Viewpoint Invariant Dense Matching for Visual Geolocalization

Gabriele Berton[1,2], Carlo Masone[2], Valerio Paolicelli[2] and Barbara Caputo[1,2]
[1]Politecnico di Torino [2]Italian Institute of Technology
[gabriele.berton, barbara.caputo]@polito.it [carlo.masone, valerio.paolicelli]@iit.it

Abstract

In this paper we propose a novel method for image matching based on dense local features and tailored for visual geolocalization. Dense local features matching is robust against changes in illumination and occlusions, but not against viewpoint shifts which are a fundamental aspect of geolocalization. Our method, called GeoWarp, directly embeds invariance to viewpoint shifts in the process of extracting dense features. This is achieved via a trainable module which learns from the data an invariance that is meaningful for the task of recognizing places. We also devise a new self-supervised loss and two new weakly supervised losses to train this module using only unlabeled data and weak labels. GeoWarp is implemented efficiently as a re-ranking method that can be easily embedded into pre-existing visual geolocalization pipelines. Experimental validation on standard geolocalization benchmarks demonstrates that GeoWarp boosts the accuracy of state-of-the-art retrieval architectures. The code and trained models will be released upon acceptance of this paper.

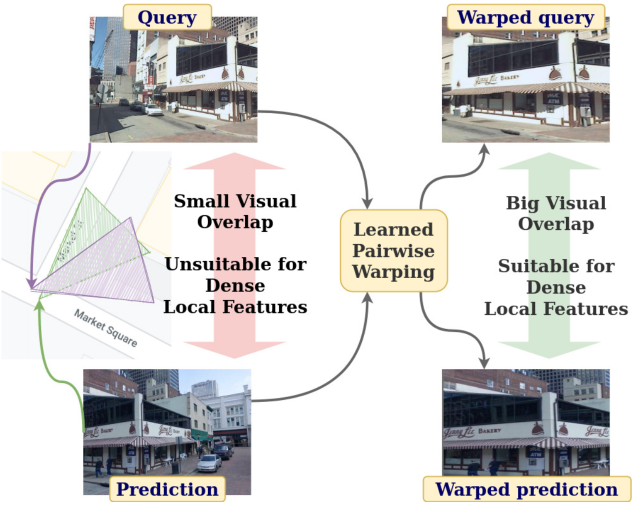

Figure 1. The appearance of two different views of the same place may differ significantly, thus making it hard to match them. Our method warps both images to a closer geometrical space and then computes their similarity using deep dense local features.

1. Introduction

Visual geolocalization (VG), *i.e.*, the task of finding the position where a given photograph was taken, is a fundamental problem in numerous applications, such as robotics localization in GPS-denied environments or augmented reality. This task is cast as an image retrieval problem where the photograph to be localized (*query*) is matched against a labeled database in the space of some global image representations. Much of the recent literature in VG has focused on improving these global representations, moving from aggregations of handcrafted local features [5, 20, 36] to more powerful and compact CNN-based global descriptors [4, 24, 11]. Nevertheless, since global descriptors summarize the whole visual content in the image, they lack robustness to occlusions and clutter [38] and may fail to capture the similarity of two views with a small overlap [23].

Using sparse local invariant features to establish direct geometrical correspondences among images is an effective way to solve this problem in general visual matching tasks. In visual geolocalization this solution is compromised by the fact that different images of the same place may have strong visual differences from one another, *e.g.*, due to illumination or seasonal variations [28]. In such circumstances the detection of keypoints becomes unreliable, causing non-repeatable local invariant features [42, 40]. A few recent studies in visual geolocalization have demonstrated that this problem can be circumvented by removing the detection step altogether and using dense grids of local features to match places across strong visual shifts [42] or with few textures [40]. However, the robustness acquired by removing the detection step comes at the cost of a reduced invariance to geometric transformations.

Since viewpoint shifts are a fundamental problem of visual geolocalization, we propose a new dense matching method, called GeoWarp, that is endowed with some invariance to geometric transformations. Our dense matching is a trainable operation that learns an invariance that is meaningful for the task of recognizing places, in a data driven manner (see Fig. 1). Being an operation dependent on the two images to be matched, it cannot be applied database wide since it would have to be computed again for each query. Therefore, we first perform a neighbor search on the database using state-of-the-art global descriptors, and then apply our novel dense matching on the shortlist of retrieved results to re-rank them. On a technical level, our dense matching revolves around a new lightweight warping regression module that can be efficiently trained in a self-supervised fashion, which makes it possible to train it on unlabeled data. To further improve the results we devise two weakly supervised losses, which allow the network to gain robustness to common issues such as occlusions and appearance shifts, requiring only weak labels from each image.

Contributions:

- We introduce a new dense matching method, tailored for visual geolocalization, that has an intrinsic invariance to viewpoint shifts. This dense matching can be easily integrated into standard retrieval pipelines for geolocalization.
- We present a new trainable method for pairwise image warping. This module is trained with three new losses: a self-supervised loss and two weakly supervised losses, that allow to rely only on unlabeled data or take advantage of weak labels, which are commonly available for visual geolocalization datasets.
- We present an extensive ablation and demonstrate on several standard datasets for visual geolocalization that our method significantly boosts the accuracy with a wide variety of retrieval networks.

2. Related works

Local features Image retrieval has been historically approached with sparse local features using hand-crafted techniques such as SIFT [26] and SURF [6]. Such features are typically aggregated into fixed-length vectors such as bag-of-visual-words [32] or VLAD [2] to ensure efficient similarity computation. To overcome challenges related to variations in viewpoint, sparse local features methods often rely on spatial verification using RANSAC [16]. Alternatively, [42] proposes using dense local features coupled with the generation of synthetic images simulating multiple views of the same scene. Although most CNN-based methods rely on global descriptors [4, 24, 30, 33, 41, 18], recent works [31, 11] show promising results employing deep local features followed by spatial verification.

Global features Since the advent of deep learning, image retrieval systems have been predominantly based on global features descriptors extracted by convolutional architectures. In such systems, images are passed to a convolutional encoder to extract dense local features, which are then fed into aggregation or pooling layers such as NetVLAD [4] or GeM [33]. Such networks are trained with ranking based losses for visual geolocalization [4, 24], whereas classification losses are commonly employed for the related task of landmark retrieval [33, 30, 41, 18].

