GSV-CITIES: TOWARD APPROPRIATE SUPERVISED VISUAL PLACE RECOGNITION

Amar Ali-bey
Université Laval
Québec, Canada

Brahim Chaib-draa
Université Laval
Québec, Canada

Philippe Giguère
Université Laval
Québec, Canada

ABSTRACT

This paper aims to investigate representation learning for large scale visual place recognition, which consists of determining the location depicted in a query image by referring to a database of reference images. This is a challenging task due to the large-scale environmental changes that can occur over time (i.e., weather, illumination, season, traffic, occlusion). Progress is currently challenged by the lack of large databases with accurate ground truth. To address this challenge, we introduce GSV-CITIES, a new image dataset providing the widest geographic coverage to date with highly accurate ground truth, covering more than 40 cities across all continents over a 14-year period. We subsequently explore the full potential of recent advances in deep metric learning to train networks specifically for place recognition, and evaluate how different loss functions influence performance. In addition, we show that performance of existing methods substantially improves when trained on GSV-CITIES. Finally, we introduce a new fully convolutional aggregation layer that outperforms existing techniques, including GeM, NetVLAD and CosPlace, and establish a new state-of-the-art on large-scale benchmarks, such as Pittsburgh, Mapillary-SLS, SPED and Nordland. The dataset and code are available for research purposes at https://github.com/amaralibey/gsv-cities.

Keywords Visual place recognition · Place recognition dataset · Visual geo-localization · Deep metric learning

1 Introduction

Visual place recognition (VPR) can be defined as the ability for a system to determine whether the location depicted in a query image has already been visited [2]. This is done by referring to a database of images of previously-visited locations, and comparing the query against them. VPR has been used in many applications. For instance, it is used in mobile robotics for localization [7, 34] and navigation [32, 33]. In particular, it is used to perform loop closure detection in SLAM algorithms [45, 12]. It is also used in image geo-localization [53].

Traditional VPR approaches were based on hand-crafted features such as SIFT [30]. These can then be further summarised into a single vector representation for the entire image such as Fisher Vectors [19, 37], Bag of Words [38, 49, 8] or VLAD [20, 3]. With the rise of deep learning, convolutional neural networks (CNNs) [25] showed impressive performance on several computer vision tasks, including image classification [14], object detection [28], and semantic segmentation [24]. For VPR, Sünderhauf *et al.* [44] showed that features extracted from intermediate layers of CNNs trained for image classification can perform better than hand-crafted features. As a result, researchers have proposed to train CNNs directly for the task of place recognition [59], demonstrating great success at large scale benchmarks [49, 52]. However, deep neural networks are data hungry, requiring large amount of training data. This has led to several datasets

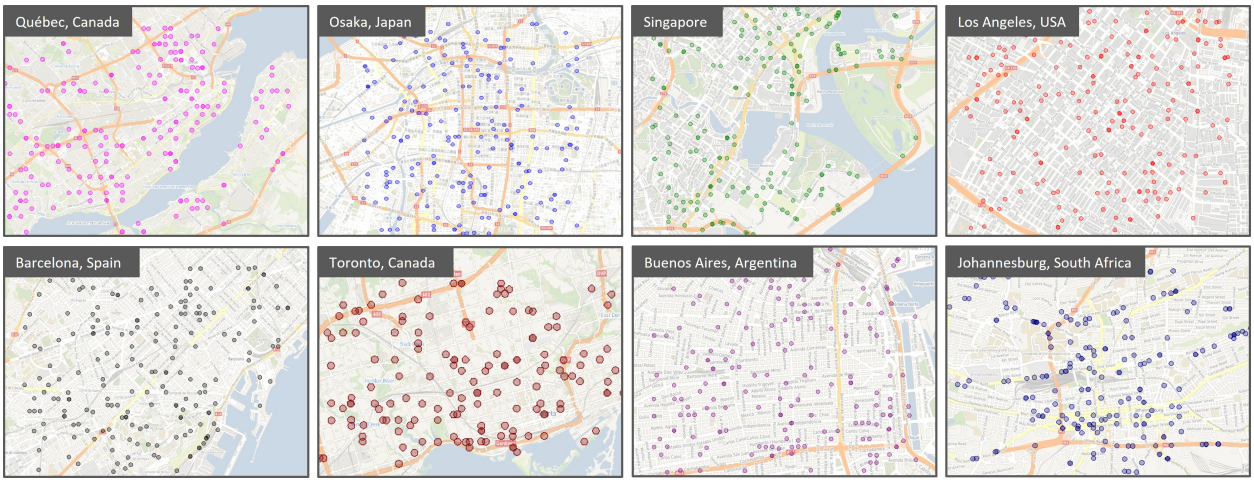


Figure 1: Sample locations in 8 major cities (among 40) in GSV-CITIES dataset. All locations are geographically distant and distributed nearly uniformly in every city, maximizing appearance diversity in urban and sub-urban areas. Each location (here, a point) is depicted by at least four images. See details in section 3.1.

being released specifically for the training and evaluation of deep neural networks for place recognition. Nevertheless, they all lacked at least one of the following aspects:

(i) *Geographical coverage*: most existing datasets are collected in small areas ranging from a city scale [31] to a small neighborhood [10] or a limited number of locations monitored by a surveillance cameras [6] which make them insufficient for training at large scale.

(ii) *Accurate ground truth*: except for small-scale datasets, all large-scale ones lack accurate ground truth, which is essential for supervised learning.

(iii) *Perceptual diversity*: some datasets do not provide enough appearance variations that can be encountered in real-world applications, such as viewpoint [6] or structural changes [36].

To the best of our knowledge, there are no datasets that provide precise ground truth, enough environmental diversity and wide geographical coverage, to enable the training of neural networks for place recognition at large scale *and* with full supervision.

The main challenge in training neural networks for place recognition resides in how to learn discriminative representations, such that images depicting the same place get similar representations while those depicting different places get dissimilar ones. So far, most state-of-the-art techniques rely on contrastive or triplet loss functions to supervise the representation learning process [59]. This consists of feeding images to the network in form of positive pairs (a pair of images depicting the same place) and negative pairs (a pair of images depicting different places) and optimize the network parameters under a constraint that maximizes the similarity between instances representing the same place or minimizes their similarity in the other case.

In order to form a negative pair, one can simply choose two distant images (e.g., 50 meters or more apart) based on their GPS coordinates. However, to form a positive pair, one cannot solely rely on GPS coordinates, as there is no guarantee that the two images are facing the same direction and therefore may not represent the same place. For instance, Arandjelovic *et al.* [2] developed a training procedure, based on a weakly-supervised triplet loss, to enable training from weakly-labeled images. As such, weak supervision is used to compensate for the lack of accurate ground truth in the dataset. Inspired by this work, most recent techniques [2, 22, 29, 9] relied on weakly-supervised loss functions to train on widely available geotagged images. Despite many advances in visual place recognition, current state-of-the-art techniques are still hindered by the bottleneck of inaccurate ground truth, and thus weak supervision remained their best option.

