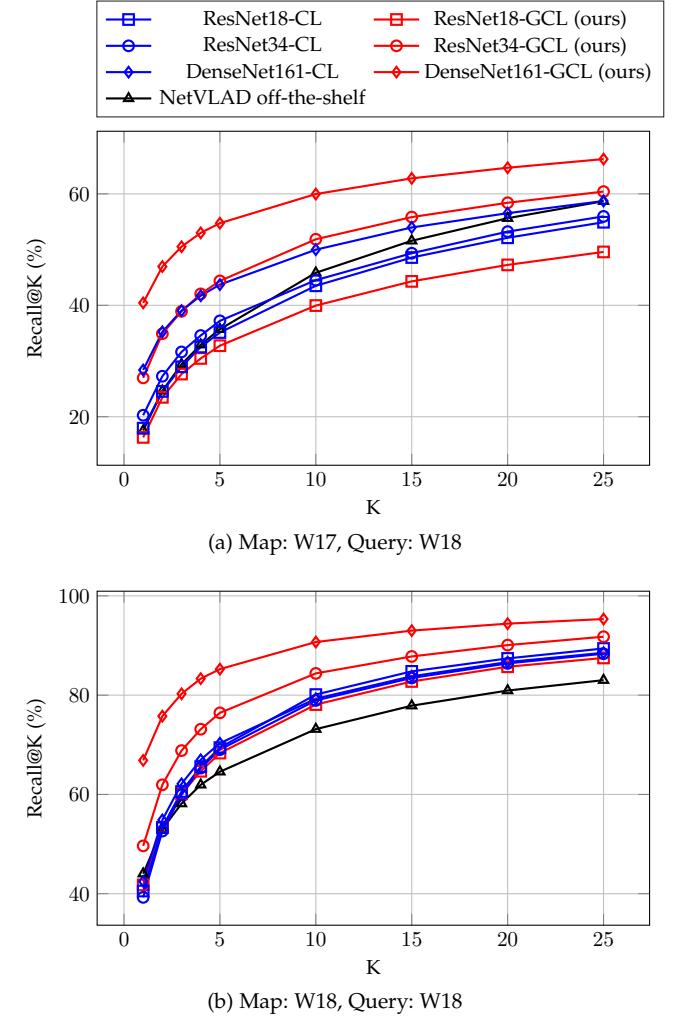


Fig. 10. Comparison of the results for the TB-Places dataset. In (a) we report the results when using W17 as map and W18 as query set. In (b), we show the top-k recall achieved when dividing W18 into map and query. The results achieved by our method are shown in red, the results of the models trained with the binary Contrastive Loss are in blue, and the results of NetVLAD are displayed in black.

Furthermore, we report the average precision results on the seven sets of the dataset in Table 3. This measure is an approximation of the area under the precision-recall curve and measures the number of pairs that can be correctly discriminated into positive and negative on the whole dataset. The models trained with the GCL function achieved higher AP than their corresponding models trained with the binary CL function. We achieved an average AP equal to 0.89 using the ResNet34-GCL model.

Finally, we tested the impact that a model trained with the GCL function and graded similarity has on the performance of a camera localization pipeline. We compared it to the case when the same model trained with the binary Contrastive Loss function is used for the image retrieval task, and report the localization error that we obtained in Table 4. For this experiment, we employed the InLoc algorithm [60], using our learned representations with the ResNet34 backbone for the retrieval step. The images retrieved using the representation learned with GCL consistently led to an improvement of the localization accuracy w.r.t. the case in

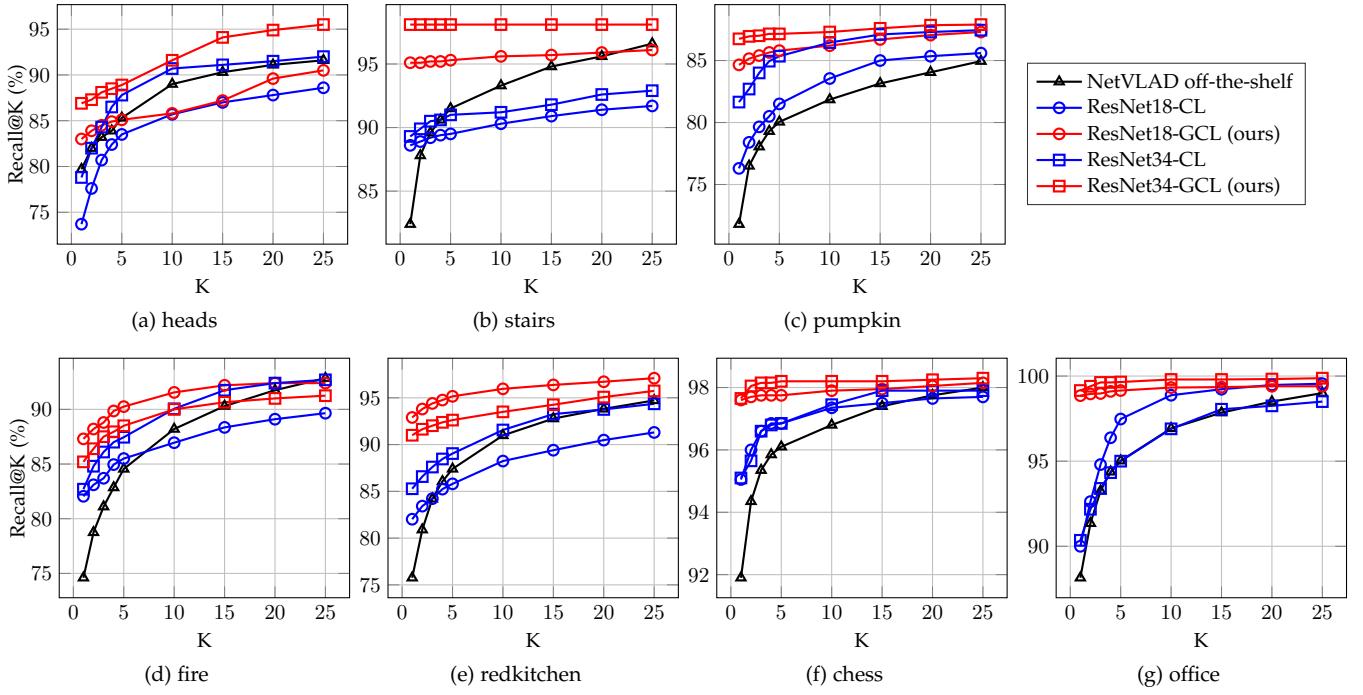


Fig. 11. Recall@K results achieved on the 7 Scenes dataset. The results of the models trained with our Generalized Contrastive Loss are shown in red, while those of the models trained with the binary Contrastive Loss are shown in blue. The Recall@K achieved by NetVLAD off-the-shelf is plotted in black.

