# Data-efficient Large Scale Place Recognition with Graded Similarity Supervision

María Leyva-Vallina
University of Groningen
m.leyva.vallina@rug.nl

Nicola Strisciuglio
University of Twente
n.strisciuglio@utwente.nl

Nicolai Petkov
University of Groningen
n.petkov@rug.nl

2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) | 979-8-3503-0129-8/23/$31.00 ©2023 IEEE | DOI: 10.1109/CVPR52729.2023.02249

## Abstract

*Visual place recognition (VPR) is a fundamental task of computer vision for visual localization. Existing methods are trained using image pairs that either depict the same place or not. Such a binary indication does not consider continuous relations of similarity between images of the same place taken from different positions, determined by the continuous nature of camera pose. The binary similarity induces a noisy supervision signal into the training of VPR methods, which stall in local minima and require expensive hard mining algorithms to guarantee convergence. Motivated by the fact that two images of the same place only partially share visual cues due to camera pose differences, we deploy an automatic re-annotation strategy to re-label VPR datasets. We compute graded similarity labels for image pairs based on available localization metadata. Furthermore, we propose a new Generalized Contrastive Loss (GCL) that uses graded similarity labels for training contrastive networks. We demonstrate that the use of the new labels and GCL allow to dispense from hard-pair mining, and to train image descriptors that perform better in VPR by nearest neighbor search, obtaining superior or comparable results than methods that require expensive hard-pair mining and re-ranking techniques.*

## 1. Introduction

Visual place recognition (VPR) is an important task of computer vision, and a fundamental building block of navigation systems for autonomous vehicles [24, 48]. It is approached either with structure-based methods, namely Structure-from-Motion [36] and SLAM [26], or with image retrieval [2, 15, 20, 29, 30, 46]. The former focus on precise relative camera pose estimation [34, 35]. The latter aim at learning image descriptors for effective retrieval of similar images to a given query in a nearest search approach [28]. The goal of descriptor learning is to ensure images of the same place to be projected onto close-by points in a latent space, and images of different places to be projected onto distant points [9, 10, 21]. Contrastive [19, 30] and triplet [2, 22, 23, 27] loss were used for this goal and resulted in state-of-the-art performance on several VPR benchmarks.

(a) reference image (b) GPS distance 6m **positive** (c) GPS distance 25.6m **negative**

Figure 1. (a) A place in the city of Amman. (b) An image taken 6m away is labeled as positive (same place), while (c) an image taken 25.6m away is labeled as negative (not the same place) despite sharing a lot of visual cues.

VPR methods are normally trained using image pairs labelled to indicate they either depict the same place or not, in a binary fashion. In practice, images of a certain place can be taken from different positions, i.e. with a different camera pose, and thus share only a part of their visual cues (or surface in 3D). In existing datasets, two images are usually labeled to be of the same place (positive) if they are taken within a predefined range (usually $25m$) computed using e.g. GPS metadata. This creates ambiguous cases. For instance, Figure 1 shows a reference image (a) of a place and two other pictures taken $6m$ (b) and $25.6m$ (c) away from its position. The images are respectively labeled as positive and negative match, although they share many visual cues (e.g. the building on the right). Binary labels are thus noisy and interfere with the training of VPR networks, that usually stall in local minima. To address this, resource- and time-costly hard pair mining strategies are used to compose the training batches. For example, training NetVLAD [2] on the Mapillary Street Level Sequences (MSLS) dataset [45] can take more than 20 days on an Nvidia v100 gpu due to the complexity of pair mining. We instead build on the observation that two images depict the same place only to a certain degree of shared cues, namely a degree of similarity, and propose to embed this information in new continuous labels for existing datasets that can be used to reduce the effect of noise in the training of effective VPR methods.

In this paper we exploit camera pose metadata or 3D information associated to image pairs as a proxy to estimate an

979-8-3503-0129-8/23/$31.00 ©2023 IEEE
DOI 10.1109/CVPR52729.2023.02249

approximate degree of similarity (hereinafter, graded similarity) between images of the same place, and use it to relabel popular VPR datasets. Graded similarity labels can be used to pick easy- and hard-pairs and compose training batches without complex pair-mining, thus speeding-up the training of VPR networks and enabling an efficient use of data. Furthermore, we embed the graded similarity into a Generalized Contrastive Loss (GCL) function that we use to train a VPR pipeline. The intuition behind this choice is that the update of network weights should not be equal for all training pairs, but rather be influenced by their similarity. The representations of image pairs with larger graded similarity should be pushed together in the latent space more strongly than those of images with a lower graded similarity. The distance in the latent space is thus expected to be a better measure of ranking images according to their similarity, avoiding the use of expensive re-ranking to improve retrieval results. We validate the proposed approaches on several VPR benchmark datasets. To the best of our knowledge, this work is the first to use graded similarity for large-scale place recognition, and paying attention to data-efficient training.

We summarize the contributions of this work as:

- new labels for VPR datasets indicating the graded similarity of image pairs. We computed the labels with automatic methods that use camera pose metadata included with the images or 3D surface information;
- a generalized contrastive loss (GCL) that exploits graded similarity of image pairs to learn effective descriptors for VPR;
- an efficient VPR pipeline trained without hard-pair mining, and that does not require re-ranking. Training our pipeline with a VGG-16 backbone converges $\sim 100x$ faster than NetVLAD with the same backbone, achieving higher VPR results on several benchmarks. The efficiency of our scheme enables training larger backbones in a short time.

## 2. Related works

**Place recognition as image retrieval.** Visual place recognition is widely addressed as a metric learning problem, in which the descriptors of images of a place are learned to be close together in a latent space [10]. Existing methods optimize ranking loss functions, such as contrastive, triplet or average precision [13, 30, 31]. An extensive benchmark of different approaches is in [6]. NetVLAD [2] is a milestone of VPR and builds on a triplet network with an end-to-end trainable VLAD layer. It requires a computationally- and memory-expensive hard-pair mining to compose proper batches and guarantee convergence. SARE [22] uses a NetVLAD backbone trained with a probabilistic attractive and repulsive mechanism, also making use of hard-pair mining. Hard-pair mining addresses issues of the training stalling in local minima due to noisy binary labels, and is used to compose the training batches so that hard pairs are selected for the training [2]. We instead use image metadata (e.g. camera pose as GPS and compass) to a-priori estimate the graded similarity of image pairs, and subsequently use it to balance hard- and easy-pairs in the training batches. This allows to train VPR models using the graded similarity of images and avoiding hard-pair mining.

Training with noisy binary labels produces image descriptors with drawbacks in nearest neighbor search retrieval, and re-ranking algorithms are necessary to post-process the retrieved results and increase VPR performance [7, 32]. Patch-NetVLAD [15] builds on a NetVLAD backbone and performs multi-scale aggregation of NetVLAD descriptors to re-rank retrieval results. A transformer architecture named TransVPR was trained using a triplet loss function and hard-pair mining in [43]. The retrieval step is combined with a costly re-ranking strategy to improve the retrieval results. We instead focus on using more informative and robust image pair labels to avoid noisy training and obtain more effective image descriptors for nearest neighbor search, with no necessity of performing re-ranking.