In this work, we consider representation learning for place recognition as a three components pipeline as shown in Fig. 4 and introduce the following contributions:

1. **Data module:** we introduce GSV-CITIES, a new large-scale dataset with a wide variety of perceptual changes over a 14-year period, covering 40 cities spread across all continents. This dataset provides *highly accurate ground truth* allowing for straightforward mini-batches sampling eliminating the bottleneck of weak supervision.

2. **Image representation:** in addition to GSV-CITIES we propose a new fully convolutional aggregation layer (Conv-AP), that generates highly efficient representations while significantly outperforming existing SotA techniques such as GeM [39], NetVLAD [2] and CosPlace [4].

3. **Online hard mining and parameters learning:** given the accurate ground truth of GSV-Cities, we enable online hard samples mining, combined with various SotA metric learning loss functions. By doing so, we show that sophisticated loss functions such as Multi-Similarity [51] can greatly improve performance for visual place recognition.

2 Related Works

In this section, we first review existing place recognition methods, then provide a summary of existing datasets that are used for the training and evaluation of such techniques. Finally, we highlight the key issue with weak supervision, which is associated with imprecise ground truth in existing datasets.

2.1 Place recognition

The problem of visual place recognition has long been framed as an image retrieval task [2, 22, 29, 42, 9], where the location of a query image is determined according to the locations of the most relevant images retrieved from a reference database. As for many other computer vision tasks, CNNs trained specifically for place recognition have shown considerable success [6, 50, 55, 2, 9]. In general, CNNs pre-trained on image classification datasets are adapted to VPR by cropping them at the last convolution layer and plugging a trainable aggregation layer that effectively aggregates feature maps into discriminative representations. For instance, Arandjelovic *et al.* [2] developed an end-to-end trainable version of VLAD descriptor [3], which can be plugged into a CNN to aggregate deep features into one descriptor. Following the success of NetVLAD, many variants have been introduced. For example, Kim *et al.* [22] proposed a method that integrates feature re-weighting into the NetVLAD descriptor. More recently, Yu *et al.* [56] proposed SPE-VLAD, a technique that incorporates pyramid structure into NetVLAD, their aim was to enhance NetVLAD with both spatial and regional features from the images. In another work, Zhang *et al.* [58] proposed two variants, Weighted and Gated NetVLAD, with the latter performing better on VPR tasks. The gating mechanism was applied on each VLAD residual vector to incorporate its personalized characteristics.

Alternative techniques to NetVLAD include R-MAC [47] that consists of extracting Region of Interest (RoI) directly from the CNN feature maps to form representations, and Generalized Mean (GeM) [39] which is a trainable generalized form of global pooling. Recently, Berton *et al.* [4] introduced CosPlace which combines GeM with a linear projection layer showing great performance boost compared to existing techniques.

In this article, we propose a new efficient aggregation technique that we call Conv-AP. It performs channel-wise pooling on the feature maps followed by spatial-wise adaptive pooling (as described in section 3.3), making the architecture fully convolutional and the output dimensionality highly configurable. Conv-AP achieves SotA results on all five benchmarks while being $16\times$ more compact than NetVLAD.

2.2 Place recognition datasets

Recently, several datasets have been released to train and evaluate different techniques of place recognition. In table 1 we summarize some of the relevant ones. **SPED** [6] was collected from 2.5k surveillance cameras. It provides accurate ground truth and seasonal changes. However, it is geographically limited and provides no viewpoint changes. **Nordland** [36] was recorded in 4 seasons with a camera mounted on a train. While this dataset provides accurate ground truth and severe seasonal changes, it lacks viewpoint and urban structural changes. **Oxford RobotCar** [31] contains over 100 trips of the same 10 km route in the city of Oxford, UK. It provides highly-accurate ground truth including 3D point cloud. Although this dataset comprises a lot of perceptual changes and contains 20M images, its geographical coverage is very limited ($< 2\,\text{km}^2$) compared to others. **Pitts250k** [49] and **TokyoTM** [48] are among

Dataset name	Geo. span	Panoramas	Places	Images	Time span	Accurate GT	Viewpoint	Season
Nordland [36]	–	0	–	–	1 year	✓	✗	✓
SPED [6]	–	0	∼2.5K	2.5M	7 months	✓	✗	✓
Oxford RobotCar [31]	< 2 km^2	–	–	**20M**	18 months	✓	✓	✓
Pittsburg 250k [49]	~ 16 km^2	∼10K	–	0.25M	–	✗	✓	✗
TokyoTM [48]	–	∼31K	–	0.19M	–	✗	✓	✗
Mapillary SLS [52]	–	–	∼23K	1.68M	7 years	✗	✓	✓
GSV-CITIES (Ours)	**2000 km^2**	**∼560K**	**∼67K**	0.56M	**14 years**	✓	✓	✓

Table 1: Comparison of datasets for large-scale place recognition.

the most used datasets for training and evaluating place recognition techniques. They have been generated from panoramas downloaded from Google Street View. Although they feature significant viewpoint variations and accurate GPS coordinates, they do not provide viewing directions for the images (bearing). Therefore, it is impossible to determine which images depict the same location, based solely on their provided GPS coordinates (positive pairs are almost impossible to form off-the-shelf). Recently, Warburg *et al.* [52] introduced **Mapillary Street Level Sequences (MSLS)**, a large-scale dataset that covers 30 cities around the world and includes challenging variations in viewpoint, season and illumination. While this dataset provides the viewing direction for each image, it lacks GPS accuracy because the sequences are sourced from Smartphone and Dashcam users. We also note that almost all images in MSLS are forward-facing, causing the road to always appear in the center of the image. Finally, **GSV-CITIES** (ours) includes highly-accurate GPS coordinates *and* viewing direction for each image (see details in section 3.1) which makes it straightforward to form positive (and negative) pairs.

2.3 Limitations associated with current datasets: weak supervision

Current state-of-the art techniques [2, 22, 42, 29, 9, 26, 13] rely on datasets of geotagged images for training. However, in such datasets, images are indexed by their GPS coordinates, and split into queries and references. Due to the noisy and weak GPS labels, there is no prior knowledge of the positive matches that are available for each training query. Therefore, most works train the network with weakly supervised loss functions [2] that rely on tuples consisting of a query image, a set of potential positives (images nearby the query according to GPS coordinates) and a set of definite negatives. From an optimization perspective, weakly-supervised loss functions tend to force each query to be closer to its already closest positive (i.e. easiest positive) [9], precluding proper supervision for learning robust feature representations. In this work, we leverage the accurate labels provided by our new dataset to efficiently train networks for place recognition in a supervised manner using existing deep metric loss functions where *difficult positives* are highly effective for learning robust representations [21].