TABLE 3

Average Precision results obtained by the networks trained with the proposed Generalized Contrastive loss function on the 7Scenes dataset, compared with those achieved by the same network architectures trained using the binary Contrastive loss function and by the NetVLAD off-the-shelf model.

Scene	NetVLAD		ResNet18		ResNet34	
	off-the-shelf		CL	GCL	CL	GCL
Heads	0.587		0.739	0.807	0.759	0.853
Stairs	0.533		0.855	0.883	0.884	0.944
Pumpkin	0.491		0.768	0.849	0.782	0.914
Fire	0.539		0.786	0.811	0.796	0.803
Redkitchen	0.439		0.790	0.876	0.782	0.902
Chess	0.645		0.943	0.964	0.945	0.974
Office	0.399		0.794	0.890	0.802	0.896
Mean	0.519		0.811	0.868	0.821	0.898

which the CL function is used. The enhanced performance of the retrieval task has a positive effect on the precision of localization algorithms.

6.4 Discussion

State-of-the-art visual place recognition pipelines rely on difficult pair and triplet mining strategies for training [4], [9]. Their goal is to compose batches of training images that substantially contribute to the learning process, evaluating their potential impact on the loss function, at the cost of GPU memory and computing time. As we discussed in Section 4.3, we opted for a naïve mining strategy, based only on the annotated graded similarity of image pairs and not on their embedded representations. We have demonstrated that a naïve mining strategy can be as effective as or better

than a more complex one, provided that the ground truth is informative enough. The results that we achieved are attributable to the use of the proposed GCL function, which allows a graded similarity ground truth to be considered. Furthermore, since our naïve strategy is much less resource demanding, it is possible to use it while training big CNN backbones in their entirety, and with batch size up to 512.

We studied the effects that training different layers of the backbones with the GCL function has on the performance of our models. We report the results that we achieved on the MSLS validation set in Table 5. The models pre-trained on the ImageNet dataset [47] achieved reasonable results. However, we obtained the highest results when the networks were further trained on the MSLS training set until Layer 3 (i.e. updating the weights of the two last convolutional blocks). Training more layers leads to overfitting.

The learned representations are of a relatively small size (2048 for ResNet50 and ResNet152) compared to NetVLAD (size of 32768). The dimensionality of the latent space representation influences the performance of image retrieval and place recognition systems. Techniques for dimensionality reduction have been explored [4], [17], usually combined with whitening, which has been demonstrated to have positive effects on image retrieval, especially in the case of CNN-learned representations [13], [40]. It can be learned end-to-end or used as a postprocessing step.

We evaluated the effect of whitening and dimensionality reduction on our learned descriptors by using PCA as postprocessing. We compute the PCA components on the descriptors of the map images and use them to project both map and query image descriptors. We perform experiments with PCA whitening with different number of dimensions, from 32 to 2048. We applied this strategy to

TABLE 4

Median translation and rotation errors (in cm and degrees, respectively) on the 7Scenes dataset using the descriptors computed using the models trained with the binary Contrastive Loss and the Generalized Contrastive Loss for retrieval and InLoc for localization.

Model	heads	stairs	pumpkin	fire	redkitchen	office	chess	Mean
ResNet34-CL	2.98cm 2.44°	36.03cm 4.28°	7.12cm 1.97°	5.87cm 2.39°	6.53cm 2.12°	5.62cm 1.85°	6.01cm 1.99°	10.02cm 2.43°
ResNet34-GCL	2.53cm 2.3°	32.13cm 3.9°	6.4cm 1.95°	5.13cm 2.11°	5.38cm 2.0°	5.16cm 1.7°	5.66cm 1.93°	8.91cm 2.27°

TABLE 5

Results achieved on the MSLS validation set by our models trained by backpropagating the gradient only partially through the network. The *last trained layer* column indicates the block of the backbone until which the gradient is backpropagated.

Last trained layer	ResNet50-GeM			ResNet152-GeM		
	R@1	R@5	R@10	R@1	R@5	R@10
Off-the-shelf	27.2	41.9	47.4	26.5	39.9	44.5
Layer4	50.1	64.6	70	40.9	56.4	63.8
Layer3	66.2	78.9	81.9	70.3	82	84.9
Layer2	60.3	74.2	77.4	65.4	77.4	80
Full	65	75.8	81.1	66.6	79.2	81.8

the ResNet50-GeM-GCL and ResNet152-GeM-GCL models (trained on MSLS) and experimented with them on the test set of Tokyo 24/7 and Pittsburgh and on the validation set of TokyoTM and MSLS. We observe that PCA whitening further improves the performance of the image representations learned by our models and boosts their generalization capabilities. As shown in Fig. 12, the whitened descriptors with higher dimensionality achieve the best results on the considered datasets. When using the whitening transform combined with PCA dimensionality reduction down to 128 dimensions of the latent space, our models achieved results comparable to those of models with 2048-dimensional latent space and no whitening. The whitening is also effective on the 2048-dimensional latent space model, which outperforms its non-whitened counterpart on all the considered datasets. We observed up to a 28.8% of improvement in the case of Tokyo 24/7 and 12.8% on the MSLS validation set.

7 CONCLUSIONS

We presented a novel Generalized Contrastive Loss function to effectively train siamese networks for image retrieval and visual place recognition. For this purpose we defined continuous measures of the degree of image similarity. We designed three techniques for the estimation of the degree of similarity of pairs of images, based on geometric information such as GPS and compass angle, 6DOF camera pose, and scene 3D reconstruction. We deployed them to re-annotate the MSLS, TB-Places and 7Scenes datasets, providing image pairs with graded similarity, rather than binary labels. The re-annotation process is automatic, thus not requiring any human intervention.