Re-ranking techniques Re-ranking techniques are commonly used in image retrieval to re-assess the retrieved predictions produced by the retrieval system, trading computational time for a boost in accuracy. A common method is query expansion [13, 18, 1, 33], where the results of a first search are filtered and aggregated to perform a second search. An alternative to query expansion is given by diffusion [15, 45], a family of methods which aim at exploiting the context similarities between all elements of the database to unveil the data manifold, which is not captured by the pairwise similarity search. Closer to our approach, other works [31, 11, 40, 37] perform a first retrieval search using global features, following it with a post-processing step done with local features (spatial verification). Among these, the closest to our work is [40] which, as in our method, uses the same encoder both to generate a global descriptor for the nearest neighbor search and to extract dense local features for re-ranking. However, the dense local features are used to find an accurate camera pose through RANSAC [16], whereas we perform a learned pairwise transformation before local features extraction.

Geometric image transformation Traditionally, correspondences between pairs of images have been computed by finding points of interest and extracting local descriptors from such points [6]. More recent works rely on features extracted from CNNs, which can be used as input for a second deep neural network [9, 14, 35, 22, 10, 34] or to a RANSAC algorithm [40]. In particular, [14] proposes to slightly perturb the patch of an image with a homography, which is then predicted with a regression VGG-like network. [40] and [9] use local features followed respectively by RANSAC [16] and DSAC [8] to predict the 6DOF camera pose within a given 3D environment. [22] uses a Siamese CNN to predict a thin-plate spline transformation between two images of birds, while [35] extends the method to work on image instances from other classes than birds. Both methods rely on a pair of images with foreground objects from the same category, with little to no occlusion, to estimate a transformation from one image to the other. While keeping in considerations valuable lessons learned from previous works, we instead propose a pairwise transformation, aimed at morphing both input images. This ensures clutter or unwanted elements to be removed from both of them, while being robust to pairs of images with little visual overlap. Moreover, to ensure robustness of the network to occlusions and dynamic objects, we propose two novel weakly supervised losses, which leverage photos from the same scene taken over the years.

3. Method

We consider the problem of geolocalizing an unseen RGB image I_q given a gallery of geotagged images $\mathcal{G} = \{(I_i, z_i)\}$, where z_i is the GPS coordinate of the image I_i. We further assume having a training dataset of geotagged images $\mathcal{T}$, divided into training queries and training gallery. We propose to address the visual geolocalization problem by first performing a similarity search over $\mathcal{G}$ based on global descriptors, which produces a set of predictions $\mathcal{P} \subset \mathcal{G}$. Then, we use a novel dense matching method to sort the top predictions in $\mathcal{P}$ based on a similarity measure with the query computed from dense local descriptors.

3.1. Place retrieval with global descriptors

As a first step, our method implements a classic pipeline for place retrieval based on global image descriptors. To generate the global descriptors we utilize a CNN that is composed of two elements:

- A convolutional encoder E, that takes an image and outputs a tensor $f \in \mathbb{R}^{h_f \times w_f \times C}$. The tensor f can be interpreted as a dense $h_f \times w_f$ grid of C-dimensional local feature descriptors and we denote the local feature at the spatial position (i, j) of the grid as $\boldsymbol{f}(i, j)$;
- A layer A that takes the tensor f and produces a vectorial representation of the image either by aggregation (*e.g.*, NetVLAD [4]) or by pooling (*e.g.*, GeM [33]).

Namely, given an image I, its global descriptor is $A(E(I))$. This network is trained specifically for place retrieval using a triplet loss and following the protocol from [4]. At inference time, given a new query I_q we perform a nearest neighbour search over the gallery $\mathcal{G}$ in the space of the global descriptors, which yields the set of predictions $\mathcal{P}$.

3.2. Re-ranking with dense local descriptors

We propose to re-rank the predictions $\mathcal{P}$ by recomputing their similarity to the query I_q using dense local feature descriptors. Although sparse invariant local features are successfully used in various visual matching problem, in visual geolocalization they have shown limited reliability due to the challenging visual conditions that can cause failures in the keypoint detection [47]. On the other hand, directly matching densely sampled local features has shown great promise [42, 40] at the cost of a limited invariance to viewpoint shifts. To overcome this limitation we re-rank the predictions $\mathcal{P}$ using a new trainable matching operation, tailored for visual geolocalization, that learns to extract dense local features that, to some extent, are invariant to viewpoint shifts (see Fig. 2).

To endow the feature extraction process with invariance to viewpoint shifts, we propose a warping regression module W that takes the query I_q and a prediction $I_p \in \mathcal{P}$ and estimates a homographic transformation for each of the two images. The mapping from images to homographic transformations is learned from the data with the goal to better align two different views of a same scene, even when they have limited overlap. The regression module will be detailed in Sec. 3.3. For now, we indicate its mapping as

$$W(I_q, I_p) = [\boldsymbol{t}_q, \boldsymbol{t}_{\boldsymbol{p}}] \tag{1}$$

where $\boldsymbol{t}_q \in \mathbb{R}^8$ and $\boldsymbol{t}_p \in \mathbb{R}^8$ are the estimated parameters for the transformations: they can be seen as the four points needed to extract a homography matrix for the eight-point transformation [19], with the remaining four points being the corners of the image (see Fig. 2). With the estimated parameters $\boldsymbol{t}_q$ and $\boldsymbol{t}_p$, the images I_q and I_p can be transformed using the well known eight-points transformation [19] to generate two warped images $\hat{I}_q$ and $\hat{I}_p$. Hereinafter we will denote this transformation as

$$\begin{aligned} \hat{I}_q &= \text{proj}(I_q, \boldsymbol{t}_q) \\ \hat{I}_p &= \text{proj}(I_p, \boldsymbol{t}_p) \end{aligned} \tag{2}$$

As it will be discussed in Sec. 3.4, this learnable transformation is trained to bring two overlapping views of the same location to a closer perspective. However, when the two images depict a different scene, the limited effect of the projection does not affect the final proximity of both images. Qualitative results of the projection are shown in Fig. 3 and in the supplementary material.

