EigenPlaces: Training Viewpoint Robust Models for Visual Place Recognition

Gabriele Berton*[1] Gabriele Trivigno*[1] Barbara Caputo[1] Carlo Masone[1]
[1] Politecnico di Torino
{gabriele.berton, gabriele.trivigno, barbara.caputo, carlo.masone}@polito.it

Abstract

Visual Place Recognition is a task that aims to predict the place of an image (called query) based solely on its visual features. This is typically done through image retrieval, where the query is matched to the most similar images from a large database of geotagged photos, using learned global descriptors. A major challenge in this task is recognizing places seen from different viewpoints. To overcome this limitation, we propose a new method, called EigenPlaces, to train our neural network on images from different point of views, which embeds viewpoint robustness into the learned global descriptors. The underlying idea is to cluster the training data so as to explicitly present the model with different views of the same points of interest. The selection of this points of interest is done without the need for extra supervision. We then present experiments on the most comprehensive set of datasets in literature, finding that EigenPlaces is able to outperform previous state of the art on the majority of datasets, while requiring 60% less GPU memory for training and using 50% smaller descriptors. The code and trained models for EigenPlaces are available at https://github.com/gmberton/EigenPlaces, while results with any other baseline can be computed with the codebase at https://github.com/gmberton/auto_VPR.

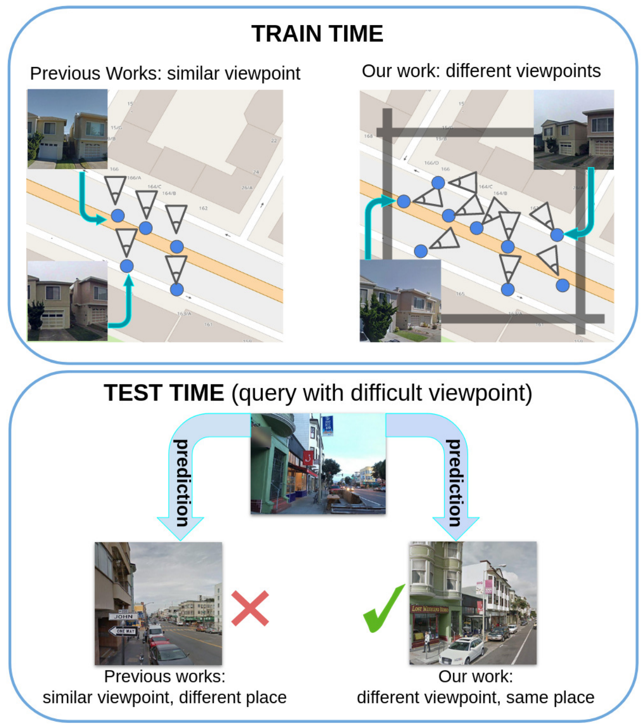

Figure 1. Most previous works [5, 31, 27, 61] train their models through metric learning, using as positive the most similar image to the query, which naturally would have same or similar orientation to the query. Other works split the dataset in classes, with images within a class having similar [1, 2] or exactly the same [8] orientation. EigenPlaces goes against this trend, creating classes in which all images are oriented towards the same point, leading to viewpoint robust models able to correctly localize highly challenging queries, for example the ones collected from a sidewalk.

1. Introduction

Visual Place Recognition (VPR) is a task that aims to predict the place where a photo (*i.e.* query) was taken, quickly and accurately, based solely on its visual features. This is typically done with an image retrieval approach [51, 14, 27, 16, 5, 15, 50, 23, 20, 31, 19, 21, 26, 57, 53, 22, 25, 60, 36, 1, 8, 10, 2, 29]: first, a deep neural network is used to extract global descriptors from the query and from a database of geo-referenced images; then, a nearest neighbor search is performed in this features space [5, 31, 27, 21, 8, 2, 1]. While such approaches have shown great potential in partially solving known problems such as scalability (by using ever more compact descriptors [62, 8, 2]) and illumination changes (through the synthetic generation of night images [11, 41, 3] or strong data augmentation [21]), recognizing images under heavy viewpoint shifts is still an open challenge. A popular strategy to handle this problem is to follow up the similarity search performed on global feature descriptors with a post-processing phase that re-ranks the retrieved results using either spatial verification [12, 30, 22] or matching densely extracted local fea-

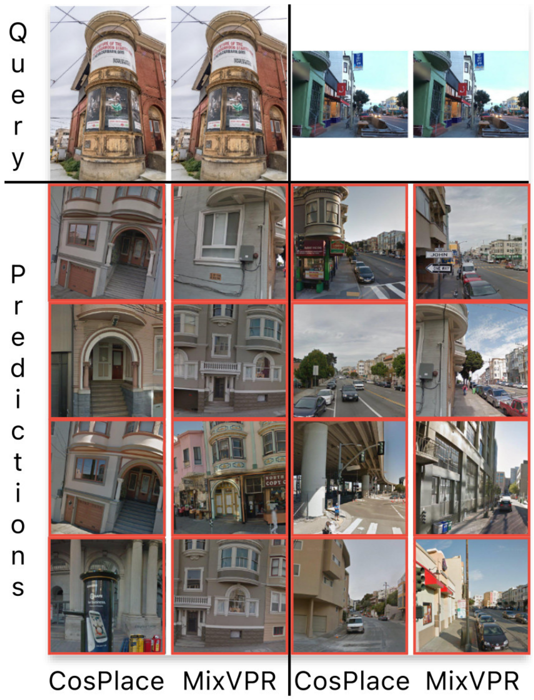

Figure 2. **Examples of 2 challenging queries** (top row) which present heavy viewpoint changes with respect to the database. The first and third columns are matches to the two query obtained with CosPlace [8], while the second and fourth column are matches obtained with MixVPR [2].

tures descriptors [9]. However, these post-processing methods are useless if the similarity search is not able to retrieve at least one matching result from the database. Moreover, these techniques are also costly, given that the local feature matching must be performed for each retrieved results, thus the number of candidates to be re-ranked is usually orders of magnitude smaller than the database [12, 9, 22, 30].

Even so, we can observe that when the queries to be recognized present heavy viewpoint shifts with respect to the database images, state-of-the-art retrieval architectures fail to find any relevant result in the highest ranked candidates (see Fig. 2). In view of these considerations, we argue that it is necessary to improve the robustness to viewpoint shifts already at the retrieval stage, by teaching the network to extract global descriptor extractor that are more invariant to perspective changes. To achieve this goal we propose EigenPlaces, a new training paradigm that clusters the training data into classes so that each class contains only multiple viewpoints depicting the same scene. This forces the model to learn global descriptors that are robust to viewpoint shifts (see Fig. 1). This is done by estimating the presence of *places* (such as building facades) based solely on the geographical distribution of training data, by splitting the training dataset in classes and finding the geographical principal components within each given class.