**Image graded similarity.** Soft assignment to positive and negative classes of image pairs was investigated in [41], where weighting of the assignment was based on the Euclidean distance between the GPS coordinates associated to the images. As the GPS distance induced label noise in the training process, hard-negative pair mining was still necessary to train VPR networks. In [12], image region similarity was coupled with the GPS weak labels in a self-supervised framework to mine hard positive samples. In [5], the authors formulated the VPR metric learning as a classification problem, splitting image training into classes based on similar GPS locations to facilitate large-scale city-wide recognition. Camera pose was used in [4] to estimate the camera frustum overlap and regress descriptors for camera (re-)localization in small-scale (indoor) environments. In [17], a weighting scheme for the contrastive loss function is proposed as a function of the distance in the latent space, which requires an extra step of normalization of the distances to avoid a divergent training. In this work, we relabel VPR datasets using camera pose and field of view overlap, or ratio of shared 3D surface as proxies to estimate the graded similarity of training image pairs. We compute the new labels once, and use them to select the training batches and directly in the optimization of the networks to obtain effective descriptors for VPR in a data-efficient manner.

**Relation and difference with prior works.** We undertake a different direction than previous works, and propose a simplified way to learn image descriptors for retrieval-based VPR. We use contrastive architectures without hard-pair mining and exploit the graded similarity of image pairs to learn

robust descriptors. Instead of developing algorithmic solutions (e.g. hard-pair mining or re-ranking) to achieve better VPR results by increasing the complexity of the methods, we focus on data-efficiency and improve similarity labels to better exploit the training data. This allows to purposely keep the complexity of the architecture simpler (a convolutional backbone and a straightforward pooling strategy) than other methods. We apply prior knowledge and use metadata about the position and orientation of the cameras to estimate a more robust ground truth image similarity that enables to drop expensive hard-mining procedures and train (bigger) networks efficiently. We show that this approach leads to reduced training time and very robust descriptors that perform well in nearest neighbour search with no need of re-ranking.

## 3. Generalized Contrastive Learning

**Preliminaries.** Contrastive approaches for metric learning in visual place recognition consider training a (convolutional) neural network $\hat{f}(x)$ so that the distance of the vector representation of similar (or dissimilar) images in a latent space is minimized (or maximized). In this work, we consider siamese networks optimized using a Contrastive Loss function [14].

Let $x_i$ and $x_j$ be two input images, with $\hat{f}(x_i)$ and $\hat{f}(x_j)$ their descriptors. The distance of the descriptors in the latent space is the $L_2$-distance $d(x_i, x_j) = \left\| \hat{f}(x_i) - \hat{f}(x_j) \right\|_2$. The Contrastive Loss $\mathcal{L}_{CL}$ used to train the networks is defined as:

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

where $\tau$ is the margin, an hyper-parameter that defines a boundary between similar and dissimilar pairs. The ground truth label $y$ is binary: 1 indicates a pair of similar images, and 0 a not-similar pair of images. In practice, however, a binary ground truth for similarity may cause the trained models to provide unreliable predictions.

**Generalized Contrastive Loss.** We reformulate the Contrastive Loss, using a generalized definition of pair similarity as a continuous value $\psi_{i,j} \in [0, 1]$. We define the Generalized Contrastive Loss function $\mathcal{L}_{GCL}$ as:

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

In contrast to Eq. 1, here the similarity $\psi_{i,j}$ is a continuous value ranging from 0 (completely dissimilar) to 1 (identical). By minimising the Generalized Contrastive Loss, the distance of image pairs in the latent space is optimized proportionally to the corresponding degree of similarity.

**Gradient of the GCL.** 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 Constrastive Loss function, the gradient is:

$$\nabla \mathcal{L}_{CL}(x_i, x_j) = \begin{cases} d(x_i, x_j), & \text{if } y = 1 \\ \min(d(x_i, x_j) - \tau, 0), & \text{if } y = 0 \end{cases} \tag{3}$$

The gradient is computed for all positive pairs, and corresponds to a direct minimization of their descriptor distance in the latent space. For negative pairs, the update of the network weights takes place only in the case the distance of the descriptors is within the margin $\tau$. If the latent vectors are already at a distance higher than $\tau$, no update is done.

The Generalized Contrastive Loss, instead, explicitly accounts for graded similarity $\psi_{i,j}$ of input pairs $(x_i, x_j)$ to weight the learning steps, and this reflects into the gradient:

$$\nabla \mathcal{L}_{GCL}(x_i, x_j) = \begin{cases} d(x_i, x_j) + \tau(\psi_{i,j} - 1), & \text{if } d(x_i, x_j) < \tau \\ d(x_i, x_j) \cdot \psi_{i,j}, & \text{if } d(x_i, x_j) \geq \tau \end{cases} \tag{4}$$

The gradient of $\mathcal{L}_{GCL}$ is modulated by the degree of similarity of the input image pairs, $\psi_{i,j}$. This results in an implicit regularization of learned latent space. In the supplementary material, we provide and compare plots of the latent space learned with the $\mathcal{L}_{CL}$ and $\mathcal{L}_{GCL}$ functions. At the extremes of the similarity range, for $\psi_{i,j} = 0$ (completely dissimilar input images) and $\psi_{i,j} = 1$ (same exact input images), the gradient is the same as in Eq. 3.

## 4. Experimental evaluation

### 4.1. Data

**Mapillary Street Level Sequences.** The Mapillary Street Level Sequences (MSLS) dataset is designed for life-long large-scale visual place recognition. It contains about 1.6M images taken in 30 cities across the world [45]. Images are divided into a training (22 cities, 1.4M images), validation (2 cities, 30K images) and test (6 cities, 66k images) set. The dataset presents strong challenges related to images taken at different times of the day, in different seasons and with strong variations of camera viewpoint. The images are provided with GPS data in UTM format and compass angle. According to the original paper [45], two images are considered similar if they are taken by cameras located within $25m$ of distance, and with less than $40^\circ$ of viewpoint variation. We created (and will release) new ground truth labels for the training set of MSLS, with specification of the graded similarity of image pairs (see next Section for details). We use the MSLS dataset to train our large-scale VPR models, which we test on the validation set, and also the private test set using the available evaluation server.

**OOD test data for generalization.** We use out-of-distribution (OOD) test sets, to evaluate the generalization abilities of our models and compare them with existing methods. We thus use the test and validation sets of several other benchmark datasets, namely the Pittsburgh30k [2], Tokyo24/7 [42], RobotCar Seasons v2 [25,33] and Extended CMU Seasons [3,33] datasets. In the supplementary material, we also report results on Pittsburgh250k and TokyoTM [42].
**TB-Places and 7Scenes.** We carried out experiments also using the TB-Places [19] and 7Scenes [38] datasets, which were recorded in small-scale environments. We train models on them and report the results in the supplementary material. TB-Places was recorded in an outdoor garden over two years and contains challenges related to drastic viewpoint variations, as well as illumination changes, and scenes mostly filled with repetitive texture of green color. Each image has 6DOF pose metadata. The 7Scenes dataset is recorded in seven indoor environments. It contains 6DOF pose metadata for each image and a 3D pointcloud of each scene.

The different format and type of metadata, namely 6DOF camera pose and 3D pointclouds of the scenes, are of interest to investigate different ways to estimate the ground truth graded similarity of image pairs. In the following, we present techniques to automatically re-label VPR datasets, when 6DOF pose or 3D pointclouds metadata are available.

### 4.2. Graded similarity labels

Images of the same place can be taken from different positions, i.e. with different camera pose, and share only part of the visual cues. On the basis of the amount of shared characteristics among images, we indeed tend to perceive images more or less similar [11]. Actual labeling of VPR datasets do not consider this and instead mark two images either similar (depicting the same place) or not (depicting different places). This does not take into account continuous relations between images, which are induced by the continuous nature of camera pose.