Finally, we extract dense local features from the warped images $\hat{I}_q$ and $\hat{I}_p$, by means of the same encoder E that was trained for producing global image representations (Sec. 3.1), i.e.,

$$\begin{aligned} \hat{f}_q &= E(\hat{I}_q) \\ \hat{f}_p &= E(\hat{I}_p) \end{aligned} \tag{3}$$

Using the same encoder E that was trained specifically for place retrieval helps to produce features that are highly discriminative for locations. The output of the proposed operation is a similarity score between the query and prediction using the distance between the local features of their warped counterparts, i.e.,

$$d_p = \sum_{i=0}^{w_f-1} \sum_{j=0}^{h_f-1} \hat{\boldsymbol{f}}_q(i,j)^T \, \hat{\boldsymbol{f}}_p(i,j) \tag{4}$$

This procedure is repeated for all the predictions in $\mathcal{P}$, which are finally sorted according to the scores yielded by (4). This makes the time complexity of our re-ranking

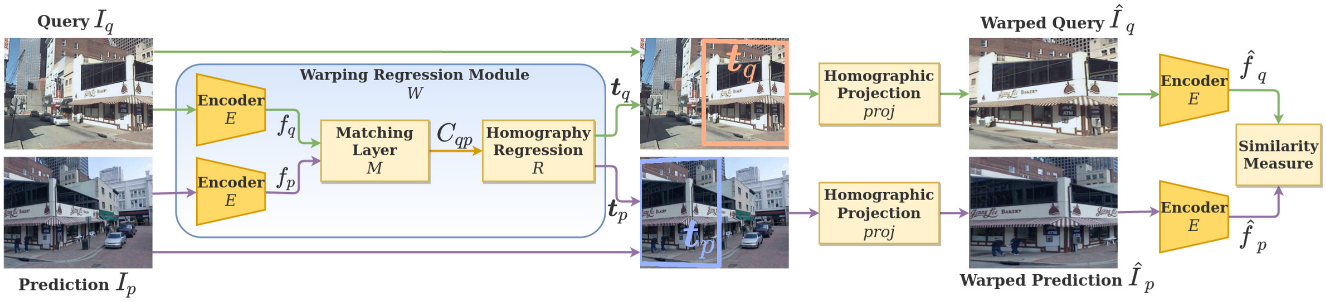

Figure 2. Diagram of our architecture's functioning at inference time. The warping regression module is entitled with estimating two quadrilaterals $\boldsymbol{t}_q$ and $\boldsymbol{t}_p$ from the two images on the left (query and prediction). The images are then warped with a homography, and their similarity is computed on their deep dense local features. Note that an ideally perfect warping which generates very similar images would be counterproductive, as images taken far away (i.e. a query-negative pair) would end up with a similar features representation.

method $O(|\mathcal{P}|)$. Since the number of predictions that needs to be processed is usually small ($|\mathcal{P}| \ll |\mathcal{G}|$), the approach is suitable for large-scale geolocalization problems.

3.3. Warping regression module

The warping regression module W in (1) was inspired by [35], albeit we limit it to homographies. Homographies aim at describing the relationship between representations of coplanar points in a scene as projected onto different viewing planes. In visual geolocalization planar surfaces are abundant, as buildings' facades often represent the most discriminative characteristic of a place, making homography ideal for our purpose. W consists of three steps (Fig. 2). First, we extract the features from the two images using the encoder E, *i.e.*,

$$\begin{aligned} f_q &= E(I_q) \\ f_p &= E(I_p) \end{aligned} \tag{5}$$

Note that E has a threefold purpose: it is used in the global descriptors extraction, the warping module and the final similarity score computation (4).

Then, the features are fed to a matching layer M that computes the correlation map $c_{qp} \in \mathbb{R}^{h_f \times w_f \times (h_f \times w_f)}$ between each pair of local feature descriptors from f_q and f_p, *i.e.*,

$$c_{qp}(i,j,k) = \boldsymbol{f}_q(i,j)^T \boldsymbol{f}_p(i_k, j_k) \tag{6}$$

For the sake of brevity, we write the matching operation as

$$M(f_q, f_p) = c_{qp} \tag{7}$$

We remark that this layer is differentiable and parameterless and we refer to [35] for further details.

Finally, the correlation map c_{qp} is given as input to a convolutional network R that is designed to estimate the transformations of the two images. In particular, the network R estimates 4 points on each of the two images, *i.e.*, a 16D vector. Formally, we denote this operation as

$$R(c_{qp}) = [\underbrace{\boldsymbol{p}_{q1}, \dots, \boldsymbol{p}_{q4}}_{\boldsymbol{t}_q}, \underbrace{\boldsymbol{p}_{p1}, \dots, \boldsymbol{p}_{p4}}_{\boldsymbol{t}_p}] = [\boldsymbol{t}_q, \boldsymbol{t}_p] \in \mathbb{R}^{16} \tag{8}$$

where $\boldsymbol{p}_{q1}, \dots, \boldsymbol{p}_{q4}$ are four points on I_q, $\boldsymbol{p}_{p1}, \dots, \boldsymbol{p}_{p4}$ are four points on I_p, and the notation $[\boldsymbol{t}_q, \boldsymbol{t}_p]$ denotes the concatenation of the two vectors (see Fig. 2).

Combining (5), (7) and (8) the warping regression (1) is summarized as,

$$W(I_q, I_p) = R\big(M(E(I_q), E(I_p))\big) = [\boldsymbol{t}_q, \boldsymbol{t}_p] \tag{9}$$

Although our warping regression module W is inspired by [35], it introduces some notable novelties. The first difference arises from the use case of the warping operation. While [35] regresses a geometric transformation for generic image matching, we focus specifically on the geolocalization problem. As already mentioned, we use the same encoder E that was trained for place retrieval, which means that we do not need to train a second encoder and that the extracted features encode more discriminative information for distinguishing places.

The second and most important difference is that the transformation module in [35] is designed to estimate only the transformation of one image while keeping the other one unchanged. On the contrary, our solution considers the more general problem in which both images can be transformed. In the spirit of deep learning, we let the module itself learn from the data whether and how much each of the two images should be transformed. This ensures greater flexibility and the possibility of achieving greater similarity between the generated pair. This is demonstrated quantitatively in the experiments presented in Sec. 4, and qualitatively in Fig. 3 and in the supplementary material. This second difference also implies that the training procedure from [35] is not applicable to our case. Therefore, we propose a new training protocol, which is discussed next in Sec. 3.4.