To empirically show the soundness of our method, we run a benchmark on the largest number of VPR datasets ever. The results show that a model trained with EigenPlaces is able to outperform previous SOTA on numerous datasets, while using 50% smaller descriptors and requiring 60% less GPU memory for training.

Our contributions are summarized as follows:

- we propose EigenPlaces, a novel training protocol whose ultimate goal is to render the model robust to viewpoint changes that it may encounter at test time;
- we perform a rigorous VPR benchmark on the most complete set of datasets in literature, to highlight not only the strengths but also the weaknesses of EigenPlaces and its predecessors;
- while our exploration shows that there is no one-win-all solution on every scenario, we note that EigenPlaces outperforms previous state of the art on a large number of datasets, while needing 60% less GPU memory to train and using 50% more compact descriptors.

The code and trained models for EigenPlaces are available at `https://github.com/gmberton/EigenPlaces`.

We also created and released a codebase to run experiments with a number of trained models (namely NetVLAD, SFRS, CosPlace, Conv-AP, MixVPR and EigenPlaces), which automatically downloads each model's weights from their official repository, to be able to run experiments within a fair and standardized framework. The codebase is available at `https://github.com/gmberton/auto_VPR`.

2. Related Work

Visual place recognition. Most early works on VPR focused on matching queries to their database counterparts through the use of local features [52], with methods such as SIFT [33], SURF [7] and RootSIFT [4] dominating the pre-deep learning landscape, although the use of global or patch features has also been investigated [38]. With the advent of deep learning, [6] found that features extracted with a CNN trained for classification can be successfully used for landmark retrieval. This inspired a number of following works, which used the same concept to extract global learned features with a number of pooling layers [43, 49, 42]. To ensure that the model learns to extract specific features for urban VPR, Arandjelovic *et al.* [5] proposed to train it on a dataset of StreetView images, while enhancing the CNN with a novel layer, named NetVLAD, that encodes 3D features maps to a highly informative vector. A number of subsequent works built on top of NetVLAD, enhancing it with an attention module [27], a novel loss [31], or a self-supervised strategy to crop training images [21]. A different training strategy was introduced by CosPlace [8], which takes inspiration from the face recognition literature [32, 54, 18] to train the model through a classification task, and then uses the same learned features to perform the retrieval. Recently, some works have also shown dramatic improvements training on the large-scale dataset of GSV-cities [1] using a Multi-Similarity loss [56], and processing the high-level features extracted by a CNN with a newly proposed Conv-AP layer [1] or a Feature-Mixer [2].

Despite the strides made by all these methods over the last years, none of them have explicitly addressed the viewpoint-invariance problem for visual place recognition. In particular, NetVLAD and its derivatives base their learning protocol on the use of the most similar database images to the query as positives, which are likely to share the same (or similar) viewpoint. MixVPR [2] uses a set of predefined images, grouped in classes, all of which share a similar viewpoint. Finally, in CosPlace all images in a class have exactly the same orientation by design.

Viewpoint invariant matching. There is also a separate body of literature dedicated to the refinement of a shortlist of candidates provided by an image retrieval module, by matching local features. Popular examples are SuperGlue [44], DELG [12], LoFTR [46], Patch-NetVLAD [22], GeoWarp [9], and others [55, 48]. These approaches are based on the premise that by leveraging local features associated to detected keypoints, it is possible to match landmarks even from different viewpoints.

Although these methods can mitigate the viewpoint shift problem, they all rely on the assumption that the shortlist of candidates resulting from retrieval stage contains at least a positive match. Our method is oriented towards improving the robustness of the retrieval stage, thus it is complementary to all these approaches. Moreover, since these techniques are computationally expensive, being able to retrieve a positive result is crucial to their applicability.

3. Method

Despite recent advances in the literature, substantial changes in viewpoint still represent a challenge even for modern SOTAs (see some examples in Fig. 2). In practice this kind of distribution shift is very common, because the database images for retrieval are usually collected via car-mounted cameras [51, 50, 34, 11, 1, 37, 13, 57], whereas the queries may come from different sources (*e.g.* photos taken by smartphones [50, 13, 8]) and can present a substantial variability in terms of viewpoint.

Knowing the positions of all *places* or *points of interest* (*e.g.* a building facade or architectural landmark) within the map, a straightforward approach to mitigate the issue could be to find all images of our train set that face towards them (*i.e.* from all viewpoint), and minimize a loss that pulls together features representing the same place. However, annotating the positions of all buildings in a city can be a challenging and expensive task, especially in a high density scenario, which would limit the scalability and practicality of a geolocalization system. To overcome this impediment, in this work we introduce a training algorithm that is able to automatically obtain different views of a given *place*, *i.e.* images that look at the same scene from different angles. Our novel method estimates the direction of a road using only the images' coordinates, and builds on the premise that points of interest lie on the side of the road.

In the following sections, we describe how we use EigenPlaces to train our networks:

- in Sec. 3.1 we explain how we split a dense dataset in non-overlapping cells, avoiding the risk of having images of the same place contained in different classes;
- in Sec. 3.2 we present how EigenPlaces selects a subset of images within a given cell, representing different views of the same place;
- in Sec. 3.3 we show the loss used to train the model using the selected images.

3.1. Map Partition

As a first preparatory step for EigenPlaces, we divide the map in $M \times M$ cells, with $M = 15$ $meters$. Next we group the cells in subsets, ensuring that within a single subset there are no neighboring cells. This guarantees that images within two cells of the same subset have no visual overlap, and thus cells within a subset can be later treated as classes for a classification task. To this end, we take inspiration from CosPlace [8], which has recently shown that a similar split can lead to good results on VPR. To build the subsets, we therefore take only one every N cells both in the latitudinal and longitudinal directions. Thus, we consider only $1/N^2$ of the cells at a time and during training we shift the set of cells after each epoch. Although this partitioning strategy is somewhat related to CosPlace's, our design relies only on the position of each image, and it does not entail the use of their orientation. Moreover, the rationale with which the classes are constructed from the cells is fundamentally different, and it is detailed in the next section.