We combine the concept of perceived visual similarity with the continuous nature of camera pose, and design a method to automatically relabel VPR dataset by annotating the graded similarity of image pairs[1]. We approximate the similarity between two images via a proxy, namely measuring the overlap of the field of view of the cameras or 3D information associated to the images.

**Graded similarity for MSLS and TB-Places: field of view overlap.** For MSLS and TB-Places dataset, images are provided with camera pose metadata in the form of a vector $\mathbf{t}$ and orientation $\alpha$. The MSLS has UTM data and compass angle information associated to the images, while in TB-Places the images are provided with precise camera pose recorded with a laser tracker and IMU.

[1]We release the labels at `https://github.com/marialeyvallina/generalized_contrastive_loss`.

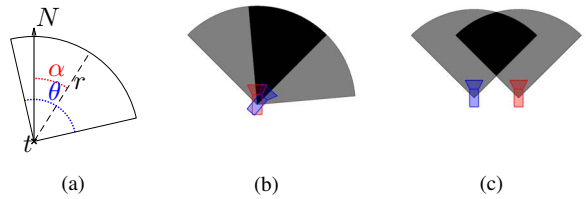



Figure 2. (a) FoV with angle $\theta$ and radius $r$. The point $\mathbf{t}$ is the camera location in the environment, and $\alpha$ is the camera orientation in the form of a compass angle with respect to the north $N$. (b) An example of FoV overal for two cameras in the same position and with orientations $40^\circ$ apart. (c) An example of FoV overlap for two cameras located 25m apart but with the same orientation.

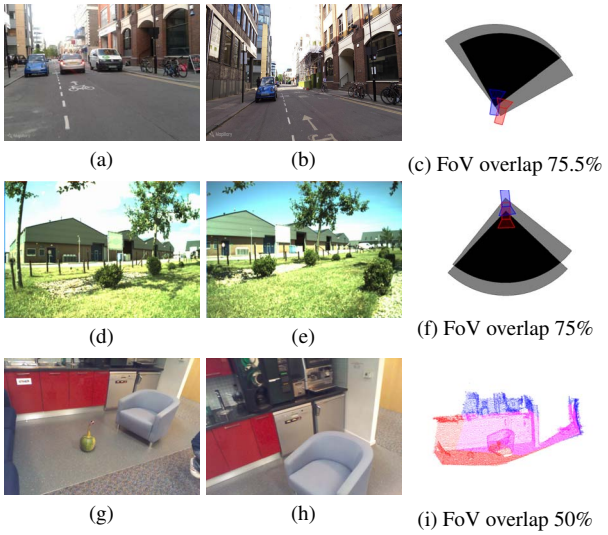



Figure 3. Examples of graded similarity estimated for MSLS (first row) and TB-Places (second row) with the FoV overlap, and 7Scenes (third row) with 3D overlap (magenta color points).

For an image, we build the field of view (FoV) of the camera as the sector of the circle centered at $\mathbf{t}$ with radius $r$, delimited by the angle range $[\alpha - \frac{\theta}{2}, \alpha + \frac{\theta}{2}]$ (see Fig. 2a), where $\theta$ is the nominal size of the FoV of the camera concerned. The FoV overlap is the intersection-over-union (IoU) of the FoV of the cameras. In the first and second row of Figure 3, we show examples of the graded similarity estimated for pairs in the MSLS and TB-Places datasets. This approach differs from the camera frusta overlap [4] that needs 3D overlap measures for precise camera pose estimation.

**Graded similarity for 7Scenes: 3D overlap.** The 7Scenes dataset has a 3D pointcloud for each scene, and 6DOF camera pose associated to the images. In this case, we can estimate the similarity overlap differently from the cases above. We project a pair of images onto the 3D pointcloud, so that we select the points associated to the two images and measure their intersection-over-union (IoU) as a measure of the image pair similarity. A similar strategy based on

maximum inliers was used for hard-pair mining in [30]. In the third row of Figure 3, we show an example of graded similarity estimation for two images in the 7Scenes dataset.

### 4.3. Place recognition pipeline

**Embeddings.** We use a fully convolutional backbone (ResNet [16], VGG16 [39] and ResNeXt [47]) with a GeM pooling layer [30], which receives as input an image $x \in R^{w_n \times h_n \times 3}$, and outputs a representation $\hat{f}(x) \in R^{d_m}$, where $d_m$ is the number of kernels of the last convolutional layer. We train $\hat{f}(x)$ using a contrastive learning framework.

**Training batch composition.** Batch composition is an important part of model training. For contrastive architectures, the selection of meaningful image tuples is crucial for the correct optimization of the model. If the selected tuples are too challenging, the training might become unstable [37]. If they are too easy, the learning might stall. This, coupled with binary pairwise labels, makes necessary to use complex descriptor-based mining strategies to ensure model convergence. The *hard-negative mining* strategy needed to train contrastive networks [2, 15, 22] periodically computes the descriptor of all training images and their pairwise distance to select certain pairs (tuples) of images to be used for the subsequent training steps. This is a memory- and computation-expensive procedure.

We do not perform hard-pair mining. We instead compose the training batches taking into account the graded similarity labels that we computed. We balance the pairs in the training batches on the basis of their annotated degree of similarity. For each batch, we make sure to select $50\%$ of positive pairs (similarity higher than $50\%$), $25\%$ of soft negative samples (similarity higher than $0\%$ and lower than $50\%$) and $25\%$ of hard negatives ($0\%$ similarity) – see Section 5 for results.

**Image retrieval.** Let us consider a set $X$ of reference images with a known camera location, and a set $Y$ of query images taken from unknown positions. In order to localize the camera that took the query images, similar images to the query are to be retrieved from the reference set. We compute the descriptors of the reference images $\hat{f}(x)\, \forall x \in X$, and of the query images $\hat{f}(y)\, \forall y \in Y$. For a given query descriptor $\hat{f}(y)$, image retrieval is performed by nearest neighbor search within the reference descriptors $\hat{f}(x)\, \forall x \in X$, retrieving $k$ images ranked by the closest descriptor distance.

### 4.4. Performance measures

We apply widely used place recognition evaluation protocols and consider a query as correctly identified if any of the top-k retrieved images are annotated as a positive match [2, 34, 45]. We computed the following metrics. For the MSLS, Pittsburgh30k, Tokyo24/7 (Pittsburgh250k, TokyoTM, TrimBot2020 and 7Scences in the supplementary material) we compute the **Top-k recall (R@k)**. It measures the percentage of queries for which at least a correct map image is present among their $k$ nearest neighbors retrieved. For the RobotCar Seasons v2 and the Extended CMU datasets, we compute the **percentage of correctly localized queries**. It measures the amount of images that are correctly retrieved for a given translation and rotation threshold.

| | | | MSLS-Val | | | MSLS-Test | | |
|---|---|---|---|---|---|---|---|---|
| **Method** | **Loss** | **Batch** | **R@1** | **R@5** | **R@10** | **R@1** | **R@5** | **R@10** |
| VGG-GeM | CL | binary | 47.0 | 60.3 | 65.5 | 27.9 | 40.5 | 46.5 |
| VGG-GeM | GCL | binary | 57.4 | 73.4 | 76.9 | 35.9 | 49.3 | 57.8 |
| VGG-GeM | CL | graded | 45.8 | 60.1 | 65.1 | 28.0 | 40.8 | 47.0 |
| VGG-GeM | GCL | graded | 65.9 | 77.8 | 81.4 | 41.7 | 55.7 | 60.6 |

Table 1. Effect of graded similarity labels on batch composition and model training.