3 Methodology

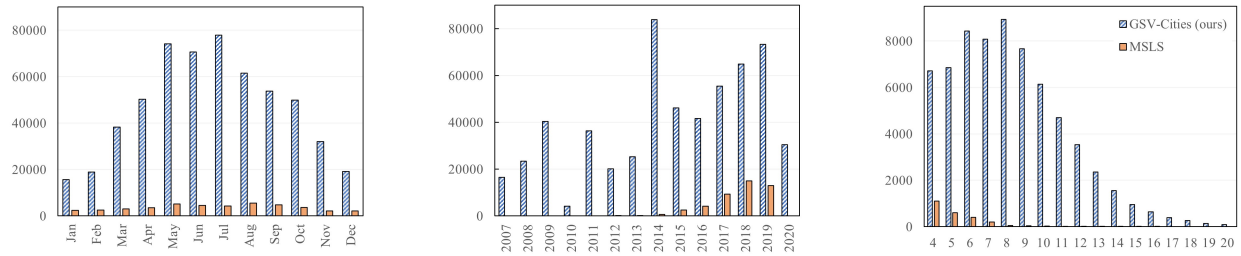
We argue that current visual place recognition methods suffer from the bottleneck of weak supervision. To this end, we collect a new dataset that provides highly accurate labels and enables full supervision. Subsequently, we demonstrate, through extensive experiments, that the performance of existing techniques can significantly be increased.

Below we present our data collection approach and an overview of the collected dataset. Then, we propose a new efficient and fully convolutional aggregation layer for visual place recognition.

3.1 Data collection and overview

Following [2, 48], we make use of Google Street View Time Machine (GSV-TM) to collect panoramic images depicting the same place over time (between 2007 and 2021). According to [1, 23], Google Street View combines several types of sensor measurements, including LiDAR data, to correct the position of the collected panoramas. Consequently, the global coordinates provided by Google Street View are of high quality. To maximize geographical diversity, we selected 40 cities spread across all continents. Then, for each city, we uniformly queried GSV-TM in interval of $0.001°$ latitude and longitude ($\sim 100 - 130$ meters). This ensured that all locations are physically distant, thus avoiding any overlap

Figure 2: Three examples of places in GSV-CITIES. Each place is depicted by a set of images (here six, in a row) representing the same physical location. As such, they are indexed by the same ID. The number of images depicting one place in GSV-CITIES varies from 4 up to 20.



(a) Yearly distribution

(b) Monthly distribution

(c) # of images per location

Figure 3: Distribution of images in GSV-CITIES (Blue stripped) versus MSLS (Orange), on a yearly and monthly scales. (c) shows the number of images *taken at different dates* in each place (e.g., GSV-CITIES contains over $8,000$ places that are depicted by 7 perspectives each).

(Fig. 1 shows location sampled in 8 different cities). Importantly, only the locations that were visited *at least four times* were retained.

We then retrieved each location's historical panoramas and, *very importantly*, their bearings (the direction according to the north pole). This allowed to generate perspectives capturing the exact same place, but at different times (e.g., in Fig. 2 we show three places, each depicted by 6 perspectives). We carried out qualitative verification without finding any failure. We compare GSV-CITIES (ours) with Mapillary-SLS in Fig. 3. We show the yearly and monthly distribution of images in Fig. 3a and Fig. 3b respectively. Clearly, GSV-CITIES is far richer and more diverse than MSLS.

GSV-CITIES contains over $67,000$k places, each of which is described by a set of images captured at 4 to 20 different dates, as can be seen in Fig. 3c. In comparison, MSLS contains $\approx 1,400$ places with images captured at 4 different dates (versus $6,600$ in GSV-CITIES) and $\approx$ 300 places with images captured at 6 different dates (versus $8,300$ in GSV-CITIES). This makes GSV-CITIES by far the *widest* and *most diverse* dataset in terms of time span and geographical scope. The geographical diversity of GSV-CITIES is exemplified in Fig. 1, with locations sampled from 8 different cities. In total, our dataset covers more than $2,000$ km^2.

Finally our dataset is organized as follows: we define a place P_i as a set of images $\{I^i\}$ depicting the same physical location $P_i = \left(\left\{I^i_1, I^i_2, ..., I^i_{k_i}\right\}, y_i\right)$, where y_i is a an ID assigned to each place. In some sense, y_i can be interpreted as the target label. Our database $\mathcal{D}$ is then the set of all these geographically-distant places $\{P_i\}$, namely $\mathcal{D} = \{P_1, P_2, ..., P_N\}$.

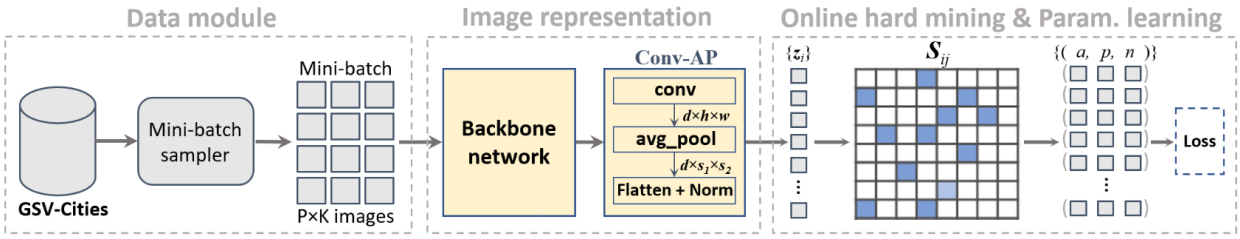


Figure 4: Framework of representation learning for visual place recognition. Our dataset, GSV-CITIES, makes it straightforward to construct batches comprised of P places each of which depicted by K images. The neural network (backbone + aggregation layer) computes a representation ($\mathbf{z}_i$) for each image in the batch. The matrix $\mathbf{S}$ comprises pairwise similarity between all instances in the batch, which are used to mine informative pairs (or triplets as in this example) in an online fashion. The loss function operates on these pairs/triplets to minimize an objective that maximizes the similarity between instances of the same place and minimizes that of different places.

3.2 Framework of representation learning for visual place recognition

Inspired by recent advances in deep metric learning [51], we want to learn place specific representations in a standardized manner. Our aim is thus to learn a mapping function $f_\theta : \mathcal{D} \subseteq \mathbb{R}^S \mapsto \Phi \subseteq \mathbb{R}^D$ represented by a neural network (backbone + aggregation layer in Fig. 4) parameterized by θ, which projects images $I_i \in \mathcal{D}$ from a source space $\mathbb{R}^S$ (RGB space in our case) into a representation space Φ where the similarity S_{ij} between two instances $(\mathbf{z}_i, \mathbf{z}_j)$ is higher if the pair is positive ($\mathbf{z}_i$ and $\mathbf{z}_j$ represent the same place) and lower if it is negative ($\mathbf{z}_i$ and $\mathbf{z}_j$ represent two different places). S_{ij} is then defined as:

$$S_{ij} = \text{sim}\left(f_\theta\left(I_i\right), f_\theta\left(I_j\right)\right) = \text{sim}\left(\mathbf{z}_i, \mathbf{z}_j\right) \tag{1}$$

where $\text{sim}\left(\cdot\,,\cdot\right)$ represents the cosine similarity. For regularization purposes, the resulting representations are normalized to the real hypersphere $\mathbb{S}^D$ [54], in which the cosine similarity S_{ij} between two instances $(\mathbf{z}_i, \mathbf{z}_j)$ becomes the inner product of their representations. With this standardized formulation, we can learn the parameters θ (i.e., train the network) using existing pair-based loss functions in a fully supervised fashion by leveraging the accurate labels of GSV-CITIES.