3.2. EigenPlaces

Given a cell from the map partition, all the images therein represent the same location and may be naively considered as a unique class by a classifier. However, the images may be taken by cameras pointing in different directions, so they may observe different scenes. Thus, a model may struggle to learn a coherent representation for them all. In CosPlace, this problem is solved by dividing a cell in multiple classes, where each class contains all the images

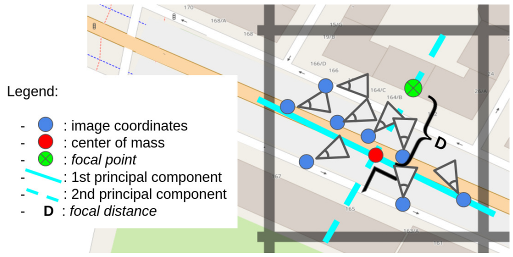

Figure 3. **Sketch of EigenPlaces's principle.** For a given cell, the first and second principal components are derived from the images' positions. The first is an estimate of a road, whereas along the second we can find *points of interest*, like facades. We choose a *focal point* on the second principal components, and then we find the images pointing towards it to represent different views of the same place. Using these different views to train a model endows it with robustness to viewpoint shifts. Note how the *focal distance* D defines the *focal point*: a $D = 0$ would lead to all the images pointing towards the center of mass of the images (*i.e*. images pointing towards each other), whereas a $D \to \infty$ makes the images pointing towards an infinitely far *focal point*, meaning that each image would have the same direction.

oriented in the same geographical direction. Our idea is instead to select in a class all the images that look at a same place, from different perspectives. To implement this idea without any extra supervision, we leverage the prior knowledge that database images are commonly collected by cameras mounted on cars [51, 50, 34, 11, 1, 37, 13, 57], *i.e*. they are aligned along roads. Following this idea, we can assume that we can find distinctive *points of interest* (*e.g*. building facades) by looking at the side of the roads.

To explain how we select these images, let us consider without lack of generality the i-th cell. Furthermore, let us denote as $X_i \in \mathbb{R}^{p\times 2}$, the matrix containing the UTM coordinates (east, north) of the p images in the cell. Then, we compute the Singular Value Decomposition (SVD) of the centered matrix $\widehat{X_i} = X_i - \mathrm{E}[X_i]$. Since $\widehat{X_i}$ is real-valued, we are guaranteed that the set of singular vectors obtained from the decomposition exists and is a orthonormal basis that can be ordered with respect to a set of non-negative singular values. Moreover, since the matrix is centered, the singular vectors are also the eigenvectors of the correlation matrix, *i.e*. the principal components.
The first eigenvector represents the direction of maximum variability in our data. As discussed before, this direction likely corresponds with the road traveled by the vehicle that collected the images. Therefore, the second eigenvector, perpendicular to the first one (and thus to the road), is likely directed to the side of the road. Consequently, we can define a focal point on the second principal component, likely towards a building's facade. Formally, we define the *focal point* c_i as :

$$c_i = \mathrm{E}[X_i] + D \times V_1 \tag{1}$$

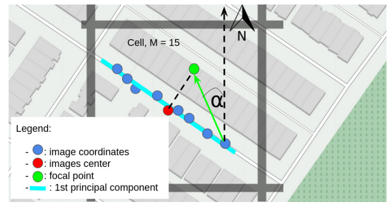

Figure 4. **Construction of a class in Eigenplaces.** To select a relevant subset of images, for each image we compute the angle α as shown in the figure. We then select only the images whose orientation is close to α.

where V_1 is the second principal component obtained from the SVD decomposition and D is a *focal distance* from the center of mass which determines the exact location of the *focal point* (see Fig. 3). Finally, within the given cell i, the images facing the *focal point* c_i are grouped in a single class and used effectively for training. Note the importance of the *focal distance* D for the construction of classes: when $D \to \infty$ the method selects all the images oriented in the same geographical direction (same orientation), whereas when $D \to 0$ the *focal point* gets closer to the mean of the images position and the method selects images facing in opposite directions.

This method assumes that the images available in the database are collected looking at all sides of the vehicle, and in particular towards the side of the street. However, this is not always the case and many VPR datasets: for example, the datasets built with autonomous driving applications in mind only contain images collected from a front facing camera (St Lucia [37], MSLS [57], SVOX [11], RobotCar [34]). In order to handle these cases, we repeat the same procedure to generate a second *focal point* along the first right eigenvector (the one aligned with the direction of the road) and create a second class from the front-facing images.

Although this method is built on the intuition that the images in a cell are likely aligned in a straight line along a road, this is certainly not true in general. For instance, at crossroads the images are distributed along multiple directions. In such cases, the eigenvectors obtained from the SVD are not aligned/orthogonal with the road, and the points of interest may end up not on buildings but somewhere else. Nevertheless, this does not detract from the method, as the goal is to feed the model with images looking at the same point from different perspectives. On the contrary, having some variability in the data so that not all points of interest are on buildings is helpful to make the model more robust.

3.3. Training

We now describe how to select images with changing viewpoints, once for a given class i we have obtained its *focal point* c_i according to Eq. (1). Given an image j and its UTM coordinates $x_j = (e_j, n_j)$:

$$\begin{aligned} c_i =& (e_c, n_c) \\ \Delta_{e_j} = e_c - e_j, & \Delta_{n_j} = n_c - n_j \\ \alpha_j =& \arctan(\frac{\Delta_{e_j}}{\Delta_{n_j}}) \end{aligned} \tag{2}$$

In practice, we select images whose orientation is closest to α_j. This computation is also exemplified in Fig. 4. The angle α_j depends on the relative positions of the image and the *focal point*, and it represents the *orientation*, *i.e*. the deviation w.r.t. the north axis. The sign of α_j will vary if the image lies on the left of the second principal component. Note that α_j varies for each of the images, and this is a key element in our approach, that allows to have the same *place* depicted from different viewpoints.

Once the dataset is split in classes, and a number of images are selected for each class, we can use such data to train in an end-to-end fashion a deep neural network. To this end, we use a Large Margin Cosine Loss (CosFace) [54], which has been shown to produce strong results in VPR [8]. The CosFace layer is defined by a single fully connected with weights matrix W^{lat}, and the loss is computed as follows:

$$\mathcal{L}_{lat} = \frac{1}{N} \sum_i -log \frac{e^{s(\cos(\theta_{y_i}) - m)}}{e^{s(\cos(\theta_{y_i}) - m)} + \sum_{i \neq j} e^{s \cos \theta_j}} \tag{3}$$

subject to

$$\begin{aligned} \cos \theta_j &= W_{lat_j}^T x_i \\ W_{lat} = \frac{W^*}{||W^*||}, & \quad x = \frac{x^*}{||x^*||} \end{aligned} \tag{4}$$

Since each cell has two different classes (one *lateral* and one *frontal*), as shown in Fig. 5, we employ two classifiers, one devoted to recognizing viewpoint shifts (with weights W_{lat}), and another one tasked with learning frontal-facing (w.r.t. the vehicle) views (with weights W_{front}). Thus our final loss comprises a *lateral* and a *frontal* component, each relying on a separate CosFace layer. The $\mathcal{L}_{front}$ has the same formulation as $\mathcal{L}_{lat}$. Finally, the final loss is:

$$\mathcal{L} = \mathcal{L}_{lat}(f, W_{lat}) + \mathcal{L}_{front}(f, W_{front}) \tag{5}$$

4. Experiments

4.1. Datasets

To deeply understand the strength and weaknesses of different methods we run experiments on a large number (16) of datasets which present a wide variety of conditions, with various degrees of intra-dataset variability.