The CNN architectures that we trained consist of a fully convolutional backbone and a global pooling layer (i.e. global average pooling or GeM). The networks that we optimized using the proposed Generalized Contrastive loss function and the graded similarity label sets consistently achieved higher results (top-5 recall up to 18% higher) than their counterparts trained with a binary Contrastive

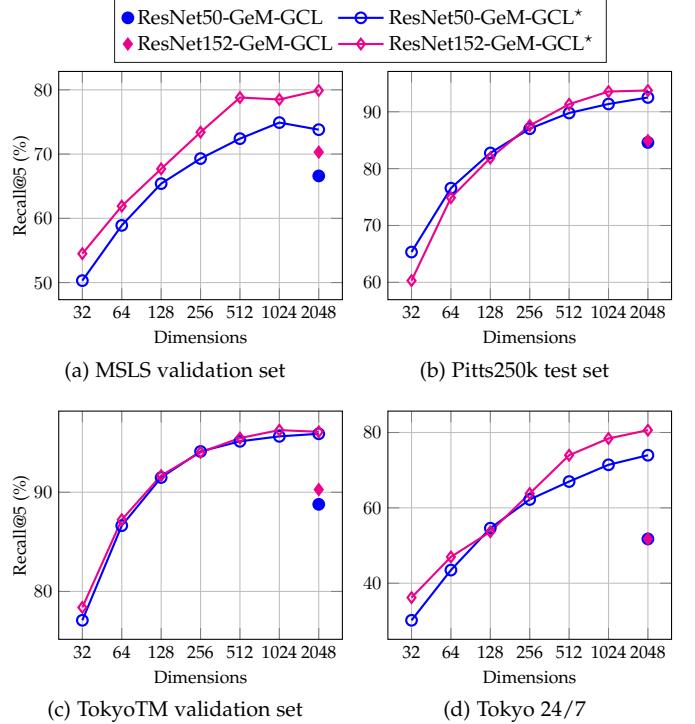


Fig. 12. Results obtained on the MSLS, Pittsburgh, Tokyo 24/7 and TokyoTM datasets by our models with and without PCA whitening. The symbol * indicates the models for which the PCA whitening was applied. For all the datasets, reducing the dimensionality of the latent space vectors and applying the whitening transform contribute to an increase of the retrieval performance.

Loss function on the MSLS, TB-Places and 7Scenes datasets. On the MSLS dataset our ResNet152-GeM-GCL network achieved a top-5 recall of 62.3%, outperforming NetVLAD (top-5 recall of 58%), establishing a new state-of-the-art result. When PCA whitening was applied, the performance was further enhanced, up to a top-5 recall of 70.7%. Furthermore, as opposed to the case of NetVLAD, the training procedure that we deploy does not require a complex pair mining process: we only ensure that each batch contains approximately the same number of positive and negative pairs. Our model also reaches competitive performance on the Pittsburgh, TokyoTM and Tokyo 24/7 datasets.

ACKNOWLEDGEMENTS

This research received partial support from the EU H2020 TrimBot2020 project (grant no. 688007). We would like to thank the Center for Information Technology of the University of Groningen for their support and for providing access to the Peregrine HPC cluster. Finally, we would like to thank Dr. Manuel López Antequera for his assistance with the MSLS dataset.