3.3 Fully convolutional feature aggregation

Following existing techniques, we use pre-trained networks as backbones cropped at the last convolutional layer. The output of the backbone is a dense 3D tensor, called feature maps $\mathbf{F} \in \mathbb{R}^{h \times w \times c}$, where $h \times w$ is the spatial resolution and c is the number of channels, which corresponds to the numbers of filters in the last convolutional layer of the backbone. $\mathbf{F}$ can be interpreted as a set of c-dimentional descriptors $\mathbf{f}_{ij} \in \mathbf{F}$ at each $h \times w$ spatial location.

We introduce a new fully convolutional feature aggregation technique, called Conv-AP that operates as follows. First, given a feature maps $\mathbf{F}$, we apply a channel-wise weighted pooling that linearly projects each spatial descriptor $\mathbf{f}_{ij} \in \mathbf{F}$ into a compact d-dimensional representation space, enabling dimensionality reduction along the channel dimension. This can be implemented using a 1×1 convolution with parameters $\mathbf{W} \in \mathbb{R}^{d \times c \times 1 \times 1}$ such as:

$$\mathbf{F}' = \mathbf{W} \circledast \mathbf{F} = \texttt{Conv}_{1\times1}(\mathbf{F}) \tag{2}$$

where $\circledast$ is the convolution operation and $\mathbf{F}' \in \mathbb{R}^{h \times w \times d}$ is the resulting feature maps with depth d. Second, we reduce the spatial dimensionality of $\mathbf{F}'$ using adaptive average pooling (`AAP`), which spatially splits the feature maps into $s_1 \times s_2$ equal sub-regions and takes their average, resulting in feature maps of size $s_1 \times s_2 \times d$. In other words, `AAP` effectively determines the stride and window size of the average pooling operation, to obtain feature maps of fixed spatial size. Note that global average pooling [27] is a special case of `AAP` where $s_1 = s_2 = 1$.

Formally, our technique, Conv-AP, can be summarized as follows. Given an image I_i, we pass it through the pre-trained backbone to obtain its feature maps $\mathbf{F}_i$, and pass them to the aggregation layer to obtain the final representation $\mathbf{z}_i$, such as:

$$\mathbf{z}_i = \texttt{AAP}_{s_1 \times s_2}\left(\,\texttt{Conv}_{1\times1}\left(\,\mathbf{F}_i\,\right)\right) \tag{3}$$

Finally, the resulting representation $\mathbf{z}_i \in \mathbb{R}^{s_1 \times s_2 \times d}$ is flattened and L_2-normalized, in order to conform to the training framework in section 3.2.

3.3.1 Loss function

Existing VPR techniques [2, 22, 29, 52] train the network in a weakly supervised manner by feeding it tuples $(q, \mathbf{P}^q, \mathbf{N}^q)$, each of which consists of a query q, a subset of potential positives $\mathbf{P}^q$ and a subset of hard negatives $\mathbf{N}^q$. The images in $\mathbf{P}^q$ are geographically close to the query q (≤ 10 meters), but they don't necessarily depict the same place (as they might not have the same orientation as q). In this case, weakly supervised triplet loss [2] is used, where only the easiest positive from $\mathbf{P}^q$ is used to form a positive pair, as such:

$$\mathcal{L}_{\text{w-triplet}} = \sum_{n_j \in \mathbf{N}^q} \left[\overbrace{\text{sim}\left(q, n_j\right)}^{\text{negative pairs}} - \overbrace{\max_{p_i \in \mathbf{P}^q} \text{sim}\left(q, p_i\right)}^{\text{the positive pair}} + m \right]_+ \tag{4}$$

where m is a margin used to exclude trivial triplets and $[\bullet]_+ = \max(\bullet, 0)$ is the hinge function. Intuitively, from all images in $\mathbf{P}^q$ the one most similar to q is the most likely to depict the same place as the query, and is hence used as the positive pair.

In this work, we follow a different approach by taking advantage of the accurate labels provided by GSV-CITIES and adopt a formulation similar to [15], where batches are formed by sampling P places (P different IDs) from the dataset, and randomly picking K images of each place (GSV-CITIES guarantees that the K images will depict the same place), thus resulting in batches of size $P \times K$ (as seen in Fig. 4). This formulation allows us to use pair-based and triplet-based loss functions in a supervised manner such as (1) Contrastive loss [11], (2) Triplet ranking loss [16] and (3) Multi-Similarity loss [51].

Contrastive loss: this function [11] aims at maximizing the similarity between positive pairs and minimizing it for negative pairs as follows:

$$\mathcal{L}_{\text{contrast}} = (1 - \mathcal{I}_{ij}) \left[S_{ij} - m\right]_+ - \mathcal{I}_{ij} S_{ij} \tag{5}$$

where $\mathcal{I}_{ij} = 1$ indicates a positive pair, and 0 otherwise. The margin m avoids optimizing indefinitely the negative pairs that are already dissimilar enough.

Triplet margin loss: this function [16] can be calculated on a triplet $(\mathbf{z}_i, \mathbf{z}_j, \mathbf{z}_k)$ as follows:

$$\mathcal{L}_{\text{triplet}} = \left[S_{ij} - S_{ik} + m\right]_+ \tag{6}$$

where the objective is to reduce the similarity S_{ik} of the negative pair $(\mathbf{z}_i, \mathbf{z}_k)$ and at the same time increase the similarity S_{ij} of the positive pair $(\mathbf{z}_i, \mathbf{z}_j)$.

Multi-Similarity loss: Wang *et al.* [51] recently introduced this loss function which incorporates advanced pair weighting schemes and can be calculated as follows:

$$\mathcal{L}_{\text{MS}} = \frac{1}{N} \sum_{i=1}^{N} \left\{ \frac{1}{\alpha} \log \left[1 + \sum_{j \in \mathcal{P}_i} e^{-\alpha(S_{ij} - m)} \right] + \frac{1}{\beta} \log \left[1 + \sum_{k \in \mathcal{N}_i} e^{\beta(S_{ik} - m)} \right] \right\} \tag{7}$$

where for each instance $\mathbf{z}_i$ in the batch, $\mathcal{P}_i$ represents the set of indices $\{j\}$ that form a positive pair with $\mathbf{z}_i$ and $\mathcal{N}_i$ the set of indices $\{k\}$ that form a negative pair with $\mathbf{z}_i$. α, β and m represent constants (hyperparameters) that control the weighting scheme (refer to [51] for more details).

Finally, there are many other pair/triplet based loss functions in metric learning literature [35, 21] that we can use for training, such as FastAP [5] and Circle loss [43].

3.3.2 Online hard mining

Current techniques that rely on weakly supervised triplet loss [2, 22, 29, 52] employ offline hard negative mining, which consists of retrieving, for each query, the most difficult negatives among all (or a subset of) images in the dataset. This is done to compensate for the drawback of easy positives used in the weakly supervised loss function (Eq. 4).

Nonetheless, offline mining is computationally expensive where the training is paused and representations of a large subset of images are computed and stored into a cache memory.