Given the large number of datasets, we split them into two categories:

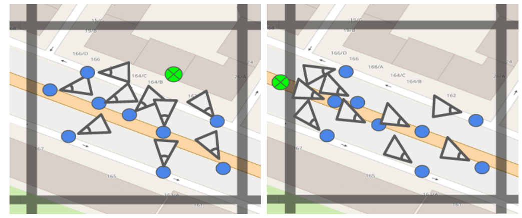

Figure 5. **Lateral vs Frontal Loss.** The image on the left shows how the views are built for the computation of the lateral loss (*i.e.* lateral with respect to the car going along the road), which makes the model robust to multi-view datasets like Pitts30k. On the right is shown the construction for the frontal loss, which improves results on frontal-view datasets like MSLS. The lateral loss places the *focal point* on the second principal component, whereas the frontal loss places it on the first principal component.

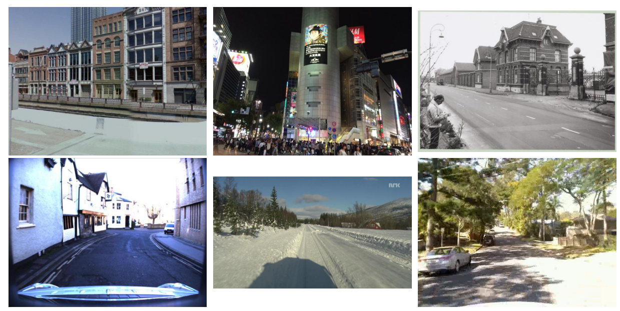

Figure 6. **Examples of images from multiple datasets.** In the top row there are queries from multi-view datasets, namely Pitts30k, Tokyo 24/7 and AmsterTime; on the bottom row queries from frontal-view datasets - SVOX sun, Nordland and St Lucia. Further examples from all datasets are reported in the Supplementary.

1. **multi-view datasets**, which contain images in any direction w.r.t. to the direction of the road;
2. **frontal-view datasets**, containing for the vast majority images along the road.

The visual differences between the two categories can be seen in Fig. 6, while the list of datasets is provided in Tab. 2 and Tab. 1.

Among the ones with largest **viewpoint variance**, we note Tokyo 24/7 [50], San Francisco Landmark [13] and SF-XL test v1 and v2 [8], all of which contain queries collected with a phone, usually from sidewalks, while the database is from streetview images. Given the nature of the task, we use mostly urban datasets, with the main exception being Nordland [47], which is a collection of photos taken across different seasons with a camera mounted on a train. Some datasets present various degrees of **day-to-night** changes, namely MSLS [57], Tokyo 24/7 [50], SF-XL test v1 [8] and SVOX Night [11]. Additionally, SVOX contains a comprehensive set of **weather domain shifts**, with overcast, rainy, snowy and sunny images. Eynsham [17] is the only completely grayscale dataset, whereas AmsterTime [58] con-

Dataset Name	AmsterTime	Eynsham	Pitts30k	Pitts250k	Tokyo 24/7	San Francisco Landmark	SF-XL test v1	SF-XL test v2
# queries	1231	24k	6.8k	8.3k	315	598	1000	598
# database	1231	24k	10k	84k	76k	1.04M	2.8M	2.8M
Orientation	multi-view	multi-view	multi-view	multi-view	multi-view	multi-view	multi-view	multi-view
Scenery	urban	urban & country	urban	urban	urban	urban	mostly urban	mostly urban
Domain Shift	long-term	none	none	none	day/night	viewpoint	viewpoint, night	viewpoint

Table 1. **Overview of multi-view datasets.** We can see huge variations in size and types of domain shift across the datasets.

Dataset Name	MSLS Val	Nordland	St Lucia	SVOX Night	SVOX Overcast	SVOX Rain	SVOX Snow	SVOX Sun
# queries	740	27592	1464	823	872	937	870	854
# database	18.9k	27592	1549	17k	17k	17k	17k	17k
Orientation	frontal-view	frontal-view	frontal-view	frontal-view	frontal-view	frontal-view	frontal-view	frontal-view
Scenery	mostly urban	country	suburb	urban	urban	urban	urban	urban
Domain Shift	day/night	summer/winter	none	day/night	weather	weather	weather	weather

Table 2. **Overview of frontal-view datasets.** We can see huge variations in size and types of domain shift across the datasets.

tains **grayscale historical** queries and modern-time RGB database images, making it the only dataset with time variations up to multiple decades.

The datasets are also representative of different sizes of covered area, with the biggest ones being San Francisco Landmark [13] (with a database covering 13.6 km^2) and SF-XL, which covers 170 km^2. An overview over each dataset individually is available in the Supplementary.

4.2. Implementation details

Architecture
In order to assess the potential of the EigenPlaces training method in improving the robustness of neural networks, we opt for a very simple architecture made of a standard convolutional neural network (VGG-16 [45] or ResNet-50 [24], following previous work [5, 27, 31, 21, 8, 1, 2]) to produce embeddings which are fed to a GeM [42] pooling. Finally a fully connected layer produces the descriptors. Hence the dimensionality of each descriptors is equivalent to the number of neurons within the fully connected layer, making it straightforward to change.

This is a much simpler architecture than most previous works, most of which [5, 31, 27, 21, 40] rely on a more complex NetVLAD layer, whereas the most recent work, namely MixVPR [2], employs a MLP-Mixer to aggregate the features provided by the backbone.

Training
EigenPlaces is trained for 200k iterations with batches of 128 images (64 for each component of the loss). We use a learning rate of $1e^{-5}$, and as optimizer we use Adam [28]. We use the same data augmentations as SFRS [21] (color jittering and random cropping). Regarding the partitioning in classes, we set M (the side of the squared cells) to 15 meters and $N = 3$. We set the *focal distance* $D = 10$ meters following preliminary experiments on the validation set, although in Sec. 4.5 we see that using $D = 20$ achieves even higher results on average on a number of datasets.