REFERENCES

- [1] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *CVPR*. IEEE, 2007, pp. 1–8.
- [2] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Lost in quantization: Improving particular object retrieval in large scale image databases," in *CVPR*, 2008, pp. 1–8.
- [3] S. Lowry, N. Sünderhauf, P. Newman, J. J. Leonard, D. Cox, P. Corke, and M. J. Milford, "Visual place recognition: A survey," *IEEE Transactions on Robotics*, vol. 32, no. 1, pp. 1–19, 2016.
- [4] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "Netvlad: Cnn architecture for weakly supervised place recognition," in *CVPR*, 2016, pp. 5297–5307.
- [5] A. Torii, J. Sivic, M. Okutomi, and T. Pajdla, "Visual place recognition with repetitive structures," *TPAMI*, 2015.
- [6] M. J. Milford and G. F. Wyeth, "Seqslam: Visual route-based navigation for sunny summer days and stormy winter nights," in *ICRA*, 2012, pp. 1643–1649.
- [7] J. Shotton, B. Glocker, C. Zach, S. Izadi, A. Criminisi, and A. Fitzgibbon, "Scene coordinate regression forests for camera relocalization in rgb-d images," in *CVPR*, 2013, pp. 2930–2937.
- [8] N. Sünderhauf, P. Neubert, and P. Protzel, "Are we there yet? challenging seqslam on a 3000 km journey across all four seasons," in *ICRA*, 2013, p. 2013.
- [9] F. Warburg, S. Hauberg, M. López-Antequera, P. Gargallo, Y. Kuang, and J. Civera, "Mapillary street-level sequences: A dataset for lifelong place recognition," in *CVPR*, 2020.
- [10] X. Zhang, L. Wang, and Y. Su, "Visual place recognition: A survey from deep learning perspective," *Pattern Recognition*, p. 107760, 2020.
- [11] C. Masone and B. Caputo, "A survey on deep visual place recognition," *IEEE Access*, pp. 1–1, 2021.
- [12] M. Lopez-Antequera, M. Leyva-Vallina, N. Strisciuglio, and N. Petkov, "Place and object recognition by cnn-based cosfire filters," *IEEE Access*, vol. 7, pp. 66157–66166, 2019.
- [13] A. Babenko and V. Lempitsky, "Aggregating local deep features for image retrieval," in *CVPR*, 2015, pp. 1269–1277.
- [14] Z. Chen, O. Lam, A. Jacobson, and M. Milford, "Convolutional neural network-based place recognition," *ACRA*, 2014.
- [15] N. Sünderhauf, S. Shirazi, F. Dayoub, B. Upcroft, and M. Milford, "On the performance of convnet features for place recognition," in *IROS*. IEEE, 2015, pp. 4297–4304.
- [16] M. Lopez-Antequera, R. Gomez-Ojeda, N. Petkov, and J. Gonzalez-Jimenez, "Appearance-invariant place recognition by discriminatively training a convolutional neural network," *Pattern Recognit. Lett.*, vol. 92, pp. 89–95, 2017.
- [17] F. Radenović, G. Tolias, and O. Chum, "Fine-tuning cnn image retrieval with no human annotation," *TPAMI*, vol. 41, no. 7, pp. 1655–1668, 2018.
- [18] M. Leyva-Vallina, N. Strisciuglio, and N. Petkov, "Place recognition in gardens by learning visual representations: data set and benchmark analysis," in *CAIP*. Springer, 2019, pp. 324–335.
- [19] J. Yu, C. Zhu, J. Zhang, Q. Huang, and D. Tao, "Spatial pyramid-enhanced netvlad with weighted triplet loss for place recognition," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 2, pp. 661–674, 2019.
- [20] T. Weyand, A. Araujo, B. Cao, and J. Sim, "Google landmarks dataset v2-a large-scale benchmark for instance-level recognition and retrieval," in *CVPR*, 2020, pp. 2575–2584.
- [21] A. Kendall, Y. Gal, and R. Cipolla, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in *CVPR*, 2018, pp. 7482–7491.
- [22] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell, "Bdd100k: A diverse driving dataset for heterogeneous multitask learning," in *CVPR*, 2020, pp. 2636–2645.
- [23] A. S. Baslamisli, T. T. Groenestege, P. Das, H.-A. Le, S. Karaoglu, and T. Gevers, "Joint learning of intrinsic images and semantic segmentation," in *ECCV*, 2018, pp. 286–302.
- [24] H. Le, A. S. Baslamisli, T. Mensink, and T. Gevers, "Three for one and one for three: Flow, segmentation, and surface normals," in *BMVC*, 2018, p. 145.
- [25] E. Talavera Martinez, M. Leyva-Vallina, M. M. K. Sarker, D. Puig, N. Petkov, and P. Radeva, "Hierarchical approach to classify food scenes in egocentric photo-streams," *IEEE J. Biomed. Health*, vol. 24, no. 3, pp. 866–877, 2019.
- [26] P. Zhang, C. Lan, W. Zeng, J. Xing, J. Xue, and N. Zheng, "Semantics-guided neural networks for efficient skeleton-based human action recognition," in *CVPR*, 2020, pp. 1112–1121.
- [27] Y.-T. Chang, Q. Wang, W.-C. Hung, R. Piramuthu, Y.-H. Tsai, and M.-H. Yang, "Weakly-supervised semantic segmentation via subcategory exploration," in *CVPR*, 2020, pp. 8991–9000.
- [28] J. Wald, H. Dhamo, N. Navab, and F. Tombari, "Learning 3d semantic scene graphs from 3d indoor reconstructions," in *CVPR*, 2020, pp. 3961–3970.
- [29] A. Kendall, M. Grimes, and R. Cipolla, "Posenet: A convolutional network for real-time 6-dof camera relocalization," in *CVPR*, 2015, pp. 2938–2946.
- [30] A. Kendall and R. Cipolla, "Geometric loss functions for camera pose regression with deep learning," in *CVPR*, 2017, pp. 5974–5983.
- [31] S. Naha, Q. Xiao, P. Banik, M. Alimoor Reza, and D. J. Crandall, "Pose-guided knowledge transfer for object part segmentation," in *CVPR Workshops*, 2020, pp. 906–907.
- [32] M. Leyva-Vallina, N. Strisciuglio, M. López-Antequera, R. Tylecek, M. Blaich, and N. Petkov, "Tb-places: A data set for visual place recognition in garden environments," *IEEE Access*, 2019.
- [33] A. Torii, R. Arandjelović, J. Sivic, M. Okutomi, and T. Pajdla, "24/7 place recognition by view synthesis," in *CVPR*, 2013.
- [34] F. Perronnin, Y. Liu, J. Sánchez, and H. Poirier, "Large-scale image retrieval with compressed fisher vectors," in *CVPR*. IEEE, 2010, pp. 3384–3391.
- [35] H. Jegou, M. Douze, C. Schmid, and P. Pérez, "Aggregating local descriptors into a compact image representation," in *CVPR*. IEEE, 2010, pp. 3304–3311.
- [36] D. Galvez-López and J. D. Tardos, "Bags of binary words for fast place recognition in image sequences," *IEEE Transactions on Robotics*, vol. 28, no. 5, pp. 1188–1197, 2012.
- [37] H. Jegou, F. Perronnin, M. Douze, J. Sánchez, P. Perez, and C. Schmid, "Aggregating local image descriptors into compact codes," *TPAMI*, vol. 34, no. 9, pp. 1704–1716, 2011.
- [38] R. Arandjelovic and A. Zisserman, "All about vlad," in *CVPR*, 2013, pp. 1578–1585.
- [39] N. Sünderhauf, S. Shirazi, F. Dayoub, B. Upcroft, and M. Milford, "On the performance of convnet features for place recognition," in *IROS*, 2015, pp. 4297–4304.
- [40] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "Cnn features off-the-shelf: an astounding baseline for recognition," in *CVPR Workshops*, 2014, pp. 806–813.
- [41] J. Thoma, D. P. Paudel, and L. Van Gool, "Soft contrastive learning for visual localization," *NeurIPS*, 2020.
- [42] M. Angelina Uy and G. Hee Lee, "Pointnetvlad: Deep point cloud based retrieval for large-scale place recognition," in *CVPR*, 2018, pp. 4470–4479.
- [43] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *CVPR*, 2017, pp. 652–660.
- [44] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1231–1237, 2013.
- [45] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 year, 1000 km: The oxford robotcar dataset," *Int. J. Robot. Res.*, vol. 36, no. 1, pp. 3–15, 2017.
- [46] T. Sattler, T. Weyand, B. Leibe, and L. Kobbelt, "Image retrieval for image-based localization revisited." in *BMVC*, vol. 1, no. 2, 2012, p. 4.
- [47] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *CVPR*, 2009, pp. 248–255.
- [48] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 million image database for scene recognition," *TPAMI*, vol. 40, no. 6, pp. 1452–1464, 2017.
- [49] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *ECCV*. Springer, 2014, pp. 740–755.
- [50] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge: A retrospective," *Int. J. Comput. Vis.*, vol. 111, no. 1, pp. 98–136, Jan. 2015.
- [51] C. Sun, A. Shrivastava, S. Singh, and A. Gupta, "Revisiting unreasonable effectiveness of data in deep learning era," in *CVPR*, 2017, pp. 843–852.