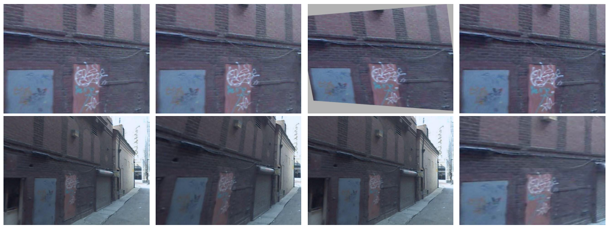

Figure 3. Qualitative results: the first column represents a query-prediction pair, the second column shows warping on the prediction using [35], the third shows warping on the query using [35], and the rightmost column is our pairwise warping.

3.4. Training the warping regression module

To train the warping regression module W in a fully supervised way we would need a dataset with training quadruplets $\{I_a, I_b, \boldsymbol{t}_a, \boldsymbol{t}_b\}$, where I_a and I_b are two images of the same location viewed from different viewpoints and $\boldsymbol{t}_a$ and $\boldsymbol{t}_b$ are the ground truth parameters of the homographic transformations. Given the lack of such a dataset, we propose a training procedure that combines a new self-supervised loss L_{ss} (Sec. 3.4.1), as well as two novel weakly supervised losses L_{fw} and L_{cons} (Sec. 3.4.2). Hence, the total loss is

$$L_{total} = \lambda_{ss} L_{ss} + \lambda_{fw} L_{fw} + \lambda_{cons} L_{cons} \tag{10}$$

Before proceeding with detailing the terms in (10), we point out that: i) the matching layer M is parameterless, and ii) the encoder E was previously trained for place retrieval and its parameters are kept frozen when training W. This second point also ensures that when training W we can rely on features that are optimized for the geolocalization task.

3.4.1 Self-supervised training

We propose to generate training quadruplets $\{I_a, I_b, \boldsymbol{t}_a, \boldsymbol{t}_b\}$ from a single training image and train the regression network in W in a self-supervised fashion. Let us consider a generic training image I with shape $w \times h$. We define a procedure to randomly sample the corners of a quadrilateral on I:

$$\begin{aligned}
\boldsymbol{p}_1 &= \left[\tfrac{U(0,k)*w}{2}, \tfrac{U(0,k)*h}{2}\right] \\
\boldsymbol{p}_2 &= \left[w - \tfrac{U(0,k)*w}{2}, \tfrac{U(0,k)*h}{2}\right] \\
\boldsymbol{p}_3 &= \left[w - \tfrac{U(0,k)*w}{2}, h - \tfrac{U(0,k)*h}{2}\right] \\
\boldsymbol{p}_4 &= \left[\tfrac{U(0,k)*w}{2}, h - \tfrac{U(0,k)*h}{2}\right] \\
\text{s.t.} \quad & \boldsymbol{p}_1[0] = \boldsymbol{p}_4[0] \\
& \boldsymbol{p}_2[0] = \boldsymbol{p}_3[0]
\end{aligned} \tag{11}$$

where $U(a, b)$ is the uniform distribution and $k \in [0, 1]$ is a constant. When k approaches 0 the four points are close to the corners of I, whereas higher values move them towards the center of the image. An example with $k = 0.8$ is shown in Fig. 4, and more examples with other values of k are shown in the supplementary material. The two constraints in (11) impose that the four points delimit a quadrilateral with two vertical parallel sides. This introduces a bias towards vertically aligned images, which represent the standard in visual geolocalization datasets [2, 42, 44, 12, 29, 46, 7, 27], as well as in real world application (e.g. autonomous vehicles imagery and user-generated photos). We apply the procedure (11) twice to generate two trapezoids on I, which we denote as $\boldsymbol{t}_x = [\boldsymbol{p}_{x1}, \boldsymbol{p}_{x2}, \boldsymbol{p}_{x3}, \boldsymbol{p}_{x4}] \in \mathbb{R}^8$ and $\boldsymbol{t}_y = [\boldsymbol{p}_{y1}, \boldsymbol{p}_{y2}, \boldsymbol{p}_{y3}, \boldsymbol{p}_{y4}] \in \mathbb{R}^8$. By construction, the intersection $\boldsymbol{t}_x \cap \boldsymbol{t}_y$, is never empty and we define $\boldsymbol{t}_z = [\boldsymbol{p}_{z1}, \boldsymbol{p}_{z2}, \boldsymbol{p}_{z3}, \boldsymbol{p}_{z4}]$ as the widest trapezoid with two vertical edges within the intersection (see Fig. 4).

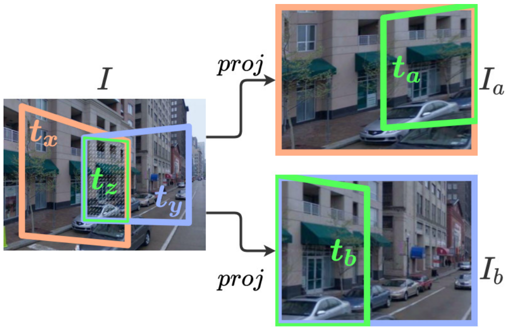

Figure 4. Self-supervised generation of training quadruplets $\{I_a, I_b, \boldsymbol{t}_a, \boldsymbol{t}_b\}$ from a single training image I (left). By construction $\boldsymbol{t}_z$ is known, and so are its projections $\boldsymbol{t}_a$ and $\boldsymbol{t}_b$.

Then, we generate the training quadruplet from I, $\boldsymbol{t}_x$, $\boldsymbol{t}_y$ and $\boldsymbol{t}_z$ by homographic projection:

$$\begin{aligned}
I_a &= \text{proj}(I, \boldsymbol{t}_x) \qquad & \boldsymbol{t}_a &= \text{proj}(\boldsymbol{t}_z, \boldsymbol{t}_x) \\
I_b &= \text{proj}(I, \boldsymbol{t}_y) \qquad & \boldsymbol{t}_b &= \text{proj}(\boldsymbol{t}_z, \boldsymbol{t}_y)
\end{aligned} \tag{12}$$

Note that in (12) we have used the same notation that was introduced in (2) to indicate both the homographic projection of an image and the projection of a set of four points. Albeit not accurate, this prevents further encumbering the notation.

Finally, we define the self-supervised warping loss as

$$L_{SS} = \|W(I_a, I_b) - [\boldsymbol{t}_a, \boldsymbol{t}_b]\|^2 \tag{13}$$

This loss guides the network to learn to estimate the points describing the area where the two input images intersect.