Training is performed on the San Francisco eXtra Large dataset (SF-XL) [8]: to select images pointing towards the *focal point* we first select the whole 360° panorama, from which we obtain a crop with the required orientation.

Evaluation
Following previous literature [5, 27, 1, 2, 8, 35, 25, 60, 55, 61], we use the recall@N as metric, defined as the percentage of queries for which at least one of the first N predictions is within a given threshold distance. The threshold is usually set to 25 meters, except for Nordland and AmsterTime. For Nordland, being the dataset a collection of aligned frames across 4 seasons, a query is considered correctly localized if at least one of its first N predictions is within 10 frames from the ground truth equivalent in the database (as in [23, 22]). On the other hand AmsterTime [58] is a collection of pairs of images, making a query correctly localized if one of its first N predictions is the query's counterpart in the database.

4.3. Comparison with previous work

Baselines. We run an extensive set of experiments to thoroughly evaluate the soundness of EigenPlaces, comparing it with a large number of open source methods from the literature. Specifically, we use older NetVLAD-based methods which rely on a VGG-16 backbone, namely NetVLAD itself [5] and SFRS [21]. We also compute results with the more recent CosPlace [8], and the latest works of Conv-AP [1] and MixVPR [2], which were trained on the Google StreetView (GSV) dataset [1]. CosPlace, Conv-AP and MixVPR provide open-source models with multiple backbone and descriptors dimensionalities, allowing us to provide a large number of comparisons with different architectures.

Following previous VPR works that use image retrieval [5, 27, 31, 62, 21, 8, 1, 40, 39, 59], we do not compare

Method	Backbone	Desc. Dim.	AmsterTime	Eynsham	Pitts30k	Pitts250k	Tokyo 24/7	San Francisco Landmark	SF-XL test v1	SF-XL test v2
CosPlace [8]	VGG-16	512	38.7	88.3	88.4	89.7	81.9	80.8	65.9	83.1
EigenPlaces (Ours)	VGG-16	512	38.0	89.4	89.7	91.2	82.2	83.8	69.4	86.3
NetVLAD [5]	VGG-16	4096	16.3	77.7	85.0	85.9	69.8	79.1	40.0	76.9
SFRS [21]	VGG-16	4096	29.7	72.3	89.1	90.4	80.3	83.1	50.3	83.8
CosPlace [8]	ResNet-50	128	39.9	88.6	89.0	89.6	81.0	82.9	69.1	86.5
MixVPR [2]	ResNet-50	128	23.1	84.8	87.7	88.7	56.8	66.9	36.7	68.4
EigenPlaces (Ours)	ResNet-50	128	37.9	89.1	89.6	90.2	79.4	85.5	72.4	86.6
CosPlace [8]	ResNet-50	512	46.4	89.9	90.2	91.7	89.5	85.6	76.7	89.0
Conv-AP [1]	ResNet-50	512	28.4	86.2	89.1	90.4	61.3	68.4	41.8	64.0
MixVPR [2]	ResNet-50	512	35.8	87.6	90.4	93.0	78.4	79.4	57.7	84.3
EigenPlaces (Ours)	ResNet-50	512	45.7	90.5	91.9	93.5	89.8	89.5	82.6	90.6
CosPlace [8]	ResNet-50	2048	47.7	90.0	90.9	92.3	87.3	87.1	76.4	88.8
Conv-AP [1]	ResNet-50	2048	31.3	86.6	90.4	92.3	71.1	71.7	47.8	68.1
EigenPlaces (Ours)	ResNet-50	2048	**48.9**	**90.7**	**92.5**	**94.1**	**93.0**	**89.6**	**84.1**	**90.8**
Conv-AP [1]	ResNet-50	4096	33.9	87.5	90.5	92.3	76.2	73.7	47.5	74.4
MixVPR [2]	ResNet-50	4096	40.2	89.4	91.5	**94.1**	85.1	83.8	71.1	88.5
Conv-AP [1]	ResNet-50	8192	35.0	87.6	90.5	92.6	72.1	74.4	49.3	75.8

Table 3. **Recall@1 on multi-view datasets**, split according to the utilized backbone and descriptors dimension. Best overall results on each dataset are in **bold**, best results for each group are underlined.

Method	Backbone	Desc. Dim.	MSLS Val	Nordland	St Lucia	SVOX Night	SVOX Overcast	SVOX Rain	SVOX Snow	SVOX Sun
CosPlace [8]	VGG-16	512	82.6	58.5	95.3	44.8	88.5	85.2	89.0	67.3
EigenPlaces (Ours)	VGG-16	512	84.2	54.5	95.4	42.3	89.4	83.5	89.2	69.7
NetVLAD [5]	VGG-16	4096	58.9	13.1	64.6	8.0	66.4	51.5	54.4	35.4
SFRS [21]	VGG-16	4096	70.0	16.0	75.9	28.6	81.1	69.7	76.0	54.8
CosPlace [8]	ResNet-50	128	85.5	54.7	98.7	35.4	88.5	80.4	86.6	65.2
MixVPR [2]	ResNet-50	128	79.1	47.8	99.0	25.9	92.3	80.9	87.7	73.5
EigenPlaces (Ours)	ResNet-50	128	83.4	50.5	98.8	29.0	90.9	83.8	91.1	68.5
CosPlace [8]	ResNet-50	512	86.9	66.5	99.1	51.6	90.0	87.3	89.5	75.9
Conv-AP [1]	ResNet-50	512	82.3	59.2	99.2	36.0	90.5	80.3	86.4	75.3
MixVPR [2]	ResNet-50	512	83.6	67.2	99.2	44.8	93.9	86.4	93.9	78.7
EigenPlaces (Ours)	ResNet-50	512	**89.5**	67.9	99.5	51.5	92.8	89.0	92.0	83.1
CosPlace [8]	ResNet-50	2048	87.4	71.9	99.6	50.7	92.2	87.0	92.0	78.5
Conv-AP [1]	ResNet-50	2048	81.2	62.3	99.3	37.9	92.0	83.7	90.2	80.3
EigenPlaces (Ours)	ResNet-50	2048	89.1	71.2	99.6	58.9	93.1	90.0	93.1	**86.4**
Conv-AP [1]	ResNet-50	4096	82.8	59.6	99.6	41.9	91.2	81.9	87.9	82.0
MixVPR [2]	ResNet-50	4096	87.2	**76.2**	99.6	**64.4**	**96.2**	**91.5**	**96.8**	84.8
Conv-AP [1]	ResNet-50	8192	82.4	62.9	**99.7**	43.4	91.9	82.8	91.0	80.4

Table 4. **Recall@1 on frontal-view datasets**, split according to the utilized backbone and descriptors dimension. Best overall results on each dataset are in **bold**, best results for each group are underlined.

pure retrieval methods like the ones in Tab. 3 with 2-stage re-ranking techniques, such as [22, 55, 12, 48].