- [52] J. Zou and L. Schiebinger, "Ai can be sexist and racist — it's time to make it fair," *Nature*, vol. 559, pp. 324–326, 07 2018.
- [53] R. Caruana, "Multitask learning," *Machine learning*, vol. 28, no. 1, pp. 41–75, 1997.
- [54] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *CVPR*, 2017, pp. 4700–4708.
- [55] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.
- [56] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, "Signature verification using a " siamese" time delay neural network," in *NeurIPS*, 1994, pp. 737–744.
- [57] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *CVPR*, vol. 1. IEEE, 2005, pp. 539–546.
- [58] N. Strisciuglio, R. Tylecek, M. Blaich, N. Petkov, P. Biber, J. Hemming, E. van Henten, T. Sattler, M. Pollefeys, T. Gevers *et al.*, "Trimbot2020: an outdoor robot for automatic gardening," in *ISR*. VDE, 2018, pp. 1–6.
- [59] H. Shi, Y. Yang, X. Zhu, S. Liao, Z. Lei, W. Zheng, and S. Z. Li, "Embedding deep metric for person re-identification: A study against large variations," in *ECCV*. Springer, 2016, pp. 732–748.
- [60] H. Taira, M. Okutomi, T. Sattler, M. Cimpoi, M. Pollefeys, J. Sivic, T. Pajdla, and A. Torii, "InLoc: Indoor Visual Localization with Dense Matching and View Synthesis," *TPAMI*, 2019.
- [61] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *J. Mach. Learn. Res.*, vol. 9, no. 11, 2008.
- [62] F. Warburg, M. Jørgensen, J. Civera, and S. Hauberg, "Bayesian triplet loss: Uncertainty quantification in image retrieval," *arXiv preprint arXiv:2011.12663*, 2020.



Nicolai Petkov received the Dr. sc. techn. degree in computer engineering (informationstechnik) from the Dresden University of Technology, Dresden, Germany. Since 1991, he has been a Professor of computer science and the Head of the Intelligent Systems Group, University of Groningen. He has authored two monographs, and has authored or co-authored over 150 scientific papers. He holds four patents. His current research interests include pattern recognition, machine learning, data analytics, and brain-inspired computing, with applications in various areas. He is a member of the editorial boards of several journals.



María Leyva-Vallina received her BSc in Software Engineering from the University of Oviedo in 2015 and her MSc in Artificial Intelligence from the Polytechnic University of Catalonia in 2017. She is currently pursuing her PhD with the Intelligent Systems group in the University of Groningen. Her main research interests are in representation learning for computer vision.



Nicola Strisciuglio received the Ph.D. degree (cum laude) in Computer Science from the University of Groningen (Netherlands) and the Ph.D. degree in Information Engineering from the University of Salerno (Italy). He is currently an Assistant Professor at the Faculty of Electrical Engineering, Mathematics and Computer Science, University of Twente (Netherlands). He has been the General Co-Chair of the 1st, 2nd and 3rd International Conference on Applications of Intelligent Systems (APPIS). His research interests include machine learning, signal processing and computer vision.

APPENDIX A ADDITIONAL RESULTS

In the following, we report extra results and observations on the performance of the proposed methods. In particular, we show how the Generalized Contrastive Loss function contributes to learn representations that better characterize the relevant parts of the input images for a robust computation of their similarity. Furthermore, we presents insights about how the GCL function contributes to a better regularization of the learned latent space.

We also study the effect that different threshold values for the ground truth similarity associated to the images have on the performance of our models. We consider two ground truth thresholds: one based on the GPS distance, in meters, between the locations in which images were taken, and another based on the annotated degree of similarity that we computed, ψ . Furthermore, we trained our models on the MSLS dataset with smaller input images (224 versus the 480 that we used before), and studied the effects that this has on the performance.

A.1 Network activation

In Fig. 13, we show the activation maps of the last convolutional layer of two networks with a ResNet50-GeM backbone, one trained using the Contrastive Loss function (CL) and the other trained using the proposed Generalized Contrastive Loss (GCL) function. We selected two example image pairs from the MSLS test set [9] corresponding to the cities of Kampala and Stockholm, one from the Pitts30k test set [4], and one from the Tokyo 24/7 dataset [33]. For all cases, we observed that the model trained with the GCL function produces higher activation for the common visual features of the images, and lower for the irrelevant parts (i.e. the road or the sky), in contrast to the model trained with the binary CL, which reacts much more weakly in the concerned areas of the pictures. In the example from Stockholm we can observe that our model does not respond to the cars (which vary from picture to picture), while it does respond strongly to the cranes (which are a more permanent feature). The example from Tokyo 24/7 is also particularly interesting: our model trained with the GCL function has high responses on the common parts of the images even under big changes of illumination.

A.2 Visualization of the latent space

In Fig. 14, we show the 2D projection of the vectors representing the difference of the latent space representation of 2000 image pairs (1000 positive and 1000 negative) randomly selected from the Copenhagen set of the MSLS dataset. For each pair, we compute the difference between the map and the query image latent representation. We use this as input to the t-SNE algorithm [61], which projects the representations onto a 2D space. We visualize the vectors produced by two models with a ResNet50-GeM backbone, one trained using the CL function (Fig. 14a) and the other using the GCL (Fig. 14b) function. The effect of the proposed GCL function is evident in the better regularized latent space, where the representation of similar image pairs (blue dots) are more consistently distributed towards the center

of the space. The representations learned using the CL function, instead, form a more scattered and noisy distribution.

A.3 Performance for different ground truths

We performed an evaluation of the performance that the models that we train achieved when varying the ground truth threshold, i.e. when defining different cut values on the criteria that define two images as similar or not. We report results on the MSLS validation dataset.

First, we consider the distance, computed using the GPS coordinates associated to image pairs, between the location from where the images were taken and define a threshold on this value. For instance, we consider as similar all the image pairs that were taken from places with a distance lower than D , and vary D between 5 meters to 50 meters. We report the result in Fig. 15a. An image is considered as correctly identified if at least one of the top-5 retrieved images is at a distance smaller than the considered threshold.

Furthermore, we consider the defined graded similarity ψ and threshold it with values between 0 and 0.9. An image is considered as correctly identified if at least one of the top-5 retrieved images is labeled with a similarity greater than the considered threshold. We report the results in Fig. 15.

In both cases, the models trained using the GCL function are more robust than those trained using the CL function to variations of the criteria and values used to establish the ground truth similarity between image pairs.