## 5. Results and discussion

**Graded similarity for batch composition and model training.** We carry out a baseline experiment to evaluate the impact of the new graded similarity labels on the effectiveness of the learned descriptors. We analyze their contribution to the composition of the batches, and directly to the training of the network by using them in combination with the proposed GCL. We first consider the traditional binary labels only, and compose batches by balancing positive and negative pairs. Subsequently, we compose the batches by considering the new graded similarity labels, and select $50\%$ of positive pairs (similarity higher than $50\%$), $25\%$ of soft negative samples (similarity higher than $0\%$ and lower than $50\%$) and $25\%$ of hard negatives ($0\%$ similarity).

In Table 1, we report the results using a VGG16 backbone on the MSLS dataset. These results demonstrate that the proposed graded similarity labels are especially useful for training descriptors that perform better in nearest neighbor search retrieval, and also contribute to form better balanced batches to exploit the data in a more efficient way. In the following, all experiments use the batch composition based on graded similarity.

**Comparison with existing works.** We compared our results with several place recognition works. We considered methods that use global descriptors like NetVLAD [2] (with 16 and 64 clusters in the VLAD layer) and methods based on two-stages retrieval and re-ranking pipelines, such as Patch-NetVLAD [15], DELG [7] and SuperGlue [32]. We compared also against TransVPR [43], a transformer with and without a re-ranking stage. Table 2 reports the results of our method in comparison to others. All the methods included in the table are based on backbones trained on the MSLS datasets. The results of Patch-NetVLAD and TransVPR are taken from the respective papers, which also contain those of DELG and SuperGlue. When trained with VGG16 as backbone, our model (VGG16-GeM-GCL) obtains an absolute improvement of R@5 equal to 11.7% compared to

| Method | PCA$_w$ | Dim | MSLS-Val R@1 | MSLS-Val R@5 | MSLS-Val R@10 | MSLS-Test R@1 | MSLS-Test R@5 | MSLS-Test R@10 | Pitts30k R@1 | Pitts30k R@5 | Pitts30k R@10 | Tokyo24/7 R@1 | Tokyo24/7 R@5 | Tokyo24/7 R@10 | RobotCar Seasons v2 .25m/2° | RobotCar Seasons v2 .5m/5° | RobotCar Seasons v2 5m/10° | Extended CMU Seasons .25m/2° | Extended CMU Seasons .5m/5° | Extended CMU Seasons 5m/10° |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| NetVLAD 64 | N | 32768 | 44.6 | 61.1 | 66.4 | 28.8 | 44.0 | 50.7 | 40.4 | 64.5 | 74.2 | 11.4 | 24.1 | 31.4 | 2.0 | 9.2 | 45.5 | 1.3 | 4.5 | 31.9 |
| NetVLAD 64 | Y | 4096 | 70.1 | 80.8 | 84.9 | 45.1 | 58.8 | 63.7 | 68.6 | 84.7 | 88.9 | 34.0 | 47.6 | 57.1 | 4.2 | 18.0 | 68.1 | 3.9 | 12.1 | 58.4 |
| NetVLAD 16 | N | 8192 | 49.5 | 65.0 | 71.8 | 29.3 | 43.5 | 50.4 | 48.7 | 70.6 | 78.9 | 13.0 | 33.0 | 43.8 | 1.8 | 9.2 | 48.4 | 1.7 | 5.5 | 39.1 |
| NetVLAD 16 | Y | 4096 | 70.5 | 81.1 | 84.3 | 39.4 | 53.0 | 57.5 | 70.3 | 84.1 | 89.1 | 37.8 | 53.3 | 61.0 | 4.8 | 17.9 | 65.3 | 4.4 | 13.7 | 61.4 |
| TransVPR [43] | - | - | 70.8 | 85.1 | 86.9 | 48 | 67.1 | 73.6 | 73.8 | 88.1 | 91.9 | - | - | - | 2.9 | 11.4 | 58.6 | - | - | - |
| SP-SuperGlue$^\star$ [32] | - | - | 78.1 | 81.9 | 84.3 | 50.6 | 56.9 | 58.3 | 87.2 | 94.8 | 96.4 | 88.2 | 90.2 | 90.2 | 9.5 | **35.4** | 85.4 | 9.5 | 30.7 | **96.7** |
| DELG$^\star$ [7] | - | - | 83.2 | 90.0 | 91.1 | 52.2 | 61.9 | 65.4 | **89.9** | **95.4** | **96.7** | **95.9** | **96.8** | **97.1** | 2.2 | 8.4 | 76.8 | 5.7 | 21.1 | 93.6 |
| Patch NetVLAD$^\star$ [15] | Y | 4096 | 79.5 | 86.2 | 87.7 | 48.1 | 57.6 | 60.5 | 88.7 | 94.5 | 95.9 | 86.0 | 88.6 | 90.5 | 9.6 | 35.3 | **90.9** | **11.8** | **36.2** | 96.2 |
| TransVPR$^\star$ [43] | - | - | **86.8** | **91.2** | 92.4 | **63.9** | 74 | 77.5 | 89 | 94.9 | 96.2 | - | - | - | **9.8** | 34.7 | 80 | - | - | - |
| NetVLAD-GCL | N | 32768 | 62.7 | 75.0 | 79.1 | 41.0 | 55.3 | 61.7 | 52.5 | 74.1 | 81.7 | 20.3 | 45.4 | 49.5 | 3.3 | 14.1 | 58.2 | 3.0 | 9.7 | 52.3 |
| NetVLAD-GCL | Y | 4096 | 63.2 | 74.9 | 78.1 | 41.5 | 56.2 | 61.3 | 53.5 | 75.2 | 82.9 | 28.3 | 41.9 | 54.9 | 3.4 | 14.2 | 58.8 | 3.1 | 9.7 | 52.4 |
| VGG-GeM-GCL | N | 512 | 65.9 | 77.8 | 81.4 | 41.7 | 55.7 | 60.6 | 61.6 | 80.0 | 86.0 | 34.0 | 51.1 | 61.3 | 3.7 | 15.8 | 59.7 | 3.6 | 11.2 | 55.8 |
| VGG-GeM-GCL | Y | 512 | 72.0 | 83.1 | 85.8 | 47.0 | 60.8 | 65.5 | 73.3 | 85.9 | 89.9 | 47.6 | 61.0 | 69.2 | 5.4 | 21.9 | 69.2 | 5.7 | 17.1 | 66.3 |
| ResNet50-GeM-GCL | N | 2048 | 66.2 | 78.9 | 81.9 | 43.3 | 59.1 | 65.0 | 72.3 | 87.2 | 91.3 | 44.1 | 61.0 | 66.7 | 2.9 | 14.0 | 58.8 | 3.8 | 11.8 | 61.6 |
| ResNet50-GeM-GCL | Y | 1024 | 74.6 | 84.7 | 88.1 | 52.9 | 65.7 | 71.9 | 79.9 | 90.0 | 92.8 | 58.7 | 71.1 | 76.8 | 4.7 | 20.2 | 70.0 | 5.4 | 16.5 | 69.9 |
| ResNet152-GeM-GCL | N | 2048 | 70.3 | 82.0 | 84.9 | 45.7 | 62.3 | 67.9 | 72.6 | 87.9 | 91.6 | 34.0 | 51.8 | 60.6 | 2.9 | 13.1 | 63.5 | 3.6 | 11.3 | 63.1 |
| ResNet152-GeM-GCL | Y | 2048 | 79.5 | 88.1 | 90.1 | 57.9 | 70.7 | 75.7 | 80.7 | 91.5 | 93.9 | 69.5 | 81.0 | 85.1 | 6.0 | 21.6 | 72.5 | 5.3 | 16.1 | 66.4 |
| ResNeXt-GeM-GCL | N | 2048 | 75.5 | 86.1 | 88.5 | 56.0 | 70.8 | 75.1 | 64.0 | 81.2 | 86.6 | 37.8 | 53.6 | 62.9 | 2.7 | 13.4 | 65.2 | 3.5 | 10.5 | 58.8 |
| ResNeXt-GeM-GCL | Y | 1024 | 80.9 | 90.7 | **92.6** | 62.3 | **76.2** | **81.1** | 79.2 | 90.4 | 93.2 | 58.1 | 74.3 | 78.1 | 4.7 | 21.0 | 74.7 | 6.1 | 18.2 | 74.9 |

Table 2. Comparison to state-of-the-art methods on benchmark datasets. All methods are trained on the MSLS training set. Our top results are underlined, while overall best results are in bold. Methods using re-ranking are in the middle part of the table and marked with $^\star$.