Discussion of results. Given the large number of datasets, we split the results in two parts:

1. in Tab. 3 we show results on multi-view datasets, where the database and queries orientation can vary across 360° ;
2. in Tab. 4 we report experiments on frontal-view datasets, *i.e.* where the vast majority (or all) of the images are forward facing.

The findings from our large set of experiments can be summarized as follows:

- Firstly we can see that older methods like NetVLAD (2016) and SFRS (2020), despite producing larger descriptors are less robust to domain shifts, and are easily outperformed by newer models especially on frontal-view datasets.
- Latest models, namely CosPlace (2022), Conv-AP (2022) and MixVPR (2023), all provide robust results even when using compact descriptors.
- There is no single model that outperforms all other ones on all datasets, and different models have different characteristics and strenghts.
- Despite not achieving state of the art on all datasets, EigenPlaces has the best overall results, which is especially noticeable on multi-view datasets from Tab. 3 which provide larger viewpoint changes.
- MixVPR on average outperforms EigenPlaces on frontal-view datasets, although at the cost of twice big-

ger descriptors.
- EigenPlaces and CosPlace provide strong results also on datasets with grayscale images, namely AmsterTime and Eynsham, despite not being trained with grayscale augmentation.
- Among the most interesting findings, we can see that with low-dimensionality descriptors (*i.e*. 128-D) CosPlace, MixVPR and EigenPlaces provide remarkably good results on datasets with little to no domain shift between database and queries (*e.g*. Pitts30k, St Lucia), although lower dimensionality descriptors still struggle on cross-domain datasets (*e.g*. AmsterTime, Tokyo 24/7, SVOX night).
- With the impressive results reached in the last two years, many datasets can be considered almost solved, with results reaching over 90% in Recall@1, with Recall@10 higher than 95% in most cases (see the Supplementary).

A more extensive set of experiments is reported in the Supplementary.

4.4. Analysis of resources

GPU memory footprint. EigenPlaces is surprisingly cheap to train, as it can train its best architecture using less than 7 GB of memory (ResNet-50 with 2048-D descriptors). This makes it quite lighter than MixVPR [2], which requires more than 18 GB of memory, using their batch size of 480 (*i.e*. 120 quadruplets). Following previous work [1, 2] we use mixed precision to reduce GPU footprint and speed up computation.

Training and evaluation time. EigenPlaces with our best architecture takes 24 hours to train on a single 3090 GPU, which is similar to the duration of training previous methods (SFRS, CosPlace, MixVPR), and we found that different descriptors dimensionality have a negligible impact on training time.

On the other hand, descriptors dimensionality is linearly correlated to two very important factors in large scale image retrieval: **memory footprint** and **matching time** (*i.e*. the time required by the nearest neighbor search). Therefore, wrt MixVPR's best configuration, our top performing model is twice as fast and requires half the memory, while achieving overall better results. Note that in real-world systems the descriptors of the database images are extracted offline, and the inference time can be computed as the sum of the query's descriptors extraction time plus the matching (kNN) time, rendering the extraction time negligible when working on large scale datasets.

4.5. Ablations

Ablation on the loss. In this section we investigate how each of the two components of the loss affects the results.

Lateral Loss	Frontal Loss	Pitts30k	Tokyo 24/7	MSLS Val	St Lucia	Average
✓		<u>90.2</u>	<u>80.0</u>	83.1	97.3	87.6
	✓	89.5	78.1	<u>85.8</u>	**99.3**	88.2
✓	✓	**90.5**	**82.2**	**86.2**	<u>99.0</u>	**89.5**

Table 5. **Ablation on the two components of the loss.** Experiments show the Recall@1 obtained with a ResNet-18 with output dimensionality 512. We can see that training with the frontal loss only achieves good results on images that are mostly made of frontal-view images (MSLS and St Lucia) but poor on others, and the model with both components of the loss achieves best overall performances.

Focal Distance (meters)	Pitts30k	Tokyo 24/7	MSLS Val	St Lucia	Average
0	89.4	74.0	82.6	98.4	86.1
10	**90.5**	82.2	**86.2**	99.0	89.5
20	90.3	**84.4**	86.1	**99.5**	**90.1**
30	90.3	82.9	85.0	**99.5**	89.4
50	90.4	83.8	85.9	**99.5**	89.9

Table 6. **Ablation on focal distance**, shown as the Recall@1 obtained with a ResNet-18 with output dimensionality 512 on multiple datasets.

An ablation is reported in Tab. 5. Experimental evidence shows that using only the lateral loss, which places the *focal point* on the second principal component (see Fig. 5), is enough to reach satisfactory results on multi-view datasets like Pitts30k and Tokyo 24/7, although it fails to produce robust embeddings for frontal-view datasets. On the other hand, relying solely on the frontal-view loss, which places the *focal point* on the first principal component, allows to attain very strong results on MSLS and St Lucia. On Pitts30k and Tokyo 24/7, this configuration suffers from a considerable drop. Finally, their combination provides a robust combination of each component's strength, and reaches good results on all datasets.

Ablation on the focal distance. In this section we compute experiments changing the *focal distance*, *i.e*. the distance between the mean of the images' position and the *focal point*. Results are in Tab. 6. Although the best overall results are achieved with a *focal distance* of 20 and 10 meters, using higher distances leads to good results on frontal-view datasets. This is not surprising, given that higher focal distances lead the training images' orientation to be further from the center of the cells (*i.e*. straight along the road), as is usually the case for these kind of datasets. On the other hand, we can see that a *focal distance* of 0 meters achieves better results than expected, considering that in this situation some of the images will be facing opposite directions.