A.4 Precision-recall on 7Scenes

We report the precision recall curves that we obtained on the 7Scenes dataset in Fig. 16. The models trained with the GCL function (red lines) consistently outperform their counterpart trained with the CL function (blue lines), when they deploy both the ResNet18 and ResNet34 backbones. The results are consistent for all the sub-sets of the 7Scenes dataset. The case of the *chess* scene is particularly interesting, for which the ResNet34-GCL model achieved near-perfect performance.

A.5 Performance for smaller input images

Complex models are often trained using lower resolution images, in order to better use the hardware resources used for the training process within their physical limitations. We report the results that we achieved on the MSLS test set by training models with the CL and GCL function on images of resolution 224×224 . We achieved lower results than other methods that use the triplet or Bayes triplet loss function. Although the GCL function contributes to a consistent improvement of performance w.r.t. when the CL function is used, we attribute the decrease of performance of our methods to the less and lower resolution information that we can exploit to learn robust representation for image similarity. However, as the memory requirements of our method are much lower than those of other more complex methods, and we do not need to perform a memory-intensive mining process to select training pairs, we can train our methods using bigger images and use large batches of images (up to 512 images on an Nvidia V100 GPU).

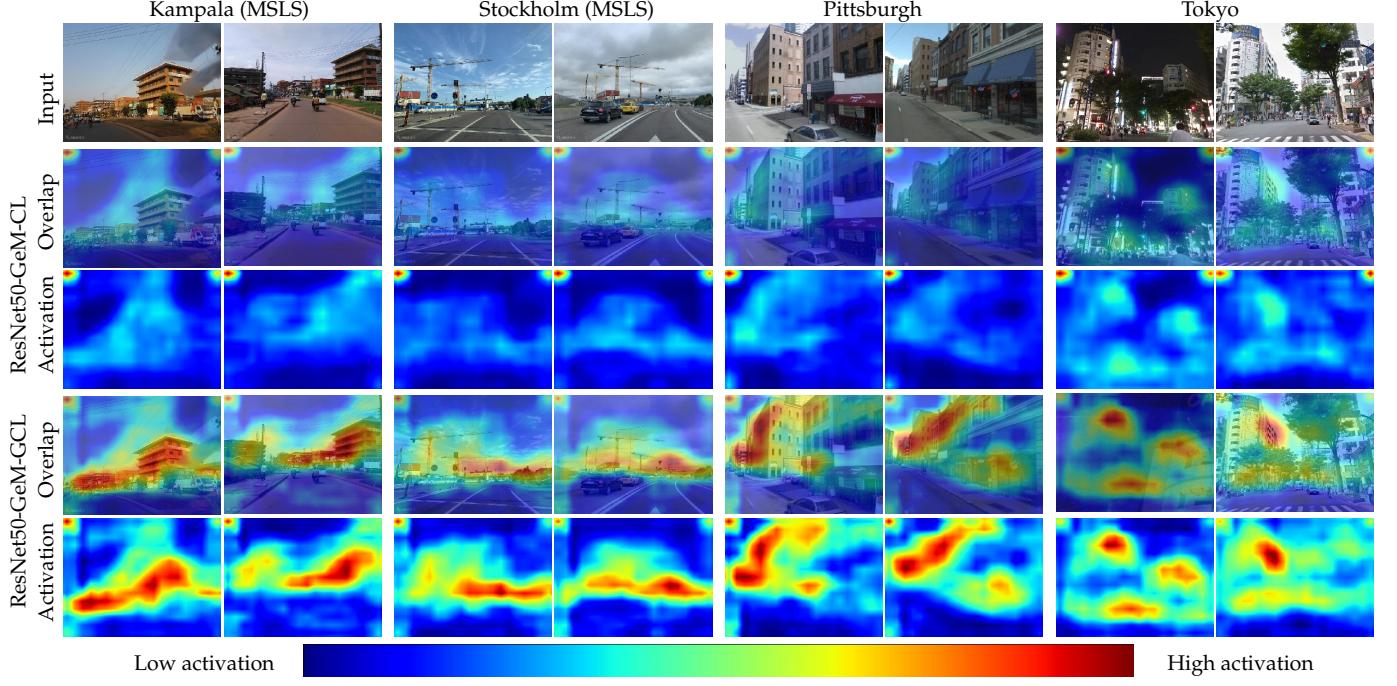


Fig. 13. CNN activations for the ResNet50-GeM-CL and the ResNet50-GeM-GCL model for several input image pairs. The first two pairs, corresponding to the first four columns, are part of the MSLS test set. The third and fourth belong to the Pittsburgh30k and Tokyo 24/7 test set, respectively. We show the activations for the last layer of the ResNet50 backbone, on their own and overlapped with the input images.

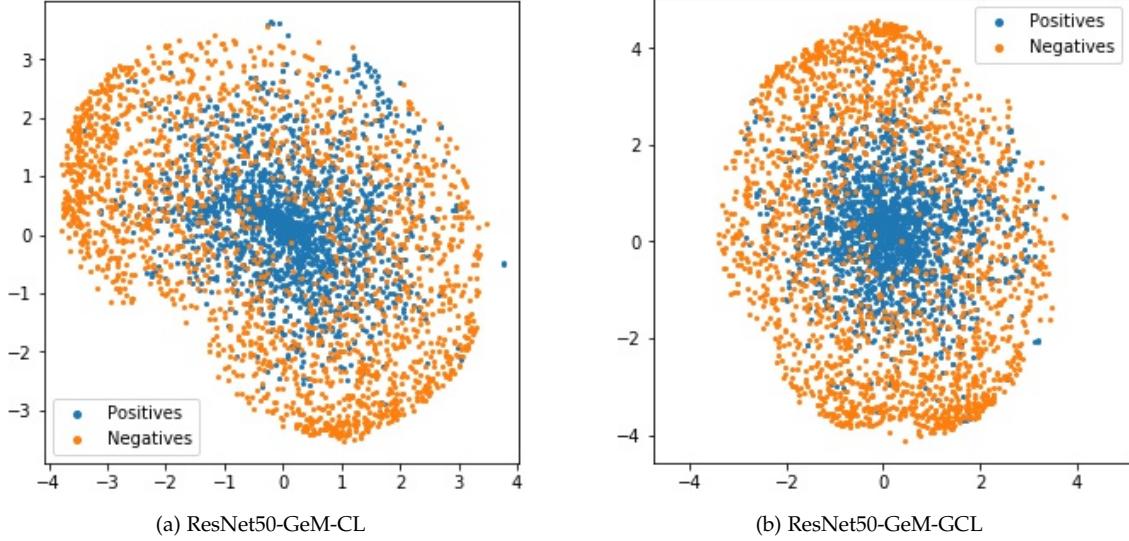


Fig. 14. Visualization of the learned embedded space. We selected 1000 random positive pairs and 1000 random negative pairs from the MSLS Copenhagen set, computed the differences between their representations and projected them into a 2D space using T-SNE.