NetVLAD-64. This shows that the proposed graded similarity labels and the GCL function contribute to learn more powerful descriptors for place recognition, while keeping the complexity of the training process lower as hard-pair mining is not used. The result improvement holds also when the descriptors are post-processed with PCA whitening.

The data- and memory-efficiency of our pipeline allows us to easily train more powerful backbones, such as ResNeXt, that is instead tricky to do for other methods due to memory and compute requirements. Our ResNeXt+GCL outperforms the best method on the MSLS test set, namely TransVPR without re-ranking by 8.9% and with re-ranking by 2.6% (absolute improvement of R@5). It compares favorably with re-ranking based methods such as Patch-NetVLAD, DELG and SuperGLUE improving the R@5 by 18.6%, 14.3% and 18.3%, respectively. We point out that we do not re-rank the retrieved images, and purposely keep the complexity of the steps at the strict necessary to perfom the VPR retrieval task. We attribute our high results mainly to the effectiveness of the descriptors learned with the GCL function using the new graded similarity labels.

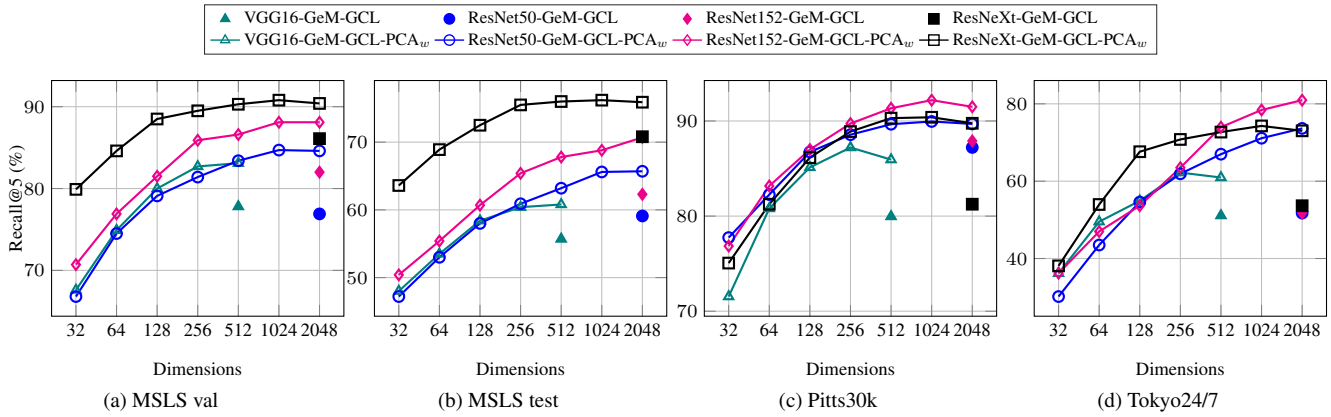
**Generalization to other datasets.** In Table 2 and Table 3, we also report the results of generalization to Pittsburgh30k, Tokyo 24/7, RobotCar Seasons v2 and Extended CMU Seasons (plus Pittsburgh250k and TokyoTM in the supplementary materials). The models trained with the GCL function generalize well to unseen datasets, in many cases better than existing methods that retrieve the $k$-nearest neighbours based on descriptor distance only. Our models also generalize well to urban localization datasets like RobotCar Seasons V2 and Extended CMU Seasons, achieving up to 21.9% and 19% of correctly localized queries within 0.5m and 5°, respectively. The results of GCL-based networks are higher than those obtained by NetVLAD, and especially higher than those of TransVPR (with no re-ranking) that uses a transformer as backbone. Note that we do not perform 6DOF pose estimation, but estimate the pose of a query image by inheriting that of the best retrieved match, and thus not compare with methods that perform refined pose estimation. This is inline with the experiments in [15].

Our models are outperformed only by methods that include a re-ranking strategies to refine the list of retrieved images, on the Pittsburgh30k, RobotCar Seasons v2 and Extended CMU Seasons datasets. However, these methods perform extra heavy computations (e.g. up to $6s$ per query in PatchNetVLAD [15]) to re-rank the list of retrieved images, not focusing on the representation capabilities of the learned descriptors themselves. Thus, we find a direct comparison with these methods not fair. On the contrary, these results demonstrate the fact the VPR descriptors learned used the proposed labels and GCL have better representation capabilities than those produced by other methods, achieving higher results in out-of-distribution experiments as well.

**Ablation study: backbone and contrastive loss.** We carried out ablation experiments using four backbones, namely VGG16 [39], ResNet50, ResNet152 [16], and ResNeXt101-32x8d (hereinafter ResNeXt) [47], and the GeM [30] global pooling layer, and an additional NetVLAD-GCL model. Extra ablation experiments with an average global pooling layer are included in the supplementary material. For each backbone, we train with the binary Contrastive Loss (CL) and our Generalized Contrastive Loss (GCL). We report the results in Table 3. The models trained with the GCL consistently outperform their counterpart trained with the CL, showing better generalization to other datasets. Moreover, we demonstrate that a VGG16-GeM architecture outperforms a more

