Deep Homography Estimation for Visual Place Recognition

Feng Lu[1,2], Shuting Dong[1,2], Lijun Zhang[3], Bingxi Liu[2,4], Xiangyuan Lan[2*], Dongmei Jiang[2], Chun Yuan[1,2*]

[1]Tsinghua Shenzhen International Graduate School, Tsinghua University
[2]Peng Cheng Laboratory
[3]Chongqing Institute of Green and Intelligent Technology, Chinese Academy of Sciences
[4]Southern University of Science and Technology
{lf22@mails, dst21@mails, yuanc@sz}.tsinghua.edu.cn, zhanglijun@cigit.ac.cn, {liubx, lanxy, jiangdm}@pcl.ac.cn

Abstract

Visual place recognition (VPR) is a fundamental task for many applications such as robot localization and augmented reality. Recently, the hierarchical VPR methods have received considerable attention due to the trade-off between accuracy and efficiency. They usually first use global features to retrieve the candidate images, then verify the spatial consistency of matched local features for re-ranking. However, the latter typically relies on the RANSAC algorithm for fitting homography, which is time-consuming and non-differentiable. This makes existing methods compromise to train the network only in global feature extraction. Here, we propose a transformer-based deep homography estimation (DHE) network that takes the dense feature map extracted by a backbone network as input and fits homography for fast and learnable geometric verification. Moreover, we design a reprojection error of inliers loss to train the DHE network without additional homography labels, which can also be jointly trained with the backbone network to help it extract the features that are more suitable for local matching. Extensive experiments on benchmark datasets show that our method can outperform several state-of-the-art methods. And it is more than one order of magnitude faster than the mainstream hierarchical VPR methods using RANSAC. The code is released at https://github.com/Lu-Feng/DHE-VPR.

Introduction

Visual place recognition (VPR), also known as visual geo-localization (Berton et al. 2022) or image localization (Liu, Li, and Dai 2019), is one of the research hotspots in robotics and computer vision communities. VPR aims to coarsely estimate the location of the query image (i.e. the current location of the mobile robot), which is commonly achieved using image retrieval methods on a database of geo-tagged images. When designing a robust VPR method, there are two challenging problems to consider: 1) Due to condition (e.g., light, weather, and season) and viewpoint variations, images captured at the same place may change significantly over time. 2) Images captured at different places can be similar, which may lead to perceptual aliasing (Lowry et al. 2016).

The VPR implementation process typically involves image retrieval and feature matching (Arandjelovic et al. 2016; Cao et al. 2020), with global or/and local features to represent place images. The global features can also be got by aggregating local features into compact feature vectors (Jégou et al. 2010), which are robust to viewpoint change and applicable for large-scale VPR. However, methods only using such features are prone to suffer from perceptual aliasing, because it neglects the spatial information of aggregated local features. This issue can be solved by matching local features with geometric verification, but it is time-consuming. A compromise pipeline (Hausler et al. 2021), called hierarchical or two-stage VPR, first retrieves top-k candidate places using global features, then re-ranks them via local feature matching between the query and candidate images.

However, in the re-ranking stage of hierarchical VPR, the matched local feature pairs based on similarity searching commonly contain some incorrect pairs (i.e., outliers). The RANSAC algorithm (Fischler and Bolles 1981) is typically used to fit the homography and remove outliers. The process of RANSAC is to iteratively sample four matched pairs at random to solve corresponding homography transformations and find the one that meets the most inliers. It is time-consuming and non-differentiable. Although there

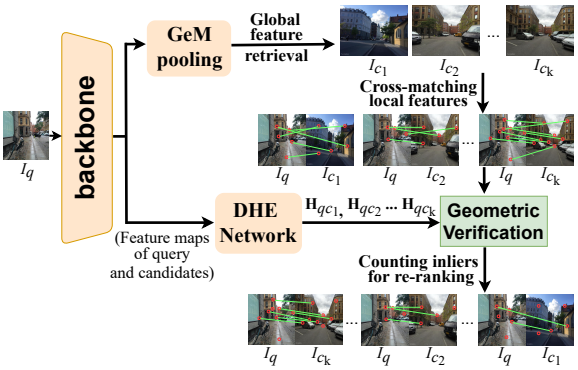

Figure 1: The two-stage place retrieval with the proposed architecture. The backbone is applied to extract feature maps. The top branch yields global features for retrieving top-k candidate images. The bottom branch employs the local features for cross-matching and the DHE network for geometric verification via regressing homography. We count inliers as image similarity for re-ranking candidates.