In this work, we are using batches of size $P \times K$, where each sample can play the role of query, positive and negative at the same time. This formulation can generate a large number of pairs (or triplets), many of which can be uninformative. Yet choosing informative positive and negative pairs within the batch is crucial to robust feature learning. An informative pair is one that produces a large loss value, thereby pushing the network to learn discriminative representations. Thus, an effective mining strategy is able to increase not only performance, but also the training speed [21]. We opt for online mining which, unlike offline mining, does not induce a lot of computation since it is performed on the fly on each batch at the end of the forward pass (as shown in Fig. 4). Many online mining strategies have been proposed in the literature [35]. For example, Online Hardest Mining (OHM) strategy [15] consists of keeping only the most difficult positive and the most difficult negative for each instance in the batch. Recently, Wang *et al.* [51] proposed a pair mining strategy that considers multiple pairwise similarities in the mining process, demonstrating great boost of performance.

In summary, the proposed framework (as shown in Fig. 4) makes it straightforward to train neural networks for place recognition without resorting to weak supervision and offline hard example mining, while also accelerating training time by orders of magnitude and improving performance of existing techniques as we show in section 4.2.

4 Experiments

In this section we describe the datasets and the metrics used for training and evaluation (section 4.1), and then show how training with GSV-CITIES improves performance of existing techniques (section 4.2). Next, we evaluate how different metric loss functions [35] perform for training place recognition networks (section 4.3). We then compare our new aggregation method (Conv-AP) to existing state-of-the-art techniques on multiple large-scale benchmarks (section 4.4). We also show some extended implementation details (4.5) and experiment how different backbone architectures affect performance (section 4.6). Finally, we show the impact of applying PCA dimensionality reduction (section 4.7).

4.1 Datasets and metrics

For evaluation, we use the following 4 benchmarks, Pitts250k-test [49], MSLS [52], SPED [57] and Nordland [57]. They contain respectively, 8k, 750, 607 and 1622 query images, and 83k, 19k, 607 and 1622 reference images. We follow the same evaluation metric of [2, 52, 57], where the Recall@k is measured. For Pitts250k and MSLS benchmarks, the query image is determined to be successfully retrieved if at least one of the top-k retrieved reference images is located within $d = 25$ meters from the query (according to their GPS coordinates).

4.1.1 Implementation details

By default, we use ResNet50 [14] as backbone pre-trained on ImageNet [40] and cropped at the last residual bloc, extended with an aggregation layer. For NetVLAD [2] we fix the number of clusters to 16, resulting in 32k-dimensional representations as in [2, 13]. Unless otherwise stated, for our method (Conv-AP) we fix the depth of the channel-wise pooling operation (the 1×1 convolution) to $d = 2048$.

For data organization, we use batches of size $P = 100$ places, each one being represented by $K = 4$ images, resulting in batches of size 400. Stochastic gradient descent (SGD) is utilized for optimization, with momentum 0.9 and weight decay 0.001. The initial learning rate of 0.03 is multiplied by 0.3 after every 5 epochs. We train for a maximum of 30 epochs using images resized to 320×320.

4.2 Importance of training with GSV-Cities

To assess the benefits of training with our dataset, we compare the performance of three existing methods when trained on either Pitts30k-train [49], MSLS-train [52] or GSV-CITIES.

Table 2 reports the performance of NetVLAD [2], GeM [39] and AVG (which represents global average pooling), trained on Pitts30k, MSLS or GSV-CITIES. As we can see, using GSV-CITIES for training drastically improves performance of all three techniques across all benchmarks. Training AVG on GSV-CITIES instead of MSLS improved its recall@1 performance (in percentage points) by 15.7 on Pitts250k, 14.2 on MSLS, 4.1 on SPED and 10.9 on

Table 2: Training on GSV-CITIES versus MSLS and Pittsburg. Performance of AVG (global average pooling), GeM (Generalized Mean) and NetVLAD are reported.

Method	Training dataset	Pitts250k-test			MSLS-val			SPED			Nordland		
		R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
AVG	MSLS [52]	62.6	82.7	88.4	59.3	71.9	75.5	54.7	72.5	77.1	4.4	8.4	10.4
	GSV-Cities (ours)	**78.3**	**89.8**	**92.6**	**73.5**	**83.9**	**85.8**	**58.8**	**77.3**	**82.7**	**15.3**	**27.4**	**33.9**
GeM [39]	MSLS [52]	72.3	87.2	91.4	65.1	76.8	81.4	55.0	70.2	76.1	7.4	13.5	16.6
	GSV-Cities (ours)	**82.9**	**92.1**	**94.3**	**76.5**	**85.7**	**88.2**	**64.6**	**79.4**	**83.5**	**20.8**	**33.3**	**40.0**
NetVLAD [2]	Pitts30k	86.0	93.2	95.1	59.5	70.4	74.7	71.0	87.1	90.4	4.1	6.6	8.2
	MSLS [52]	48.7	70.6	78.9	48.6	63.4	70.5	37.9	56.0	64.9	2.4	5.0	6.6
	GSV-Cities (ours)	**90.5**	**96.2**	**97.4**	**82.6**	**89.6**	**92.0**	**78.7**	**88.3**	**91.4**	**32.6**	**47.1**	**53.3**

Nordland. Note that the relative improvement on Nordland is 240%. We also report the performance of training GeM on GSV-CITIES versus MSLS, where its recall@1 improved by, respectively, 10.6, 10.4, 9.4 and 13.4 percentage points, which is clearly a significant boost of performance.

Most interestingly, NetVLAD drastically increases in performance when trained on GSV-CITIES, ($86.0\% \rightarrow 90.5\%$) on Pitts250k , ($59.5\% \rightarrow 82.6\%$) on MSLS-val, and ($4.1\% \rightarrow 32.6\%$) on Nordland. This highlights the importance of the accurate ground truth of our dataset. Note that when NetVLAD is trained on MSLS, it reaches convergence after **55 days** of training (reported by the authors of [13]), compared to **8 hours** of training on GSV-CITIES, which translates to 99.4% less training time (in other words, *it takes less time to train NetVLAD 165 times on GSV-Cities than to train it once on MSLS*). This large difference is due to the fact that using MSLS for training requires a lot of offline mining which incurs significant computational overhead, whereas training on GSV-CITIES relies on online mining, which is much faster and requires no additional computation.

4.3 Comparing different loss functions

In this experiment, we carried out comparisons between five different metric learning loss functions for training place recognition networks: (1) Contrastive loss [11]; (2) Triplet loss [15]; (3) FastAP loss [5]; (4) Circle loss [43]; and (5) Multi-Similarity loss [51]. PyTorch implementation of these loss functions (and many others) can be found in [35]. For training, we used a subset of 20k places from GSV-CITIES and trained the network for a maximum of 15 epochs. We evaluated on two challenging benchmarks, Pitts30k-test [49] and MSLS-val [52].

As shown in Table 3, performance vary depending on the loss function used for the training. Multi-Similarity loss [51] achieves the best results on both benchmarks, which might be explained by its sophisticated pair mining and weighting strategy. This shows the significance of the training loss function for place recognition networks. Notice how the contrastive loss improves to second best when coupled with a sophisticated mining strategy, which highlights the importance of online informative sample mining during the training. We believe GSV-CITIES paves the way for new research into VPR-specific loss functions.