APPENDIX B GRADIENT COMPUTATIONS

Let us consider two input images x_i and x_j , their latent representations $\hat{f}(x_i)$ and $\hat{f}(x_j)$, and define $d(x_i, x_j)$ the Euclidean distance between the representations, such as:

$$d(x_i, x_j) = \left\| \hat{f}(x_i) - \hat{f}(x_j) \right\|_2$$

For simplicity of notation, hereinafter we refer to $d(x_i, x_j)$ as d .

B.1 Contrastive loss

The Contrastive Loss function is defined as:

$$\mathcal{L}_{CL} = \begin{cases} \frac{1}{2}d^2, & \text{if } y = 1 \\ \frac{1}{2} \max(\tau - d, 0)^2, & \text{if } y = 0 \end{cases}$$

where y corresponds to the binary ground truth label and τ corresponds to the margin.

In order to compute the gradient for this function, we consider three cases, depending on the ground truth label y and the value of the distance d .

Case 1) $y = 1$

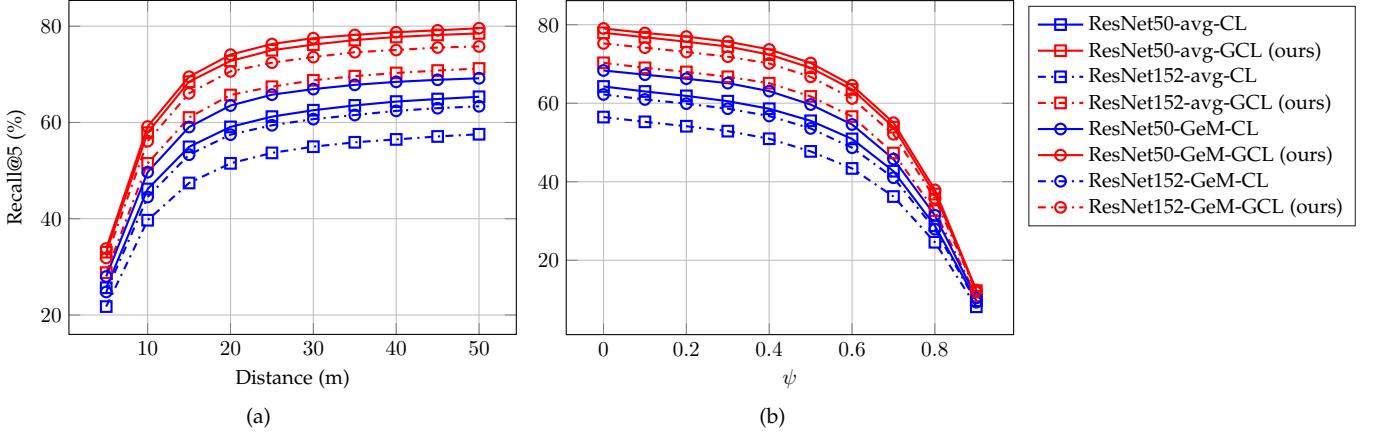


Fig. 15. Comparison of the results achieved by our methods with models trained with the binary contrastive loss for (a) different distance thresholds and (b) different ψ threshold values. Our methods are plotted in red, while the methods trained with the binary contrastive loss function are shown in blue.

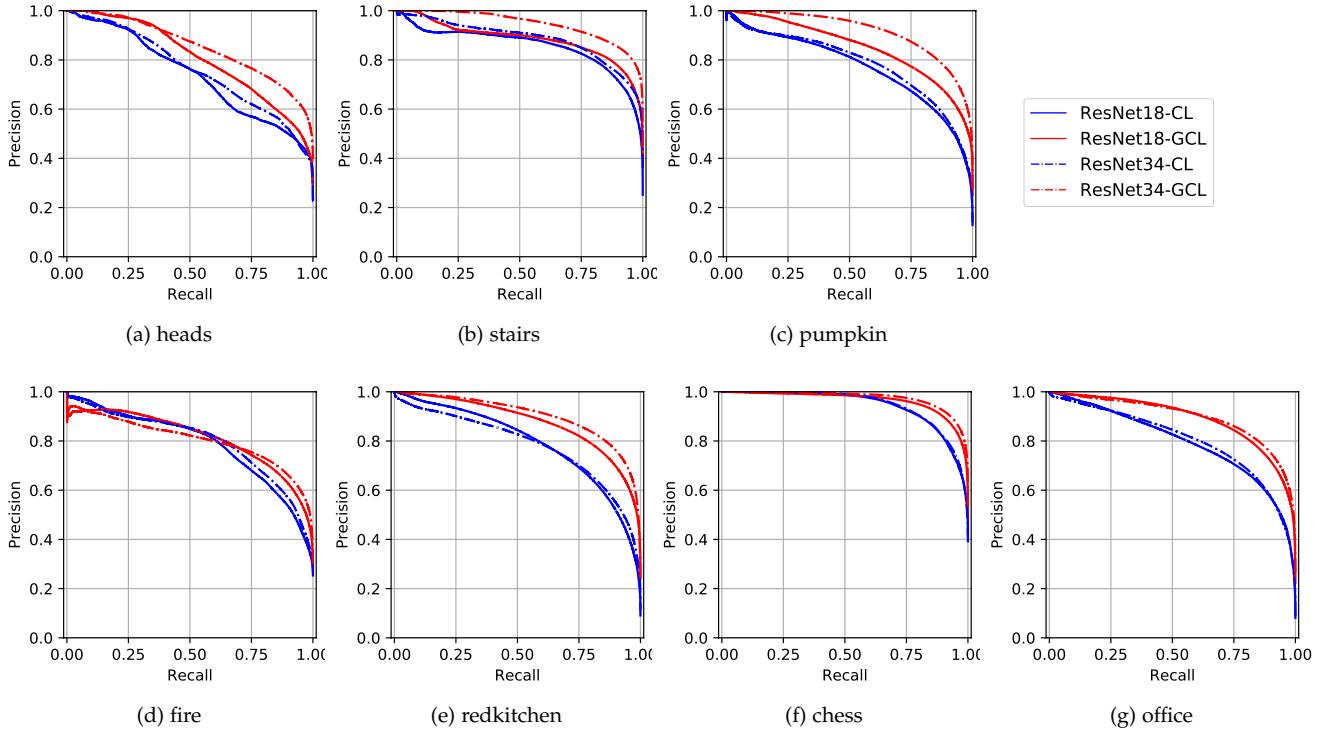


Fig. 16. Precision-recall curves achieved on the 7Scenes dataset by ResNet18 and ResNet34 models trained using the Contrastive Loss function (CL, blue lines) and the proposed Generalized Contrastive loss (GCL, red lines) function.