3.4.2 Weakly supervised losses

The synthetic quadruplets produced by the self-supervised method presented in Sec. 3.4.1 can be used effectively to train the network, however they do not provide a realistic representation of the data distribution presented at in-

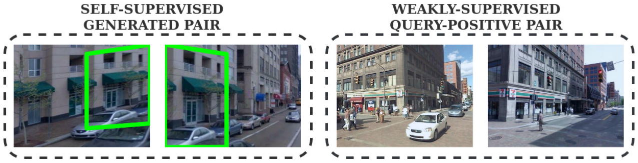

Figure 5. Examples of data used for training. With self-supervised generated images we have the advantage of knowing the intersection's ground truth, while the use of weakly supervised query-positive pairs better simulates a test-time situation.

ference time by query-predictions pairs. The synthetic images, being extracted from a single image, contain the same dynamic objects (such as vehicles and pedestrians) as well the same textures (*i.e.*, same color of the sky, same vegetation). On the contrary, the pairs of queries and predictions seen at inference time are photos taken at different times, sometimes months or years apart. This might result in an accuracy drop during inference. To mitigate this unwanted behaviour, we propose to use pairs of queries and predictions from the training set $\mathcal{T}$, which we mine in a weakly supervised fashion (see Fig. 5). Formally, we form these pairs by taking all pairs of training queries $\{(I_q, z_q)\}$ and training gallery samples $\{(I_g, z_g)\}$ that satisfy these constraints:

$$d_{geo}(z_q, z_g) < t_{geo} \tag{14}$$

$$\|A(E(I_q)), A(E(I_g))\|^2 < t_{feat} \tag{15}$$

where $d_{geo}(a, b)$ is the geographical distance between the location of two images, *i.e.*, the distance in meters computed from their tagged GPS coordinates, t_{geo} is a distance threshold (usually set to 25 meters) and t_{feat} is a threshold in the features space. Constraints (14) and (15) impose that the image from the gallery is taken in the proximity of the query, and that the two are similar in the features space, making it very unlikely for the two images to contain a different scene. This mining is performed once, before the start of the training, and it generates a set of pairs of images $\{(I_q, I_g)\}$, where I_q is a training query and I_g is its related positive. Note that the same query image might be represented more than once in the set, while others might not belong to it. We use this set of training pairs in conjunction with two new weakly supervised losses.

Features-wise loss The first weakly supervised loss is designed to ensure that the features extracted from the training pair after warping are as close as possible, as this is the goal of the matching operation. The features-wise loss is constructed as follows. First, we regress the homographies from the pair of images using the warping regression module W, warp them and extract their features, *i.e.*,

$$\begin{aligned} W(I_q, I_g) &= [\boldsymbol{t}_q, \boldsymbol{t}_g] \\ \hat{f}_q &= E\left(\text{proj}(I_q, \boldsymbol{t}_q)\right) \\ \hat{f}_g &= E\left(\text{proj}(I_g, \boldsymbol{t}_g)\right) \end{aligned} \tag{16}$$

Afterwards, we compute the features-wise loss as

$$L_{fw} = \sum_{i=0}^{w_f - 1} \sum_{j=0}^{h_f - 1} \left(\hat{\boldsymbol{f}}_q(i,j)^T \, \hat{\boldsymbol{f}}_g(i,j) \right)^2 \tag{17}$$

While the self-supervised loss (13) might rely on dynamic objects or lighting conditions to learn an estimate of the warping, this loss guarantees the output to be robust to such scene variations.

The loss L_{fw} has some similarities with the VGG-based perceptual loss [17], as both losses aim at reducing the distance between two images after projecting them into features spaces. However, differently from the perceptual loss, our L_{fw} computes a similarity between the two images using only high level features that are extracted from the last convolutional layer of the encoder. Since the encoder is previously trained on a visual geolocalization task, this ensures that the features are mostly extracted from parts of the image which are informative for this task. In this way the network's generated homography will focus mostly on static objects, whereas using a perceptual loss would force the network to learn a homography based on the entire scene represented in the image.

Consistency loss Given the lack of homography labels in our task, we propose to use self-generated pseudo-labels as ground truths to further improve the robustness. We generate the pseudo-labels as follows:

$$\begin{aligned} \frac{1}{N} \sum_{i=1}^{N} \tau_i^{-1}(W(\tau_i(I_q), \tau_i(I_g))) &= [\boldsymbol{t}'_q, \boldsymbol{t}'_g] \\ \frac{1}{N} \sum_{i=1}^{N} \tau_i^{-1}(W(\tau_i(I_g), \tau_i(I_q))) &= [\boldsymbol{t}''_g, \boldsymbol{t}''_q] \\ [\boldsymbol{t}^*_q, \boldsymbol{t}^*_g] &= \frac{[\boldsymbol{t}'_q, \boldsymbol{t}'_g] + [\boldsymbol{t}''_q, \boldsymbol{t}''_g]}{2} \end{aligned} \tag{18}$$

where τ is used to indicate an invertible geometric transformation applied to the images, τ^{-1} is its inverse applied to the estimated points, N represents the number of transformations to apply, and $[\boldsymbol{t}^*_q, \boldsymbol{t}^*_g]$ are the generated pseudo-labels for the required homography. Note how using a higher number of transformations (N) guarantees for the pseudo-labels to be more accurate (*i.e.* closer to the ground truths), at the expense of computational requirements. After detaching $[\boldsymbol{t}^*_q, \boldsymbol{t}^*_g]$ from the computational graph, we define the consistency loss as follows:

$$L_{cons} = \frac{1}{2N} \sum_{i=1}^{N} \left(\tau_i^{-1}(W(\tau_i(I_q), \tau_i(I_g))) - [\boldsymbol{t}_q^*, \boldsymbol{t}_g^*]\right)^2 + \left(\tau_i^{-1}(W(\tau_i(I_g), \tau_i(I_q))) - [\boldsymbol{t}_g^*, \boldsymbol{t}_q^*]\right)^2 \quad (19)$$

A trivial solution for minimizing the consistency loss L_{cons} is to teach the warping regression module to extract points at the corner of the images, effectively making the whole warping regression module and homographic projection an identity transformation. This is clearly seen in Tab. 3, where it is shown that L_{cons} on its own brings no advantage, while the improvements are noticeable when it is combined with the other losses.