Table 3: Comparing different metric learning loss functions for training place recognition networks. MS-miner is the pair mining strategy used in Multi-Similarity loss, and OHM is a strategy that uses only the most difficult triplets in the batch. A subset of 20k places from GSV-CITIES was used for the training.

Loss function	Pitts30k-test			MSLS-val		
	R@1	R@5	R@10	R@1	R@5	R@10
Contrastive [11]	86.7	94.0	95.7	67.8	79.3	83.5
Contrastive [11] + MS-miner	87.8	94.5	96.0	71.8	80.7	84.1
Triplet [15] + OHM	85.2	93.0	95.1	60.4	72.6	76.1
FastAP [5]	87.0	93.6	95.5	67.7	80.3	82.8
Circle [43]	86.9	94.3	95.8	72.2	82.8	85.5
Multi-Similarity [51]	**89.2**	**95.1**	**96.4**	**76.9**	**85.7**	**88.0**

Table 4: Comparison with state-of-the-art approaches on 4 place recognition benchmarks. Note that all models are trained on GSV-CITIES and directly evaluated on Pitts250k, MSLS, SPED and Nordland datasets. Conv-AP$_{s\times s}$ is our method, with s representing the height and width of the adaptive average pooling operation.

Method	Pitts250k-test			MSLS-val			SPED			Nordland		
	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
AVG [2]	78.3	89.8	92.6	73.5	83.9	85.8	58.8	77.3	82.7	15.3	27.4	33.9
GeM [39]	82.9	92.1	94.3	76.5	85.7	88.2	64.6	79.4	83.5	20.8	33.3	40.0
NetVLAD [2]	90.5	96.2	97.4	82.6	89.6	92.0	78.7	88.3	91.4	32.6	47.1	53.3
SPE-VLAD [56]	89.9	96.1	97.3	78.2	86.8	88.8	73.1	85.5	88.7	25.5	40.1	46.1
Gated NetVLAD [58]	89.7	95.9	97.1	82.0	88.9	91.4	75.6	87.1	90.8	34.4	50.4	57.7
CosPlace [4]	91.5	96.9	97.9	83.0	89.9	91.8	75.3	85.9	88.6	34.4	49.9	56.5
Conv-AP$_{1\times1}$ (ours)	90.5	96.2	97.5	80.3	89.6	91.6	75.0	86.8	90.3	25.8	40.8	46.8
Conv-AP$_{2\times2}$ (ours)	**92.4**	97.4	98.4	**83.4**	**90.5**	**92.3**	80.1	90.3	93.6	**38.2**	**54.8**	**61.2**
Conv-AP$_{3\times3}$ (ours)	92.2	**97.6**	**98.6**	83.2	89.2	91.1	80.9	90.3	93.4	34.8	50.1	56.2
Conv-AP$_{4\times4}$ (ours)	92.2	97.4	98.3	80.1	88.6	90.3	**81.2**	**90.3**	**93.9**	34.3	48.6	55.8

4.4 Comparison to state-of-the-art

In this section, we compare the performance of our proposed aggregation method (Conv-AP) to existing techniques. We test four variants of Conv-AP$_{s\times s}$, by varying the size s of the adaptive average pooling operation. For fair comparison, we train AVG [2], GeM [39], NetVLAD [2], SPE-VLAD [56], Gated NetVLAD [58], CosPlace [4] and Conv-AP on GSV-CITIES dataset, using the exact same configurations and hyperparameters.

Experimental results are shown in Table 4. As we can see, our method achieves substantially higher results than existing state-of-the-art on all benchmarks, reaching a new state-of-the-art on Pitts250k (92.4%), MSLS (83.4%), SPED (81.2%) and Nordland (38.2%).

Overall, Conv-AP$_{2\times2}$, which uses an adaptive average pooling window of size 2×2 (spatially splits the feature maps into four equally sub-regions), obtains the best results, beating NetVLAD and CosPlace on every benchmark. Furthermore, it is interesting to note that when $s = 1$ (Conv-AP$_{1\times1}$) we see a relative performance drop, we believe this is due to the spatial dimension being collapsed to 1×1, potentially resulting in the loss of any spatial order present in the feature maps (see section 4.5 for more experimental details).

Finally, we did not perform re-ranking of the top retrieved candidates as done in Patch-NetVLAD [13] and SuperGlue [41]. This is a technique known to boost recall@k by running a second matching pass that performs spatial verification of the local features. That being said, our method (Conv-AP) still outperforms Patch-NetVLAD and SuperGlue by a large margin. For instance, on MSLS-val, Conv-AP achieves recall@1 that is 3.9 points higher than Patch-NetVLAD and 5 points higher than SuperGlue. Moreover, on the Mapillary Challenge, Conv-AP outperforms Patch-NetVLAD by 10.4 points (68.0% vs 57.6% recall@5).

4.5 Further analysis

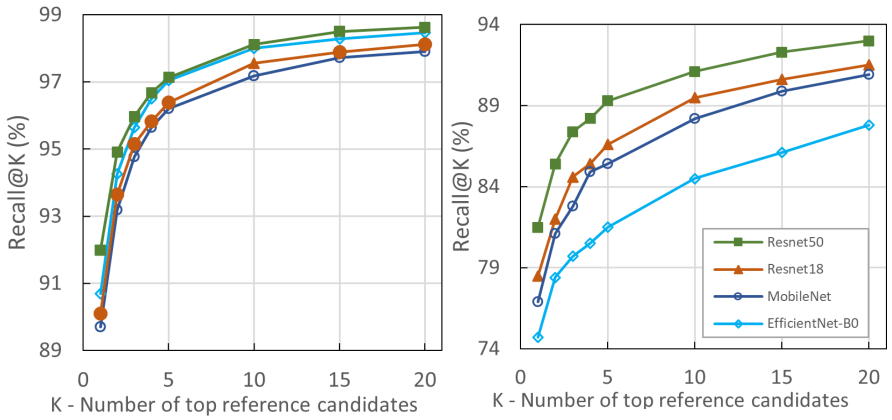
In this section, we investigate the robustness of our method with respect to its hyperparameters. Conv-AP produces representations of size $d\times s_1\times s_2$ (that are flattened and L_2-normalized). Each parameter influences the size of the output and most likely performance. In Table 5, we conduct comprehensive experiments to show how the depth d and the spatial size s of Conv-AP affect performance. We note by Conv$_d$-AP$_{s\times s}$ each variant. We observe that reducing the depth d does not necessarily result in lower performance. For instance, reducing the depth from $d = 2048$ to $d = 256$ produces 8 times smaller representations with negligible effect on overall performance. This illustrates how Conv-AP can be configured to generate highly efficient representations without performance loss. For example, Conv$_{512}$-AP$_{2\times2}$ outperforms NetVLAD while being $16\times$ more compact (2048-D vs 32768-D). Moreover, Conv$_{2048}$-AP$_{1\times1}$ and Conv$_{512}$-AP$_{2\times2}$ both produce 2048-D representations. However, the latter outperforms the former on both benchmarks. This confirms our hypothesis in section 4.4, which states that collapsing the spatial dimension to 1×1 may lead to a loss of spatial order in the resulting representations causing drop in performance.