The loss function becomes:

$$\mathcal{L}_{CL} = \frac{1}{2}d^2$$

and its derivative with respect to d is:

$$\nabla \mathcal{L}_{CL} = \frac{\partial}{\partial d} (\mathcal{L}_{CL}) = \frac{\partial}{\partial d} \left(\frac{1}{2}d^2 \right) = d$$

Case 2) $y = 0, d < \tau$ The loss function becomes:

$$\mathcal{L}_{CL} = \frac{1}{2}(\tau - d)^2$$

and its derivative with respect to d is:

$$\begin{aligned} \nabla \mathcal{L}_{CL} &= \frac{\partial}{\partial d} (\mathcal{L}_{CL}) = \frac{\partial}{\partial d} \left[\frac{1}{2}(\tau - d)^2 \right] = \\ &= (\tau - d)(-1) = d - \tau \end{aligned}$$

Case 3) $y = 0, d \geq \tau$

$$\mathcal{L}_{CL} = 0$$

and the gradient $\mathcal{L}_{CL} = 0$ as well. Thus, case 2 and case 3, for $y = 0$, can be grouped as:

$$\nabla \mathcal{L}_{CL} = \begin{cases} d - \tau, & \text{if } d < \tau \\ 0, & \text{if } d \geq \tau \end{cases}$$

TABLE 6

Additional results on the MSLS test set for images of input size 224x224.

Backbone	Input size	Pooling	Loss	R@1	R@5	R@10
ResNet50	224x224	avg	CL	15.6	27.8	34.9
			GCL	26.1	42	48.5
	480x640	avg	CL	24.9	39.0	44.6
			GCL	35.8	52.0	59.0
	224x224	GeM	TL [62]	37.2	52.2	58.2
			Bayes TL [62]	36.6	51.3	57.4
			CL	17.3	31.8	37.9
			GCL	28.7	44.1	50.3
	480x640	GeM	CL	29.7	44.0	50.7
			GCL	43.3	59.1	65.0
ResNet152	224x224	avg	CL	18.9	32.2	38.2
			GCL	30.8	44.9	52.8
	480x640	avg	CL	29.7	44.2	51.3
			GCL	43.5	59.2	65.2
	224x224	GeM	CL	22.5	37.6	43.8
			GCL	32.1	47.3	54.3
	480x640	GeM	CL	34.1	50.8	56.8
			GCL	45.7	62.3	67.9

and simplified as:

$$\nabla \mathcal{L}_{CL} = \min(d - \tau, 0)$$

Finally, the **gradient of the Contrastive Loss function** is:

$$\nabla \mathcal{L}_{CL} = \begin{cases} d, & \text{if } y = 1 \\ \min(d - \tau, 0), & \text{if } y = 0 \end{cases}$$

B.2 Generalized Contrastive loss

We defined the Generalized Contrastive Loss function as:

$$\mathcal{L}_{GCL} = \psi_{i,j} \cdot \frac{1}{2}d^2 + (1 - \psi_{i,j}) \cdot \frac{1}{2} \max(\tau - d, 0)^2$$

where $\psi_{i,j}$ is the ground truth degree of similarity between the input images x_i and x_j , and its values are in the interval $[0, 1]$. To compute the gradient of the GCL function we consider two cases, namely when 1) the distance d between the representations is lower than the margin τ and 2) the alternative case when d is larger than τ .

Case 1) $d < \tau$ The Generalized Contrastive loss function becomes:

$$\mathcal{L}_{GCL} = \psi_{i,j} \cdot \frac{1}{2}d^2 + (1 - \psi_{i,j}) \cdot \frac{1}{2}(\tau - d)^2$$

and its derivative with respect to d is:

$$\begin{aligned} \nabla \mathcal{L}_{GCL} &= \\ &= \frac{\partial}{\partial d} \mathcal{L}_{GCL} = \\ &= \frac{\partial}{\partial d} \left[\psi_{i,j} \cdot \frac{1}{2}d^2 + (1 - \psi_{i,j}) \cdot \frac{1}{2}(\tau - d)^2 \right] = \\ &= \psi_{i,j} \cdot d + (1 - \psi_{i,j})(\tau - d)(-1) = \\ &= \psi_{i,j} \cdot d + d - \tau - \psi_{i,j} \cdot d + \psi_{i,j} \cdot \tau = \\ &= d + \tau(\psi_{i,j} - 1) \end{aligned}$$

Case 2) $d \geq \tau$

The Generalized Contrastive loss function becomes:

$$\mathcal{L}_{GCL} = \psi_{i,j} \cdot \frac{1}{2}d^2 + (1 - \psi_{i,j}) \cdot \frac{1}{2}(0)^2 = \psi_{i,j} \cdot \frac{1}{2}d^2$$

and its derivative with respect to d is:

$$\begin{aligned} \nabla \mathcal{L}_{GCL} &= \\ &= \frac{\partial}{\partial d} \mathcal{L}_{GCL} = \\ &= \frac{\partial}{\partial d} \left[\psi_{i,j} \cdot \frac{1}{2}d^2 \right] \\ &= d \cdot \psi_{i,j} \end{aligned}$$

Finally, the **gradient of the Generalized Contrastive Loss function** is:

$$\nabla \mathcal{L}_{GCL} = \begin{cases} d + \tau(\psi_{i,j} - 1), & \text{if } d < \tau \\ d \cdot \psi_{i,j}, & \text{if } d \geq \tau \end{cases} \quad (6)$$