4. Experiments

4.1. Setup

Datasets and metric We evaluate our method on two standard visual geolocalization datasets: Pitts30k [4] and R-Tokyo [44]. The metric used to validate the results is the recall@1 as defined in [4], *i.e.*, the percentage of queries for which the first prediction is not further than a certain threshold. In our experiments we show results for commonly used thresholds [4, 24, 38, 43, 3], namely 10m, 25m, and 50m, corresponding to different accuracy requirements (from finer to coarser).

Implementation details We test our dense matching on top of various retrieval systems, using different encoders (AlexNet [25], VGG16 [39] and ResNet50 [21]) and aggregation layers (GeM [33] and NetVLAD [4]). Prior to the matching layer, the feature maps produced by the encoder are resized to $15 \times 15 \times C$, where C is the number of channels at the last convolutional layer, to ensure that the method works with images of different sizes. The correlation map outputted by the matching layer is then passed to the homography regression, implemented through a sequence of six convolutional layers and a fully connected layer with output dimensionality 16. This final layer is initialized with weights set to 0, and with biases such that the initial estimated points (prior to any training) correspond to the four corners of the input images. We use a batch size of 16 generated image pairs for the L_{ss} and 16 query-positive pairs for the weakly supervised losses. We train the warping regression module for 50000 iterations. Regarding the consistency loss L_{cons}, we use $N = 2$ geometric transformations, namely the identity transformation and an horizontal flip. We set t_{geo} to 25 meters, $t_{feat} = 1.2$, $|\mathcal{P}| = 5$, and the warping coefficient k to 0.6 (see an ablation on k in Fig. 6). Finally, we set $\lambda_{ss} = 1$, $\lambda_{fw} = 10$, and $\lambda_{cons} = 0.1$.

Backbone	Method	Pitts30k			R-Tokyo		
		10m	25m	50m	10m	25m	50m
AlexNet	GeM	50.4	65.3	70.7	19.2	23.2	24.2
AlexNet	GeM + GeoWarp **(ours)**	**61.5**	**74.7**	**78.4**	**27.9**	**34.0**	**34.7**
AlexNet	NV	64.8	78.7	83.1	41.8	45.8	46.8
AlexNet	NV + GeoWarp **(ours)**	**67.6**	**80.8**	**84.4**	**44.1**	**48.8**	**49.5**
VGG16	GeM	55.4	70.6	76.3	33.7	40.4	45.8
VGG16	GeM + GeoWarp **(ours)**	**65.3**	**79.2**	**83.1**	**49.2**	**55.9**	**58.6**
VGG16	NV	67.2	82.5	86.5	50.2	56.9	59.9
VGG16	NV + GeoWarp **(ours)**	**70.2**	**83.3**	**86.7**	**54.5**	**61.6**	**63.6**
ResNet-50	GeM	68.3	81.4	84.4	36.0	41.8	45.5
ResNet-50	GeM + GeoWarp **(ours)**	**70.4**	**82.7**	**85.6**	**44.1**	**49.5**	**51.9**
ResNet-50	NV	70.2	84.3	87.3	65.0	72.4	74.4
ResNet-50	NV + GeoWarp **(ours)**	**72.1**	**84.8**	**87.8**	**69.7**	**74.4**	**75.4**

Table 1. Recall@1 of well-established baseline methods with and without GeoWarp. NV stands for NetVLAD [4].

4.2. Results

The results of the experiments are reported in Tab. 1. We see that GeoWarp achieves a substantial improvement with all the configurations of encoder and aggregation methods. Notably, the impact of GeoWarp is stronger with GeM than with NetVLAD. In general, we see that NetVLAD provides better embeddings than GeM, thus reducing the impact of our method. However, this comes at the cost of using global descriptors that are 64 times heavier than GeM's, making it unsuitable for large-scale problems. Considering that GeoWarp's computation time does not depend on the size of the gallery, the results achieved with a compact descriptor as GeM proves that our solution is viable for large-scale problems.

4.3. Comparison with other methods

We compare the performance of GeoWarp to state-of-the-art re-ranking solutions for visual geolocalization and landmark recognition. First, we consider two popular techniques for image retrieval problems, query expansion (QE) and diffusion. For QE, we use the official implementation from Gordo et al. [18] which combines it with database-side augmentation (DBA). For diffusion, we use the implementation from Yang et al. diffusion [45]. In both cases, extensive grid searches were performed to find the best hyperparameters. We compute results for DenseVlad [42] and SuperGlue [37] using the implementations provided by the respective authors. We compare to the spatial verification approach from DELG [11], using the official implementation trained on the Google Landmark Dataset [31]. Given the nature of the method, it cannot be fine-tuned on visual geolocalization datasets in a weakly supervised manner. We compare to InLoc [40], which uses dense local features to re-rank the predictions via RANSAC. Lastly, since our warping regression module was inspired by Rocco et al. [35], we test the dense matching strategy from GeoWarp, but replacing our module W with the one from [35]. In particular, we use their released VGG16-based model trained on images from Google StreetView in Tokyo.

Backbone	Method	Pitts30k			R-Tokyo		
		10m	25m	50m	10m	25m	50m
VGG16	NV	67.2	82.5	86.5	50.2	56.9	59.9
VGG16	NV + QE + DBA [18]	66.5	82.3	86.4	51.2	58.2	62.3
VGG16	NV + diffusion [45]	65.1	80.9	85.5	52.2	58.6	62.0
VGG16	NV + Rocco et al. [35]	68.4	81.9	85.7	51.9	59.3	61.0
VGG16	InLoc [40]	44.5	72.8	79.5	36.4	49.5	53.5
VGG16	NV + GeoWarp **(ours)**	**70.2**	**83.3**	**86.7**	**54.5**	**61.6**	**63.6**
-	DenseVLAD [42]	63.6	77.3	81.6	35.4	39.4	40.1
-	SuperGlue [37]	72.0	**84.9**	**88.1**	65.3	73.1	74.7
ResNet-50	NV	70.2	84.3	87.3	65.0	72.4	74.4
ResNet-50	NV + QE + DBA [18]	68.6	83.7	87.1	66.7	72.1	74.1
ResNet-50	NV + diffusion [45]	67.6	81.6	85.1	62.3	69.0	72.4
ResNet-50	DELG [11]	65.4	83.0	88.0	60.6	73.0	75.1
ResNet-50	NV + GeoWarp **(ours)**	**72.1**	84.8	87.8	**69.7**	**74.4**	**75.4**

Table 2. Recall@1 of state-of-the-art methods for geolocalization and retrieval. NV stands for NetVLAD [4]. Note that DenseVLAD does not use a CNN backbone, while SuperGlue uses one ad hoc.