Embeddings invariance. In Fig. 7 we test whether our proposed training algorithm is indeed effective in embedding viewpoint robustness in the model. In a randomly selected cell, we sort the images along the first principal component,

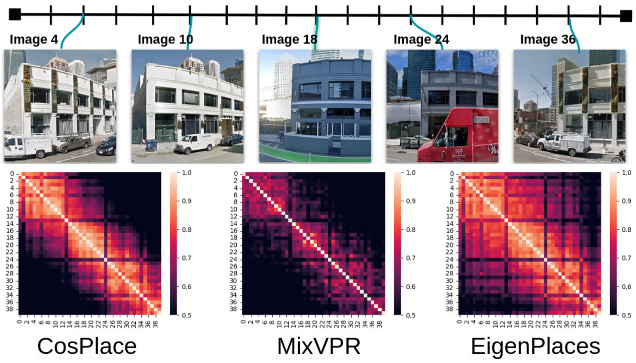

Figure 7. **Confusion matrices with the cosine similarity among 40 images representing the same place, from different viewpoints.** The cosine similarity is computed in features space, with the three most relevant methods of CosPlace, MixVPR and EigenPlaces. For example, the value withing the matrix at position (2, 34) is the cosine similarity between the second and 34th image within a given cell. We can see that EigenPlaces is able to have high correlation even from images that have very different viewpoint (*i.e*. with indexes distant from each other), whereas previous works only share similar features among images that have very close point of view. Some descriptors are quite different from the others as those images might have occlusion (*e.g*. the red truck in image 24).

extract the features of the images oriented towards the *focal point* and compute a similarity matrix. The obtained matrix shows along its rows and columns what happens to the embeddings when traversing the principal component. For previous works, this analysis shows clearly a rapid decrease in embedding similarity when changing the viewpoint, whereas EigenPlaces ensures more robustness.

5. Conclusions

In this work we introduced a novel training algorithm for VPR, that tackles the challenge of perspective shifts. After dividing the available map into fine-grained cells, our method builds classes by inferring from the data inside each cell a point of interest that is depicted from as many different viewpoints as possible. By minimizing a loss that asks the network to recognize the same point from various perspectives, we embed viewpoint-invariance into a feature extractor. We support our contribution through extensive experiments on a vast amount of datasets with diverse characteristics and challenges. We discuss how each dataset can highlight different capabilities in a model, and despite the wide variety of test cases we show that using EigenPlaces we obtain SOTA result in the majority of cases, while using lighter descriptors than previous works.

References

[1] Amar Ali-bey, Brahim Chaib-draa, and Philippe Giguère. Gsv-cities: Toward appropriate supervised visual place recognition. *Neurocomputing*, 513:194–203, 2022. 1, 3, 4, 6, 7, 8

[2] Amar Ali-bey, Brahim Chaib-draa, and Philippe Giguère. Mixvpr: Feature mixing for visual place recognition. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2998–3007, 2023. 1, 2, 3, 6, 7, 8

[3] Asha Anoosheh, Torsten Sattler, Radu Timofte, Marc Pollefeys, and Luc Van Gool. Night-to-day image translation for retrieval-based localization. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 5958–5964. IEEE, 2019. 1

[4] R. Arandjelović and Andrew Zisserman. Three things everyone should know to improve object retrieval. pages 2911–2918, 2012. 2

[5] Relja Arandjelović, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. NetVLAD: CNN architecture for weakly supervised place recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6):1437–1451, 2018. 1, 2, 6, 7

[6] Artem Babenko, Anton Slesarev, A. Chigorin, and V. Lempitsky. Neural codes for image retrieval. *ArXiv*, abs/1404.1777, 2014. 2

[7] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Computer Vision and Image Understanding*, 110:346–359, 06 2008. 2

[8] Gabriele Berton, Carlo Masone, and Barbara Caputo. Rethinking visual geo-localization for large-scale applications. In *CVPR*, June 2022. 1, 2, 3, 5, 6, 7

[9] Gabriele Berton, Carlo Masone, Valerio Paolicelli, and Barbara Caputo. Viewpoint invariant dense matching for visual geolocalization. In *IEEE International Conference on Computer Vision*, pages 12169–12178, October 2021. 2, 3

[10] 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 Conference on Computer Vision and Pattern Recognition*, 2022. 1

[11] Gabriele Berton, Valerio Paolicelli, Carlo Masone, and Barbara Caputo. Adaptive-attentive geolocalization from few queries: A hybrid approach. In *IEEE Winter Conference on Applications of Computer Vision*, pages 2918–2927, January 2021. 1, 3, 4, 5

[12] B. Cao, A. Araujo, and J. Sim. Unifying deep local and global features for image search. In *European Conference on Computer Vision*, pages 726–743. Springer Int. Publishing, 2020. 1, 2, 3, 7

[13] D. M. Chen, G. Baatz, K. Köser, S. S. Tsai, R. Vedantham, T. Pylvänäinen, K. Roimela, X. Chen, J. Bach, M. Pollefeys, B. Girod, and R. Grzeszczuk. City-scale landmark identification on mobile devices. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 737–744, 2011. 3, 4, 5, 6

[14] Z. Chen, A. Jacobson, N. Sünderhauf, B. Upcroft, L. Liu, C. Shen, I. Reid, and M. Milford. Deep learning features at scale for visual place recognition. In *2017 IEEE International Conference on Robotics and Automation*, pages 3223–3230, 2017. 1

[15] Z. Chen, L. Liu, I. Sa, Z. Ge, and M. Chli. Learning context flexible attention model for long-term visual place recognition. *IEEE Robotics and Automation Letters*, 3(4):4015–4022, 2018. 1

[16] Z. Chen, F. Maffra, I. Sa, and M. Chli. Only look once, mining distinctive landmarks from ConvNet for visual place recognition. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 9–16, 2017. 1

[17] M. Cummins and P. Newman. Highly scalable appearance-only slam - FAB-MAP 2.0. In *Robotics: Science and Systems*, 2009. 5

[18] Jiankang Deng, J. Guo, and S. Zafeiriou. ArcFace: Additive angular margin loss for deep face recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4685–4694, 2019. 3

[19] A.-D. Doan, Y. Latif, T.-J. Chin, Y. Liu, T.-T. Do, and I. Reid. Scalable place recognition under appearance change for autonomous driving. In *IEEE International Conference on Computer Vision*, pages 9319–9328, October 2019. 1

[20] S. Garg, N. Suenderhauf, and M. Milford. Semantic–geometric visual place recognition: a new perspective for reconciling opposing views. *The International Journal of Robotics Research*, 2019. 1

[21] 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, Cham, 2020. Springer International Publishing. 1, 3, 6, 7

[22] Stephen Hausler, Sourav Garg, Ming Xu, Michael Milford, and Tobias Fischer. Patch-netvlad: Multi-scale fusion of locally-global descriptors for place recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 14141–14152, 2021. 1, 2, 3, 6, 7

[23] S. Hausler, A. Jacobson, and M. Milford. Multi-process fusion: Visual place recognition using multiple image processing methods. *IEEE Robotics and Automation Letters*, 4(2):1924–1931, 2019. 1, 6

[24] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 6

[25] Sarah Ibrahimi, Nanne van Noord, Tim Alpherts, and Marcel Worring. Inside out visual place recognition. In *British Machine Vision Conference*, 2021. 1, 6