| Method | Loss | PCA$_w$ | Dim | MSLS-Val R@1 | MSLS-Val R@5 | MSLS-Val R@10 | MSLS-Test R@1 | MSLS-Test R@5 | MSLS-Test R@10 | Pitts30k R@1 | Pitts30k R@5 | Pitts30k R@10 | Tokyo24/7 R@1 | Tokyo24/7 R@5 | Tokyo24/7 R@10 | RobotCar Seasons v2 0.25m/2° | RobotCar Seasons v2 0.5m/5° | RobotCar Seasons v2 5.0m/10° | Extended CMU Seasons 0.25m/2° | Extended CMU Seasons 0.5m/5° | Extended CMU Seasons 5.0m/10° |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| NetVLAD | CL | N | 32768 | 38.5 | 56.6 | 64.1 | 24.8 | 39.1 | 46.1 | 24.7 | 48.3 | 61.1 | 7.0 | 18.1 | 24.8 | 0.9 | 5.2 | 31.5 | 1.0 | 1.4 | 11.1 |
| | GCL | N | 32768 | 62.7 | 75.0 | 79.1 | 41.0 | 55.3 | 61.7 | 52.5 | 74.1 | 81.7 | 20.3 | 45.4 | 49.5 | 3.3 | 14.1 | 58.2 | 3.0 | 9.7 | 52.3 |
| | CL | Y | 4096 | 39.6 | 60.3 | 65.3 | 26.4 | 40.5 | 48.2 | 27.5 | 51.6 | 64.1 | 6.7 | 16.2 | 25.7 | 0.0 | 0.0 | 0.0 | 1.0 | 3.3 | 25.4 |
| | GCL | Y | 4096 | 63.2 | 74.9 | 78.1 | 41.5 | 56.2 | 61.3 | 53.5 | 75.2 | 82.9 | 28.3 | 41.9 | 54.9 | 3.4 | 14.2 | 58.8 | 3.1 | 9.7 | 52.4 |
| VGG-GeM | CL | N | 512 | 47.0 | 60.3 | 65.5 | 27.9 | 40.5 | 46.5 | 51.2 | 71.9 | 79.7 | 24.1 | 39.4 | 47.0 | 3.1 | 13.2 | 55.0 | 2.8 | 8.6 | 44.5 |
| | GCL | N | 512 | 65.9 | 77.8 | 81.4 | 41.7 | 55.7 | 60.6 | 61.6 | 80.0 | 86.0 | 34.0 | 51.1 | 61.3 | 3.7 | 15.8 | 59.7 | 3.6 | 11.2 | 55.8 |
| | CL | Y | 512 | 61.4 | 75.1 | 78.5 | 36.3 | 49.0 | 54.1 | 64.7 | 81.5 | 86.8 | 36.2 | 54.0 | 57.8 | 4.2 | 18.7 | 62.5 | 4.4 | 13.4 | 56.5 |
| | GCL | Y | 512 | 72.0 | 83.1 | 85.8 | 47.0 | 60.8 | 65.5 | 73.3 | 85.9 | 89.9 | 47.6 | 61.0 | 69.2 | 5.4 | 21.9 | 69.2 | 5.7 | 17.1 | 66.3 |
| ResNet50-GeM | CL | N | 2048 | 51.4 | 66.5 | 70.8 | 29.7 | 44.0 | 50.7 | 61.5 | 80.0 | 86.9 | 30.8 | 46.0 | 56.5 | 3.2 | 15.4 | 61.5 | 3.2 | 9.6 | 49.5 |
| | GCL | N | 2048 | 66.2 | 78.9 | 81.9 | 43.3 | 59.1 | 65.0 | 72.3 | 87.2 | 91.3 | 44.1 | 61.0 | 66.7 | 2.9 | 14.0 | 58.8 | 3.8 | 11.8 | 61.6 |
| | CL | Y | 1024 | 63.2 | 76.6 | 80.7 | 37.9 | 53.0 | 58.5 | 66.2 | 82.2 | 87.3 | 36.2 | 51.8 | 61.0 | 5.0 | 21.1 | 66.5 | 4.7 | 13.4 | 51.6 |
| | GCL | Y | 1024 | 74.6 | 84.7 | 88.1 | 52.9 | 65.7 | 71.9 | 79.9 | 90.0 | 92.8 | 58.7 | 71.1 | 76.8 | 4.7 | 20.2 | 70.0 | 5.4 | 16.5 | 69.9 |
| ResNet152-GeM | CL | N | 2048 | 58.0 | 72.7 | 76.1 | 34.1 | 50.8 | 56.8 | 66.5 | 83.8 | 89.5 | 34.6 | 57.1 | 63.5 | 3.3 | 15.2 | 64.0 | 3.2 | 9.7 | 52.2 |
| | GCL | N | 2048 | 70.3 | 82.0 | 84.9 | 45.7 | 62.3 | 67.9 | 72.6 | 87.9 | 91.6 | 34.0 | 51.8 | 60.6 | 2.9 | 13.1 | 63.5 | 3.6 | 11.3 | 63.1 |
| | CL | Y | 2048 | 66.9 | 80.9 | 83.8 | 44.8 | 59.2 | 64.8 | 71.2 | 85.8 | 89.8 | 54.3 | 68.9 | 75.6 | **6.1** | **23.5** | 68.9 | 4.8 | 14.2 | 55.0 |
| | GCL | Y | 2048 | 79.5 | 88.1 | 90.1 | 57.9 | 70.7 | 75.7 | **80.7** | **91.5** | **93.9** | **69.5** | **81.0** | **85.1** | 6.0 | 21.6 | 72.5 | 5.3 | 16.1 | 66.4 |
| ResNeXt-GeM | CL | N | 2048 | 62.6 | 76.4 | 79.9 | 40.8 | 56.5 | 62.1 | 56.0 | 77.5 | 85.0 | 37.8 | 54.9 | 62.5 | 1.9 | 10.4 | 54.8 | 2.9 | 9.0 | 52.6 |
| | GCL | N | 2048 | 75.5 | 86.1 | 88.5 | 56.0 | 70.8 | 75.1 | 64.0 | 81.2 | 86.6 | 37.8 | 53.6 | 62.9 | 2.7 | 13.4 | 65.2 | 3.5 | 10.5 | 58.8 |
| | CL | Y | 1024 | 74.3 | 87.0 | 89.6 | 49.9 | 63.8 | 69.4 | 70.9 | 85.7 | 90.2 | 50.8 | 67.6 | 74.3 | 3.8 | 17.2 | 68.2 | 4.9 | 14.4 | 61.7 |
| | GCL | Y | 1024 | **80.9** | **90.7** | **92.6** | 62.3 | **76.2** | **81.1** | 79.2 | 90.4 | 93.2 | 58.1 | 74.3 | 78.1 | 4.7 | 21.0 | **74.7** | **6.1** | **18.2** | **74.9** |

Table 3. Ablation study on backbone, Contrastive (CL) vs Generalized Contrastive (GCL) loss, and PCA. All models are trained on MSLS.




(a) MSLS val (b) MSLS test (c) Pitts30k (d) Tokyo24/7

Figure 4. Ablation results on the MSLS validation, MSLS test, Pittsburgh30k and Tokyo 24/7 datasets, with different PCA dimensions.

complex NetVLAD when trained with our GCL function. We also perform whitening and PCA on the descriptors, which further boost the performance.

**Ablation study: PCA and whitening.** We also study the effect of whitening and PCA dimensionality reduction from 32 to 2048 dimensions. Figure 4 shows the results on the MSLS, Pittsburgh30k and Tokyo 24/7 datasets. In general, the larger the size of the descriptors, the better the results. However, our models maintain comparably high results when the descriptors are whitened and reduced to 256 dimensions, still outperforming the full-size descriptors without whitening. We observed up to a 28.2% improvement in the case of Tokyo 24/7 and 12.8% on the MSLS validation set. When comparing our VGG16-GeM-GCL model reduced to 256 dimensions with NetVLAD (16× less descriptor size), we still achieve higher results (R@5 of 82.7% vs 80.8% on MSLS validation, 60.4% vs 58.9% on MSLS test, 87.2% vs 84.7% on Pittsburg30k and 62.2% vs 47.6% on Tokyo 24/7). It is to highlight that the contributions of the PCA/whitening and the GCL are complementary, meaning that they can be used together to optimize retrieval performance.