The results of all methods are reported in Tab. 2, which shows that on average GeoWarp gives more accurate results than all other methods, on varying distance thresholds. We see that both diffusion and QE+DBA fail to achieve significant improvements. This is likely due to the fact that visual geolocalization datasets usually have just a few positives per query, whereas these methods work best with landmark retrieval datasets, where positives are abundant. The spatial verification with DELG [11] shows slight improvements only when using a high distance threshold (50m) for positive results. InLoc [40] proves detrimental for re-ranking NetVLAD's global features, which is likely due to its reliance of low level local features (extracted from the VGG's conv5 and conv3) which are not robust to outdoor scene variations. Finally, the dense matching using the warping from Rocco et al. [35] does give a small improvement, particularly on the 10m and 25m distance thresholds. However, this improvement is, on average, 2% less than when using our warping, confirming the importance of our pairwise warping architecture and geolocalization-specific losses.

4.4. Ablation study

We perform an extensive ablation study (see Tab. 3) to verify the significance of each term in the total loss (10). We see that without the guidance of the self-supervised loss (13), re-ranking with local features is generally detrimental with respect to the baseline. As expected, using solely the consistency loss gives results on par with not training the homography regression, as the warping simply performs an identity transformation. While the effect of separately applying each of the two weakly supervised losses leads to similar results, the orthogonality of each loss's gains is clear, as the three losses combined achieve best results.

Features type	λ_{ss}	λ_{fw}	λ_{cons}	VGG16 GeM	VGG16 NV	ResNet50 GeM	ResNet50 NV
global	0	0	0	70.5	85.1	82.9	86.4
local	0	0	0	74.3	83.2	78.2	80.7
local	0	0	0.1	74.2	83.1	78.3	80.8
local	0	10	0	74.3	82.9	78.1	80.9
local	0	10	0.1	74.3	83.0	77.8	80.8
local	1	0	0	78.6	87.0	84.2	87.0
local	1	0	0.1	78.8	87.1	84.3	86.8
local	1	10	0	78.8	87.1	**84.6**	86.9
local	1	10	0.1	**79.0**	**87.2**	84.5	**87.2**

Table 3. Ablation results over the three losses on the validation set of Pitts30k [4].

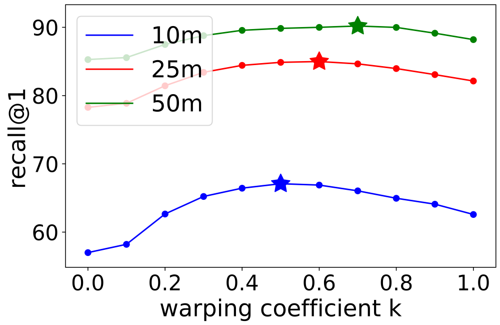

Figure 6. Ablation of warping coefficient k. Higher k correspond to more aggressive warping, which is useful when a rougher estimate of the position is needed.

We also perform a second ablation study focused on the parameter k used in the self-supervised training method. Figure 6 shows the recall@1 curves at different values of k. We observe that higher values of k are better for coarser localization problems. This is due to the fact that a high k produces more difficult training images, inducing the warping regression module to learn more aggressive homography transformations. On the other hand, when the goal is to achieve a more precise localization it is undesirable to warp too much the query-prediction pairs, and a lower k is more appropriate. Qualitative results of the effect of the parameter k are reported in the supplementary material.

5. Conclusions

We have presented GeoWarp, a novel method for image matching tailored for visual geolocalization. GeoWarp combines the robustness of deep dense local features to clutter and lighting variations with a learnable architecture that provides invariance to viewpoint shifts. This novel architecture is trained with novel loss functions that only require unlabeled data or can take advantage of weak labels. We evaluate GeoWarp on two well-established datasets, demonstrating the soundness of our approach against state-of-the-art retrieval and geolocalization methods, over which we show significant improvements.

Acknowledgements We acknowledge that the research activity herein was carried out using the IIT HPC infrastructure.

References

[1] Relja Arandjelovic and Andrew Zisserman. Three things everyone should know to improve object retrieval. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2911–2918. IEEE Computer Society, 2012. 2

[2] Relja Arandjelovic and Andrew Zisserman. All about vlad. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1578–1585. IEEE Computer Society, 2013. 2, 5

[3] R. Arandjelović and Andrew Zisserman. Dislocation: Scalable descriptor distinctiveness for location recognition. In *ACCV*, 2014. 7

[4] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(6):1437–1451, 2018. 1, 2, 3, 7, 8

[5] Charbel Azzi, Daniel C. Asmar, Adel H. Fakih, and John S. Zelek. Filtering 3d keypoints using gist for accurate image-based localization. In *Brit. Mach. Vis. Conf.* BMVA Press, 2016. 1

[6] 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

[7] Gabriele Moreno Berton, Valerio Paolicelli, Carlo Masone, and Barbara Caputo. Adaptive-attentive geolocalization from few queries: A hybrid approach. In *IEEE Winter Conf. Appl. Comput. Vis.*, pages 2918–2927, January 2021. 5

[8] Eric Brachmann, Alexander Krull, Sebastian Nowozin, Jamie Shotton, Frank Michel, Stefan Gumhold, and Carsten Rother. Dsac - differentiable ransac for camera localization. *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017. 2

[9] Eric Brachmann and Carsten Rother. Learning less is more - 6d camera localization via 3d surface regression. In *IEEE Conf. Comput. Vis. Pattern Recog.*, June 2018. 2

[10] Eric Brachmann and Carsten Rother. Neural- Guided RANSAC: Learning where to sample model hypotheses. In *Int. Conf. Comput. Vis.*, 2019. 2

[11] Bingyi Cao, A. Araujo, and Jack Sim. Unifying deep local and global features for image search. In *Eur. Conf. Comput. Vis.*, 2020. 1, 2, 7, 8

[12] 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 Conf. Comput. Vis. Pattern Recog.*, pages 737–744, 2011. 5