[26] A. Khaliq, S. Ehsan, Z. Chen, M. Milford, and K. McDonald-Maier. A holistic visual place recognition approach using lightweight CNNs for significant viewpoint and appearance changes. *IEEE Transactions on Robotics*, 36(2):561–569, 2020. 1

[27] 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*, pages 3251–3260, 2017. 1, 3, 6

[28] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 12 2014. 6

[29] María Leyva-Vallina, Nicola Strisciuglio, and Nicolai Petkov. Data-efficient large scale place recognition with graded similarity supervision. *CVPR*, 2023. 1

[30] Dongfang Liu, Yiming Cui, Liqi Yan, Christos Mousas, Baijian Yang, and Yingjie Chen. Densernet: Weakly supervised visual localization using multi-scale feature aggregation. *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 6101–6109, May 2021. 1, 2

[31] Liu Liu, Hongdong Li, and Yuchao Dai. Stochastic Attraction-Repulsion Embedding for Large Scale Image Localization. In *IEEE International Conference on Computer Vision*, 2019. 1, 3, 6

[32] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 212–220, 2017. 3

[33] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, 2004. 2

[34] W. Maddern, G. Pascoe, C. Linegar, and P. Newman. 1 Year, 1000km: The Oxford RobotCar Dataset. *The International Journal of Robotics Research*, 2017. 3, 4

[35] Carlo Masone and Barbara Caputo. A survey on deep visual place recognition. *IEEE Access*, 9:19516–19547, 2021. 6

[36] Riccardo Mereu, Gabriele Trivigno, Gabriele Berton, Carlo Masone, and Barbara Caputo. Learning sequential descriptors for sequence-based visual place recognition. *IEEE Robotics and Automation Letters*, 7(4):10383–10390, 2022. 1

[37] Michael Milford and G. Wyeth. Mapping a suburb with a single camera using a biologically inspired slam system. *IEEE Transactions on Robotics*, 24:1038–1053, 2008. 3, 4

[38] A. Oliva and A. Torralba. Building the gist of a scene: the role of global image features in recognition. *Progress in brain research*, 155:23–36, 2006. 2

[39] Guohao Peng, Yufeng Yue, Jun Zhang, Zhenyu Wu, Xiaoyu Tang, and Danwei Wang. Semantic reinforced attention learning for visual place recognition. In *IEEE International Conference on Robotics and Automation*, pages 13415–13422. IEEE, 2021. 6

[40] Guohao Peng, Jun Zhang, Heshan Li, and Danwei Wang. Attentional pyramid pooling of salient visual residuals for place recognition. In *IEEE International Conference on Computer Vision*, pages 885–894, October 2021. 6

[41] Horia Porav, Will Maddern, and Paul Newman. Adversarial training for adverse conditions: Robust metric localisation using appearance transfer. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 1011–1018. IEEE, 2018. 1

[42] F. Radenović, G. Tolias, and O. Chum. Fine-tuning CNN Image Retrieval with No Human Annotation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018. 2, 6

[43] A. Razavian, J. Sullivan, A. Maki, and S. Carlsson. Visual Instance Retrieval with Deep Convolutional Networks. *CoRR*, abs/1412.6574, 2015. 2

[44] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: Learning feature matching with graph neural networks. In *CVPR*, 2020. 3

[45] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015. 6

[46] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. LoFTR: Detector-free local feature matching with transformers. *CVPR*, 2021. 3

[47] N. Sünderhauf, P. Neubert, and P. Protzel. Are we there yet? challenging SeqSLAM on a 3000 km journey across all four seasons. In *Proc. of Workshop on Long-Term Autonomy, IEEE International Conference on Robotics and Automation*, page 2013, 2013. 5

[48] Fuwen Tan, Jiangbo Yuan, and Vicente Ordonez. Instance-level image retrieval using reranking transformers. In *IEEE International Conference on Computer Vision*, 2021. 3, 7

[49] Giorgos Tolias, R. Sicre, and H. Jégou. Particular object retrieval with integral max-pooling of CNN activations. *CoRR*, abs/1511.05879, 2016. 2

[50] A. Torii, R. Arandjelović, J. Sivic, M. Okutomi, and T. Pajdla. 24/7 place recognition by view synthesis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(2):257–271, 2018. 1, 3, 4, 5

[51] A. Torii, J. Sivic, M. Okutomi, and T. Pajdla. Visual place recognition with repetitive structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(11):2346–2359, 2015. 1, 3, 4

[52] Akihiko Torii, Josef Sivic, Masatoshi Okutomi, and Tomas Pajdla. Visual place recognition with repetitive structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(11):2346–2359, 2015. 2

[53] A. Torii, Hajime Taira, Josef Sivic, M. Pollefeys, M. Okutomi, T. Pajdla, and Torsten Sattler. Are large-scale 3d models really necessary for accurate visual localization? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43:814–829, 2021. 1

[54] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. Cosface: Large margin cosine loss for deep face recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5265–5274. Computer Vision Foundation / IEEE Computer Society, 2018. 3, 5

[55] Ruotong Wang, Yanqing Shen, Weiliang Zuo, Sanping Zhou, and Nanning Zheng. Transvpr: Transformer-based place recognition with multi-level attention aggregation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13648–13657, June 2022. 3, 6, 7

[56] Xun Wang, Xintong Han, Weilin Huang, Dengke Dong, and Matthew R Scott. Multi-similarity loss with general pair weighting for deep metric learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5022–5030, 2019. 3

[57] Frederik Warburg, Soren Hauberg, Manuel Lopez-Antequera, Pau Gargallo, Yubin Kuang, and Javier Civera. Mapillary street-level sequences: A dataset for lifelong place recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 2020. 1, 3, 4, 5

[58] B. Yildiz, S. Khademi, R. Siebes, and J. Van Gemert. Amstertime: A visual place recognition benchmark dataset for severe domain shift. In *2022 26th International Conference on Pattern Recognition (ICPR)*, pages 2749–2755, Los Alamitos, CA, USA, aug 2022. IEEE Computer Society. 5, 6

[59] 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, 2020. 6

[60] 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. 1, 6

[61] Jian Zhang, Yunyin Cao, and Qun Wu. Vector of locally and adaptively aggregated descriptors for image feature representation. *Pattern Recognition*, 116:107952, 2021. 1, 6

[62] Yingying Zhu, Jiong Wang, Lingxi Xie, and Liang Zheng. Attention-based pyramid aggregation network for visual place recognition. In *2018 ACM Multimedia Conference on Multimedia Conference, MM 2018, Seoul, Republic of Korea, October 22-26, 2018*, pages 99–107. ACM, 2018. 1, 6