**Comparison with other loss functions.** We compared with other loss functions used for VPR, by following up on the experiments in [41]. The loss functions included in the comparison in [41] are the triplet loss [2] (also with Huber distance [40]), quadruplet loss [8], lazy triplet and lazy quadruplet loss [1], plus functions that embed mechanisms to circumvent the use of binary labels, namely a multi-similarity loss [44], log-ratio [18] and soft contrastive loss [41]. The results reported in Table 4 show that the GCL achieves higher localization accuracy especially when stricter thresholds for distance and angle are set. These results indicate that the GCL descriptors are better effective in neighbor search and retrieval, and their ranking based on distance from the query descriptor is a more reliable measure of visual place similarity. All methods in the upper part of the

| Loss function | All | | | Urban | | | Suburban | | | Park | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0.25m/2° | 0.5m/5° | 5m/10° | 0.25m/2° | 0.5m/5° | 5m/10° | 0.25m/2° | 0.5m/5° | 5m/10° | 0.25m/2° | 0.5m/5° | 5m/10° |
| Triplet (original NetVLAD) [2] | 6.0 | 15.5 | 59.9 | 9.4 | 22.6 | 71.2 | 3.9 | 11.8 | 60.1 | 3.2 | 9.2 | 45.2 |
| Quadruplet [8] | 6.9 | 17.5 | 62.3 | 10.7 | 25.2 | 73.3 | 4.4 | 13.0 | 61.4 | 3.9 | 10.8 | 47.9 |
| Lazy triplet [1] | 6.4 | 16.5 | 58.6 | 9.9 | 23.5 | 69.8 | 4.1 | 11.9 | 58.2 | 3.5 | 10.1 | 42.0 |
| Lazy quadruplet [1] | 7.3 | 18.5 | 61.7 | 11.4 | 26.9 | 72.7 | 4.9 | 13.9 | 64.1 | 3.7 | 10.7 | 44.1 |
| Triplet + Huber distance [40] | 6.0 | 15.3 | 55.9 | 9.5 | 22.9 | 69.0 | 4.4 | 12.4 | 57.3 | 3.0 | 8.4 | 39.6 |
| Log-ratio [18] | 6.7 | 17.4 | 58.8 | 10.5 | 24.9 | 71.4 | 4.6 | 13.4 | 57.4 | 3.5 | 10.2 | 42.8 |
| Multi-similarity [44] | 7.4 | 18.8 | 66.3 | 12.0 | 28.8 | 81.6 | 5.1 | 14.6 | 63.9 | 3.8 | 10.9 | 52.7 |
| Soft contrastive [41] | 8.0 | 20.5 | **70.4** | 12.7 | 30.7 | **84.6** | 5.1 | 14.9 | **67.9** | 4.5 | 12.6 | **56.8** |
| **GCL (Ours)** | **9.2** | **22.8** | 65.8 | **14.7** | **34.2** | 82.6 | **5.7** | **16.1** | 64.6 | **5.1** | **14.0** | 49.1 |
| *GCL (Ours w/ ResNeXt)* | *9.9* | *24.3* | *75.5* | *15.4* | *36.0* | *89.6* | *6.7* | *18.4* | *76.8* | *5.6* | *14.9* | *60.3* |

Table 4. Comparison of localization results (on CMU Seasons) of VGG16 backbones trained with several metric loss functions. Methods in the upper part deploy a NetVLAD pooling layer.

table deploy a VGG16 backbone with a NetVLAD pooling layer and make use of hard-negative pair mining. We also use a VGG16 backbone and do not perform hard-negative pair mining, substantially reducing the training time and memory requirements. This allows us to also train backbones with larger capacity, e.g. ResNeXt, on a single V100 GPU in less than a day, of which we report the results in italics for completeness.

**Processing time.** The GCL function and the graded similarity labels contribute to training effective models in a data- and computation-efficient way, largely improving on the resources and time required to train NetVLAD (see Table 5). Our VGG16-GeM-GCL model obtains higher results than NetVLAD while requiring $6\times$ less memory and about $100\times$ less time to converge. We point out that NetVLAD is the backbone of several other methods for VPR such as Patch-NetVLAD, DELG and SuperGlue in Table 2, thus making the comparison in Table 5 relevant from a larger perspective. The graded similarity and GCL function contribute to an efficient use of training data. A single epoch, i.e. a model sees a certain training pair only once, is sufficient for the training of GCL-based models to converge. The low memory and time requirements also enable the training of models with larger backbones, that obtain very high results while still keeping the resource usage low. We point out the data-efficient training that we deployed can stimulate further and faster progress in VPR, as it enables to train larger backbones, and perform extensive hyperparameter optimization or more detailed ablation studies.

| Model | memory | epochs | t/epoch | t/converge |
|---|---|---|---|---|
| NetVLAD-16-TL | 9.67GB | 30 (22) | 24h | 22d |
| NetVLAD-64-TL | - | 10 (7) | 36h | 10.5d |
| VGG16-GeM-GCL | 1.49GB | 1 | 5h | 5h |
| ResNet50-GeM-GCL | 1.65 GB | 1 | 6h | 6h |
| ResNet152-GeM-GCL | 3.78 GB | 1 | 14h | 14h |
| ResNetXt-GeM-GCL | 4.77 GB | 1 (1/2) | 28h | 14h |

Table 5. Training time and GPU memory utilization for a batch size of 4 images. In the epochs column, the number in parenthesis is the number of epochs until convergence.

## 6. Conclusions

We extended the learning of image descriptors for visual place recognition by using measures of camera pose similarity and 3D surface overlap as proxies for graded image pair similarity to re-annotate existing VPR datasets (i.e. MSLS, 7Scenes and TB-Places). We demonstrated that the new labels can be used to effectively compose training batches without the need of hard-pair mining, decisively speeding-up training time while reducing memory requirements. Furthermore, we reformulated the Contrastive Loss function, proposing a Generalized Contrastive Loss (GCL). The GCL exploits the graded similarity of image pairs, and contributes to learning way better performing image descriptors for VPR than those of other losses that are not designed to use graded image similarity labels (i.e. triplet, quadruplet and their variants) and that require hard-pair mining during training. Models trained with the GCL and new graded similarity labels obtain comparable or higher results that several existing VPR methods, including those that apply re-ranking of the retrieved images, while keeping a more efficient use of the data, training time and memory. We achieved good generalization to unseen environments, showing robustness to domain shifts on the Pittsburgh30k, Tokyo 24/7, RobotCar Seasons v2 and Extended CMU Seasons datasets. The combination of graded similarity annotations and a loss function that can embed them in the training paves a way to learn more effective descriptors for VPR in a data- and resource-efficient manner.

## References


[1] Mikaela Angelina Uy and Gim Hee Lee. Pointnetvlad: Deep point cloud based retrieval for large-scale place recognition. In *CVPR*, pages 4470–4479, 2018.

[2] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly

supervised place recognition. In *CVPR*, pages 5297–5307, 2016.

[3] Hernan Badino, Daniel Huber, and Takeo Kanade. The CMU Visual Localization Data Set. `http://3dvis.ri.cmu.edu/data-sets/localization`, 2011.

[4] Vassileios Balntas, Shuda Li, and Victor Prisacariu. Relocnet: Continuous metric learning relocalisation using neural nets. In *ECCV*, pages 751–767, 2018.

[5] Gabriele Berton, Carlo Masone, and Barbara Caputo. Rethinking visual geo-localization for large-scale applications, 2022.

[6] Gabriele Berton, Riccardo Mereu, Gabriele Trivigno, Carlo Masone, Gabriela Csurka, Torsten Sattler, and Barbara Caputo. Deep visual geo-localization benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5396–5407, June 2022.

[7] Bingyi Cao, André Araujo, and Jack Sim. Unifying deep local and global features for image search. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *ECCV 2020*, pages 726–743, 2020.

[8] Weihua Chen, Xiaotang Chen, Jianguo Zhang, and Kaiqi Huang. Beyond triplet loss: A deep quadruplet network for person re-identification. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1320–1329, 2017.