Table 5: Performance of Conv$_d$-AP$_{s\times s}$ where d is the depth of the channel-wise pooling and s is the size of the adaptive average pooling operation. ResNet50 is used as backbone. Output dimension represents the size of the resulting flattened representations.

Conv-AP hyperparams.		Output dimension	Pitts250k-test			MSLS-val		
d	$s_1\times s_2$		R@1	R@5	R@10	R@1	R@5	R@10
256	1×1	256	89.7	96.1	97.6	80.4	88.5	90.5
512		512	90.5	96.5	97.8	80.4	88.4	90.8
1024		1024	90.4	96.0	97.4	81.8	89.7	90.9
2048		2048	90.5	96.2	97.5	80.3	89.6	91.6
256	2×2	1024	91.9	97.4	98.2	81.4	89.5	92.2
512		2048	92.0	97.4	98.2	82.7	90.4	92.8
1024		4096	92.3	97.4	98.4	83.4	90.5	92.3
2048		8192	92.4	97.4	98.4	82.2	89.1	90.9

4.6 Backbone architectures

We also compare between different backbone architectures, EfficientNet [46], ResNet [14] and MobileNet [17]. Fig. 5 shows that compact backbones, such as ResNet18, MobileNet and EfficientNet-B0, can achieve top-notch performance on challenging benchmarks while requiring small memory footprint. ResNet50 obtains the best overall performance, especially on MSLS.



(a) Pitts250k-test (b) MSLS-val

Figure 5: Performance of Conv-AP coupled with different backbone architectures. All models have been trained on GSV-CITIES.

4.7 Dimensionality reduction

Most state-of-the-art techniques use Principal Component Analysis (PCA) and Whitening [18] to reduce the dimension of the resulting representations in order to obtain compact image descriptors suitable for efficient storage. Although, our method (Conv-AP) can be configured to generate highly compact representations out of the box (e.g., by fixing $d = 128$ and $s = 2$, we obtain 512-D outputs), we nevertheless perform PCA for fair comparison. All techniques are trained on GSV-CITIES, and PCA is learned on a subset of 10k images. Fig. 6 shows recall@1 performance on Pitts250k-test. Our method outperforms all other techniques for any dimension size. For instance, 512-D Conv-AP still outperforms 2048-D NetVLAD while being $4\times$ more compact. Furthermore, these results, in direct agreement with those in Table 5, show that Conv-AP can generate highly efficient representations out of the box with minimal performance degradation.

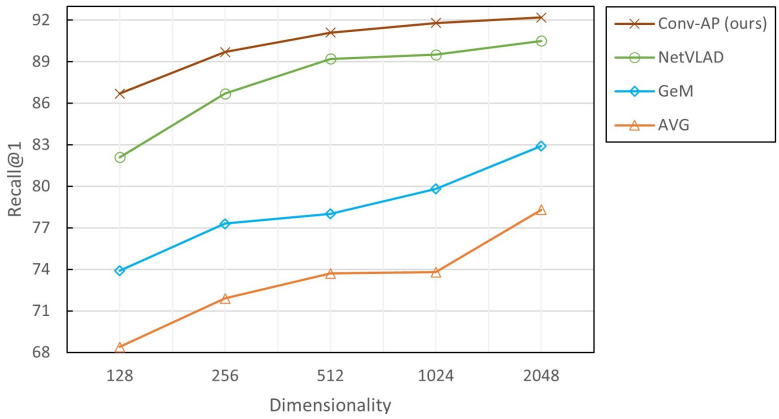


Figure 6: Recall@1 performance on Pitts250k-test after PCA dimensionality reduction. Note the log-scale of the x-axis. Conv-AP convincingly outperforms all other techniques. 512-D Conv-AP performs better than $4\times$ larger 2048-D NetVLAD.

5 Conclusion

In this paper, we introduced GSV-CITIES, a large-scale dataset (560k images from 67k locations) for appropriate training of visual place recognition methods. Importantly, the highly accurate ground truth of GSV-CITIES eliminated the bottleneck of weak supervision that is currently limiting existing techniques, while improving their performance as well as drastically reducing training time. Capitalizing on that, we showed that metric learning loss function can improve performance of VPR techniques when accurate labels are provided. We believe that this paves the way for further research into place recognition-specific architectures and loss functions. Finally, we introduced Conv-AP, a fully convolutional aggregation method that significantly outperforms existing techniques. In this context, we established new state-of-the-art on the challenging Pitts250k-test, MSLS-val, SPED and Nordland benchmarks.

Acknowledgement. This work has been partially supported by Natural Sciences and Engineering Research Council of Canada (NSERC), The Fonds de Recherche du Québec Nature et technologies (FRQNT). We gratefully acknowledge the support of NVIDIA Corporation with the donation of a GPU used for our experiments.

References

[1] Dragomir Anguelov, Carole Dulong, Daniel Filip, Christian Frueh, Stéphane Lafon, Richard Lyon, Abhijit Ogale, Luc Vincent, and Josh Weaver. Google street view: Capturing the world at street level. *Computer*, 43(6):32–38, 2010.

[2] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. NetVLAD: CNN architecture for weakly supervised place recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5297–5307, 2016.

[3] Relja Arandjelovic and Andrew Zisserman. All about VLAD. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1578–1585, 2013.

[4] Gabriele Berton, Carlo Masone, and Barbara Caputo. Rethinking visual geo-localization for large-scale applications. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4878–4888, 2022.

[5] Fatih Cakir, Kun He, Xide Xia, Brian Kulis, and Stan Sclaroff. Deep metric learning to rank. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1861–1870, 2019.

[6] Zetao Chen, Adam Jacobson, Niko Sünderhauf, Ben Upcroft, Lingqiao Liu, Chunhua Shen, Ian Reid, and Michael Milford. Deep learning features at scale for visual place recognition. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3223–3230, 2017.

[7] Sean Philip Engelson. *Passive Map Learning and Visual Place Recognition*. PhD thesis, Yale University, 1994.

[8] Dorian Gálvez-López and Juan D Tardos. Bags of binary words for fast place recognition in image sequences. *IEEE Transactions on Robotics*, 28(5):1188–1197, 2012.

[9] Yixiao Ge, Haibo Wang, Feng Zhu, Rui Zhao, and Hongsheng Li. Self-supervising fine-grained region similarities for large-scale image localization. In *European Conference on Computer Vision (ECCV)*, pages 369–386. Springer, 2020.

[10] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research (IJRR)*, 32(11):1231–1237, 2013.

[11] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 1735–1742, 2006.

[12] Christian Häne, Lionel Heng, Gim Hee Lee, Friedrich Fraundorfer, Paul Furgale, Torsten Sattler, and Marc Pollefeys. 3D visual perception for self-driving cars using a multi-camera system: Calibration, mapping, localization, and obstacle detection. *Image and Vision Computing*, 68:14–27, 2017.

[13] Stephen Hausler, Sourav Garg, Ming Xu, Michael Milford, and Tobias Fischer. Patch-netvlad: Multi-scale fusion of locally-global descriptors for place recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14141–14152, 2021.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[15] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*, 2017.