[13] Ondrej Chum, James Philbin, Josef Sivic, Michael Isard, and Andrew Zisserman. Total recall: Automatic query expansion with a generative feature model for object retrieval. In *Int. Conf. Comput. Vis.*, pages 1–8. IEEE Computer Society, 2007. 2

[14] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Deep image homography estimation. 2016. 2

[15] Michael Donoser and Horst Bischof. Diffusion processes for retrieval revisited. In *IEEE Conf. Comput. Vis. Pattern Recog.*, June 2013. 2

[16] M. Fischler and R. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 2

[17] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, June 2016. 6

[18] A. Gordo, J. Almazan, J. Revaud, and D. Larlus. End-to-end learning of deep visual representations for image retrieval. *Int. J. Comput. Vis.*, 2017. 2, 7, 8

[19] Richard I. Hartley. In defense of the eight-point algorithm. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(6):580–593, 1997. 3

[20] James Hays and Alexei A. Efros. Im2gps: estimating geographic information from a single image. In *IEEE Conf. Comput. Vis. Pattern Recog.* IEEE Computer Society, 2008. 1

[21] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 770–778, 2016. 7

[22] Angjoo Kanazawa, W. David Jacobs, and Manmohan Chandraker. Warpnet: Weakly supervised matching for single-view reconstruction. *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016. 2

[23] T. Kanji. Self-localization from images with small overlap. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4497–4504, 2016. 1

[24] Hyo Jin Kim, Enrique Dunn, and Jan-Michael Frahm. Learned contextual feature reweighting for image geo-localization. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017. 1, 2, 7

[25] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Adv. Neural Inform. Process. Syst.*, volume 25. Curran Associates, Inc., 2012. 7

[26] David Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.*, 60:91–, 11 2004. 2

[27] W. Maddern, G. Pascoe, C. Linegar, and P. Newman. 1 Year, 1000km: The Oxford RobotCar Dataset. *The International Journal of Robotics Research*, 2017. 5

[28] C. Masone and B. Caputo. A survey on deep visual place recognition. *IEEE Access*, 9:19516–19547, 2021. 1

[29] Piotr Mirowski, Matthew Koichi Grimes, Mateusz Malinowski, Karl Moritz Hermann, Keith Anderson, Denis Teplyashin, Karen Simonyan, Koray Kavukcuoglu, Andrew Zisserman, and Raia Hadsell. Learning to navigate in cities without a map. In *Adv. Neural Inform. Process. Syst.*, 2018. 5

[30] Tony Ng, Vassileios Balntas, Yurun Tian, and Krystian Mikolajczyk. SOLAR: Second-order loss and attention for image retrieval. In *Eur. Conf. Comput. Vis.*, 2020. 2

[31] Hyeonwoo Noh, Andre Araujo, Jack Sim, Tobias Weyand, and Bohyung Han. Large-scale image retrieval with attentive deep local features. In *Int. Conf. Comput. Vis.*, 2017. 2, 7

[32] J. Philbin, O. Chum, M. Isard, Josef Sivic, and Andrew Zisserman. Object retrieval with large vocabularies and fast spatial matching. *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1–8, 2007. 2

[33] F. Radenović, G. Tolias, and O. Chum. Fine-tuning CNN image retrieval with no human annotation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2018. 2, 3, 7

[34] Anita Rau, Guillermo Garcia-Hernando, Danail Stoyanov, Gabriel J. Brostow, and Daniyar Turmukhambetov. Predicting visual overlap of images through interpretable non-metric box embeddings. In *European Conference on Computer Vision (ECCV)*, 2020. 2

[35] I. Rocco, R. Arandjelović, and J. Sivic. Convolutional neural network architecture for geometric matching. *IEEE Trans. Pattern Anal. Mach. Intell.*, pages 2553–2567, 2018. 2, 4, 5, 7, 8

[36] B. C. Russell, J. Sivic, J. Ponce, and H. Dessales. Automatic alignment of paintings and photographs depicting a 3d scene. In *Int. Conf. Comput. Vis. Worksh.*, pages 545–552, 2011. 1

[37] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: Learning feature matching with graph neural networks. In *CVPR*, 2020. 2, 7, 8

[38] Torsten Sattler, Will Maddern, Carl Toft, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, Fredrik Kahl, and Tomas Pajdla. Benchmarking 6dof outdoor visual localization in changing conditions. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018. 1, 7

[39] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *Int. Conf. Learn. Represent.*, 2015. 7

[40] Hajime Taira, Masatoshi Okutomi, Torsten Sattler, Mircea Cimpoi, Marc Pollefeys, Josef Sivic, Tomas Pajdla, and Akihiko Torii. InLoc: Indoor Visual Localization with Dense Matching and View Synthesis. In *IEEE Conf. Comput. Vis. Pattern Recog.*, Salt Lake City, United States, June 2018. 1, 2, 3, 7, 8

[41] Giorgos Tolias, Ronan Sicre, and Hervé Jégou. Particular Object Retrieval With Integral Max-Pooling of CNN Activations. In *Int. Conf. Learn. Represent.*, Int. Conf. Learn. Represent., pages 1–12, San Juan, Puerto Rico, May 2016. 2

[42] A. Torii, R. Arandjelović, J. Sivic, M. Okutomi, and T. Pajdla. 24/7 place recognition by view synthesis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(2):257–271, 2018. 1, 2, 3, 5, 7, 8

[43] A. Torii, J. Sivic, M. Okutomi, and T. Pajdla. Visual place recognition with repetitive structures. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(11):2346–2359, 2015. 7

[44] 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 Conf. Comput. Vis. Pattern Recog.*, June 2020. 5, 7

[45] Fan Yang, Ryota Hinami, Yusuke Matsui, Steven Ly, and Shin'ichi Satoh. Efficient image retrieval via decoupling diffusion into online and offline processing. *AAAI*, 33:9087–9094, 07 2019. 2, 7, 8

[46] A.R. Zamir and M. Shah. Image geo-localization based on multiple nearest neighbor feature matching using generalized graphs. *IEEE Trans. Pattern Anal. Mach. Intell.*, PP(99):1–1, 2014. 5

[47] Zichao Zhang, Torsten Sattler, and D. Scaramuzza. Reference pose generation for long-term visual localization via learned features and view synthesis. *Int. J. Comput. Vis.*, 2020. 3