[9] Wei Chen, Yu Liu, Weiping Wang, Erwin M. Bakker, Theodoros Georgiou, Paul W. Fieguth, Li Liu, and Michael S. Lew. Deep image retrieval: A survey. *CoRR*, abs/2101.11282, 2021.

[10] Zetao Chen, Stephanie Lowry, Adam Jacobson, Zongyuan Ge, and Michael Milford. Distance metric learning for feature-agnostic place recognition. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2556–2563, 2015.

[11] Agnés Desolneux, Lionel Moisan, and Jean-Michel Morel. Gestalt theory. In *From Gestalt Theory to Image Analysis: A Probabilistic Approach*, pages 11–30. Springer New York, 2008.

[12] Yixiao Ge, Haibo Wang, Feng Zhu, Rui Zhao, and Hongsheng Li. Self-supervising fine-grained region similarities for large-scale image localization. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 369–386, 2020.

[13] Albert Gordo, Jon Almazán, Jerome Revaud, and Diane Larlus. End-to-end learning of deep visual representations for image retrieval. *International Journal of Computer Vision*, 124(2):237–254, 2017.

[14] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *CVPR*, volume 2, pages 1735–1742. IEEE, 2006.

[15] Stephen Hausler, Sourav Garg, Ming Xu, Michael Milford, and Tobias Fischer. Patch-netvlad: Multi-scale fusion of locally-global descriptors for place recognition. In *CVPR*, pages 14141–14152, 2021.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[17] Sungyeon Kim, Dongwon Kim, Minsu Cho, and Suha Kwak. Embedding transfer with label relaxation for improved metric learning. In *CVPR*, pages 3967–3976, 2021.

[18] Sungyeon Kim, Minkyo Seo, Ivan Laptev, Minsu Cho, and Suha Kwak. Deep metric learning beyond binary supervision. In *CVPR*, pages 2288–2297, 2019.

[19] Maria Leyva-Vallina, Nicola Strisciuglio, Manuel López-Antequera, Radim Tylecek, Michael Blaich, and Nicolai Petkov. Tb-places: A data set for visual place recognition in garden environments. *IEEE Access*, 2019.

[20] María Leyva-Vallina, Nicola Strisciuglio, and Nicolai Petkov. Place recognition in gardens by learning visual representations: data set and benchmark analysis. In *CAIP*, pages 324–335. Springer, 2019.

[21] Chundi Liu, Guangwei Yu, Maksims Volkovs, Cheng Chang, Himanshu Rai, Junwei Ma, and Satya Krishna Gorti. Guided similarity separation for image retrieval. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32, 2019.

[22] Liu Liu, Hongdong Li, and Yuchao Dai. Stochastic attraction-repulsion embedding for large scale image localization. In *CVPR*, pages 2570–2579, 2019.

[23] Manuel Lopez-Antequera, Ruben Gomez-Ojeda, Nicolai Petkov, and Javier Gonzalez-Jimenez. Appearance-invariant place recognition by discriminatively training a convolutional neural network. *Pattern Recognit. Lett.*, 92:89–95, 2017.

[24] Stephanie Lowry, Niko Sünderhauf, Paul Newman, John J Leonard, David Cox, Peter Corke, and Michael J Milford. Visual place recognition: A survey. *IEEE Transactions on Robotics*, 32(1):1–19, 2016.

[25] Will Maddern, Geoffrey Pascoe, Chris Linegar, and Paul Newman. 1 year, 1000 km: The oxford robotcar dataset. *Int. J. Robot. Res.*, 36(1):3–15, 2017.

[26] Michael J Milford and Gordon F Wyeth. Seqslam: Visual route-based navigation for sunny summer days and stormy winter nights. In *ICRA*, pages 1643–1649, 2012.

[27] Guohao Peng, Yufeng Yue, Jun Zhang, Zhenyu Wu, Xiaoyu Tang, and Danwei Wang. Semantic reinforced attention learning for visual place recognition. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13415–13422, 2021.

[28] Noé Pion, Martin Humenberger, Gabriela Csurka, Yohann Cabon, and Torsten Sattler. Benchmarking image retrieval for visual localization. In *International Conference on 3D Vision*, 2020.

[29] Filip Radenović, Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondřej Chum. Revisiting oxford and paris: Large-scale image retrieval benchmarking. In *CVPR*, pages 5706–5715, 2018.

[30] Filip Radenović, Giorgos Tolias, and Ondřej Chum. Fine-tuning cnn image retrieval with no human annotation. *TPAMI*, 41(7):1655–1668, 2018.

[31] Jerome Revaud, Jon Almazán, Rafael S Rezende, and Cesar Roberto de Souza. Learning with average precision: Training image retrieval with a listwise loss. In *ICCV*, pages 5107–5116, 2019.

[32] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: Learning feature matching with graph neural networks. In *CVPR*, 2020.

[33] Torsten Sattler, Will Maddern, Carl Toft, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, et al. Benchmarking 6dof outdoor visual localization in changing conditions. In *CVPR*, pages 8601–8610, 2018.

[34] Torsten Sattler, Tobias Weyand, Bastian Leibe, and Leif Kobbelt. Image retrieval for image-based localization revisited. In *BMVC*, volume 1, page 4, 2012.

[35] Torsten Sattler, Qunjie Zhou, Marc Pollefeys, and Laura Leal-Taixé. Understanding the limitations of cnn-based absolute camera pose regression. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3297–3307, 2019.

[36] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[37] Hailin Shi, Yang Yang, Xiangyu Zhu, Shengcai Liao, Zhen Lei, Weishi Zheng, and Stan Z Li. Embedding deep metric for person re-identification: A study against large variations. In *ECCV*, pages 732–748. Springer, 2016.

[38] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene coordinate regression forests for camera relocalization in rgb-d images. In *CVPR*, pages 2930–2937, 2013.

[39] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *ICLR*, 2015.

[40] Janine Thoma, Danda Pani Paudel, Ajad Chhatkuli, and Luc Van Gool. Geometrically mappable image features. *IEEE Robotics and Automation Letters*, 5(2):2062–2069, 2020.

[41] Janine Thoma, Danda P Paudel, and Luc Van Gool. Soft contrastive learning for visual localization. *NeurIPS*, 2020.

[42] A. Torii, J. Sivic, M. Okutomi, and T. Pajdla. Visual place recognition with repetitive structures. *TPAMI*, 2015.

[43] Ruotong Wang, Yanqing Shen, Weiliang Zuo, Sanping Zhou, and Nanning Zhen. Transvpr: Transformer-based place recognition with multi-level attention aggregation. In *CVPR*, 2022.

[44] Xun Wang, Xintong Han, Weilin Huang, Dengke Dong, and Matthew R. Scott. Multi-similarity loss with general pair weighting for deep metric learning. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5017–5025, 2019.

[45] Frederik Warburg, Søren Hauberg, Manuel López-Antequera, Pau Gargallo, Yubin Kuang, and Javier Civera. Mapillary street-level sequences: A dataset for lifelong place recognition. In *CVPR*, 2020.

[46] Tobias Weyand, Andre Araujo, Bingyi Cao, and Jack Sim. Google landmarks dataset v2-a large-scale benchmark for instance-level recognition and retrieval. In *CVPR*, pages 2575–2584, 2020.

[47] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *CVPR*, pages 1492–1500, 2017.

[48] Mubariz Zaffar, Sourav Garg, Michael Milford, Julian Kooij, David Flynn, Klaus McDonald-Maier, and Shoaib Ehsan. Vpr-bench: An open-source visual place recognition evaluation framework with quantifiable viewpoint and appearance change. *International Journal of Computer Vision*, 129(7):2136–2174, 2021.