[16] Elad Hoffer and Nir Ailon. Deep metric learning using triplet network. In *International Workshop on Similarity-Based Pattern Recognition*, pages 84–92. Springer, 2015.

[17] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1314–1324, 2019.

[18] Hervé Jégou and Ondřej Chum. Negative evidences and co-occurences in image retrieval: The benefit of pca and whitening. In *European conference on computer vision*, pages 774–787. Springer, 2012.

[19] Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez. Aggregating local descriptors into a compact image representation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3304–3311, 2010.

[20] Herve Jegou, Florent Perronnin, Matthijs Douze, Jorge Sánchez, Patrick Perez, and Cordelia Schmid. Aggregating local image descriptors into compact codes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(9):1704–1716, 2011.

[21] Mahmut Kaya and Hasan Şakir Bilge. Deep metric learning: A survey. *Symmetry*, 11(9):1066, 2019.

[22] Hyo Jin Kim, Enrique Dunn, and Jan-Michael Frahm. Learned contextual feature reweighting for image geo-localization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3251–3260, 2017.

[23] Bryan Klingner, David Martin, and James Roseborough. Street view motion-from-structure-from-motion. In *IEEE International Conference on Computer Vision (ICCV)*, pages 953–960, 2013.

[24] Fahad Lateef and Yassine Ruichek. Survey on semantic segmentation using deep learning techniques. *Neurocomputing*, 338:321–348, 2019.

[25] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[26] María Leyva-Vallina, Nicola Strisciuglio, and Nicolai Petkov. Generalized contrastive optimization of siamese networks for place recognition. *arXiv preprint arXiv:2103.06638*, 2021.

[27] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *arXiv preprint arXiv:1312.4400*, 2013.

[28] Li Liu, Wanli Ouyang, Xiaogang Wang, Paul Fieguth, Jie Chen, Xinwang Liu, and Matti Pietikäinen. Deep learning for generic object detection: A survey. *International Journal of Computer Vision*, 128(2):261–318, 2020.

[29] Liu Liu, Hongdong Li, and Yuchao Dai. Stochastic attraction-repulsion embedding for large scale image localization. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2570–2579, 2019.

[30] David G Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[31] Will Maddern, Geoffrey Pascoe, Chris Linegar, and Paul Newman. 1 year, 1000 km: The Oxford RobotCar dataset. *The International Journal of Robotics Research (IJRR)*, 36(1):3–15, 2017.

[32] Yoshio Matsumoto, Masayuki Inaba, and Hirochika Inoue. Visual navigation using view-sequenced route representation. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 1, pages 83–88, 1996.

[33] Michael Milford and Gordon Wyeth. Persistent navigation and mapping using a biologically inspired slam system. *The International Journal of Robotics Research*, 29(9):1131–1153, 2010.

[34] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D. Tardos. ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015.

[35] Kevin Musgrave, Serge Belongie, and Ser-Nam Lim. Pytorch metric learning. *arXiv preprint arXiv:2008.09164*, 2020.

[36] Daniel Olid, José M Fácil, and Javier Civera. Single-view place recognition under seasonal changes. *arXiv preprint arXiv:1808.06516*, 2018.

[37] Florent Perronnin, Yan Liu, Jorge Sánchez, and Hervé Poirier. Large-scale image retrieval with compressed fisher vectors. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3384–3391, 2010.

[38] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2007.

[39] Filip Radenović, Giorgos Tolias, and Ondřej Chum. Fine-tuning CNN image retrieval with no human annotation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(7):1655–1668, 2018.

[40] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[41] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4938–4947, 2020.

[42] Zachary Seymour, Karan Sikka, Han-Pang Chiu, Supun Samarasekera, and Rakesh Kumar. Semantically-aware attentive neural embeddings for long-term 2D visual localization. In *British Machine Vision Conference (BMVC)*, 2019.

[43] Yifan Sun, Changmao Cheng, Yuhan Zhang, Chi Zhang, Liang Zheng, Zhongdao Wang, and Yichen Wei. Circle loss: A unified perspective of pair similarity optimization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6398–6407, 2020.

[44] Niko Sünderhauf, Sareh Shirazi, Feras Dayoub, Ben Upcroft, and Michael Milford. On the performance of convnet features for place recognition. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4297–4304, 2015.

[45] Takafumi Taketomi, Hideaki Uchiyama, and Sei Ikeda. Visual SLAM algorithms: a survey from 2010 to 2016. *IPSJ Transactions on Computer Vision and Applications*, 9(1):1–11, 2017.

[46] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114. PMLR, 2019.

[47] Giorgos Tolias, Ronan Sicre, and Hervé Jégou. Particular object retrieval with integral max-pooling of cnn activations. *arXiv preprint arXiv:1511.05879*, 2015.

[48] Akihiko Torii, Relja Arandjelovic, Josef Sivic, Masatoshi Okutomi, and Tomas Pajdla. 24/7 place recognition by view synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1808–1817, 2015.

[49] Akihiko Torii, Josef Sivic, Tomas Pajdla, and Masatoshi Okutomi. Visual place recognition with repetitive structures. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 883–890, 2013.

[50] Tsun-Hsuan Wang, Hung-Jui Huang, Juan-Ting Lin, Chan-Wei Hu, Kuo-Hao Zeng, and Min Sun. Omnidirectional cnn for visual place recognition and navigation. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2341–2348. IEEE, 2018.

[51] Xun Wang, Xintong Han, Weilin Huang, Dengke Dong, and Matthew R Scott. Multi-similarity loss with general pair weighting for deep metric learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5022–5030, 2019.

[52] Frederik Warburg, Soren Hauberg, Manuel López-Antequera, Pau Gargallo, Yubin Kuang, and Javier Civera. Mapillary street-level sequences: A dataset for lifelong place recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2626–2635, 2020.

[53] Tobias Weyand, Ilya Kostrikov, and James Philbin. Planet-photo geolocation with convolutional neural networks. In *European Conference on Computer Vision (ECCV)*, pages 37–55. Springer, 2016.

[54] Chao-Yuan Wu, R Manmatha, Alexander J Smola, and Philipp Krahenbuhl. Sampling matters in deep embedding learning. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2840–2848, 2017.

[55] Peng Yin, Lingyun Xu, Xueqian Li, Chen Yin, Yingli Li, Rangaprasad Arun Srivatsan, Lu Li, Jianmin Ji, and Yuqing He. A multi-domain feature learning method for visual place recognition. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 319–324. IEEE, 2019.

[56] Jun Yu, Chaoyang Zhu, Jian Zhang, Qingming Huang, and Dacheng Tao. Spatial pyramid-enhanced netvlad with weighted triplet loss for place recognition. *IEEE transactions on neural networks and learning systems*, 31(2):661–674, 2019.

[57] 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*, pages 1–39, 2021.

[58] Jian Zhang, Yunyin Cao, and Qun Wu. Vector of locally and adaptively aggregated descriptors for image feature representation. *Pattern Recognition*, 116:107952, 2021.

[59] Xiwu Zhang, Lei Wang, and Yan Su. Visual place recognition: A survey from deep learning perspective. *Pattern Recognition*, 113:107760, 2021.