*Corresponding authors.

are some differentiable RANSAC models (Brachmann et al. 2017, 2019, 2021), they are inapplicable to the VPR task. This leads to some state-of-the-art (SOTA) VPR methods (Hausler et al. 2021; Wang et al. 2022) compromising to train the network only in global feature extraction. Deep homography estimation (**DHE**) (Nguyen et al. 2018), which uses deep neural networks to fit homography, is a more viable solution. GeoWarp (Berton et al. 2021) first used a similar way in VPR to align two different views of a same place in the urban scene. Regardless of the existing deep homography methods or the GeoWarp work, they all regress a homography to align two images of a planar scene (or approximated as a plane). However, many scenes in VPR (e.g., suburban scene) do not satisfy this condition. In fact, fitting appropriate homography to remove outliers and use the number of inliers as image similarity, as in SOTA two-stage VPR works (Hausler et al. 2021; Wang et al. 2022), is suitable for more scenes in VPR and can achieve better performance.

In this paper, we propose a transformer-based DHE network that takes the dense feature map extracted by a backbone network as input and regresses a homography matrix to decide the inliers of matched local feature pairs as image similarity (see Fig. 1). Given that none of the VPR training datasets have homography annotations, we propose a re-projection error of inliers (REI) loss to train the DHE network using the reference number of inliers provided by RANSAC as supervision. This process can be jointly trained with the backbone, making the feature map extracted by the backbone more suitable for local feature matching. Since our method uses the DHE network (without RANSAC) in inference, it can also make the re-ranking stage much faster.

We name our proposed method DHE-VPR. The main **contributions** can be highlighted as follows:

1) We introduce a novel hierarchical VPR architecture, in which a DHE network (rather than RANSAC) is adopted to fit homography to count inliers of matched local feature pairs. This makes up for the time-consuming and non-differentiable defect of RANSAC.

2) We propose a re-projection error of inliers loss to train the DHE network without additional homography labels. The error can be back-propagated to the backbone, making the learned local features more suitable for re-ranking.

3) Extensive experiments show that the proposed method outperforms several SOTA methods. And it is more than one order of magnitude faster than the existing two-stage VPR methods with RANSAC-based geometric verification.

Related Work

One-Stage VPR: The early VPR methods commonly got the most similar place images through direct retrieval without considering re-ranking. We call them one-stage VPR. These methods represented the place image with global descriptors produced by aggregating local descriptors or processing the whole image. For example, aggregation algorithms like Bag of Words (Angeli et al. 2008) and VLAD (Jégou et al. 2010; Lowry and Andreasson 2018; Khaliq et al. 2020) have been employed to aggregate local descriptors such as SURF (Bay et al. 2008). With the significant advancements of deep learning in computer vision tasks, many VPR methods (Sünderhauf et al. 2015; Garg et al. 2017, 2018; Camara et al. 2020; Arandjelovic et al. 2016; Liu et al. 2021; Chen et al. 2017a; Yin et al. 2019; Naseer et al. 2017; Ali-bey et al. 2022; Berton, Masone, and Caputo 2022) have opted to use deep features for image representation to achieve better performance. Similarly, several works (Arandjelovic et al. 2016; Hou et al. 2018; Peng et al. 2021; Yu et al. 2019) also incorporated the traditional aggregation models into neural networks. However, relying solely on aggregated features often leads to perceptual aliasing due to the lack of spatial information. To overcome this, some approaches utilized image sequence matching (Milford and Wyeth 2012; Hansen et al. 2014; Naseer et al. 2014; Doan et al. 2019; Lu et al. 2021; Garg and Milford 2021) to achieve robust VPR under extreme variations in illumination, weather, and season. And other methods (Sünderhauf et al. 2015; Chen et al. 2017b; Xin et al. 2019; Gao et al. 2020) mined discriminative landmarks for VPR. Moreover, VPR models commonly require training on large-scale place datasets with weak supervision. CRN (Jin Kim et al. 2017) and SFRS (Ge et al. 2020) mined hard positive samples instead of the simplest positive sample (Arandjelovic et al. 2016) for training more robust VPR networks.

Two-Stage VPR: More recently, the hierarchical strategy with re-ranking candidate images for VPR (Hausler and Milford 2020; Hausler et al. 2021; Berton et al. 2021; Wang et al. 2022; Lu et al. 2023; Zhang et al. 2023) has gained more attention. The hierarchical (two-stage) VPR methods commonly first searched top-k candidate places using compact global descriptors, and then re-ranked candidates by local feature matching. The global features are typically produced using aggregation/pooling methods, e.g., NetVLAD (Arandjelovic et al. 2016) and Generalized Mean (GeM) pooling (Radenović, Tolias, and Chum 2018). The re-ranking stage usually involves geometric verification (Hausler et al. 2021; Cao et al. 2020; Wang et al. 2022), which takes into account the spatial relationship of local features and thus is able to address the perceptual aliasing encountered by global features. However, most of them required the RANSAC algorithm (Fischler and Bolles 1981) for fitting homography, which is time-consuming and non-differentiable. GeoWarp (Berton et al. 2021) first applied a convolutional neural network to regress a homography transformation for aligning two images taken at a same place with different viewpoints. However, the aligned two images need to be in a plane, which can be satisfied mainly in the urban environment. Our work can be seen as a method that draws on the advantages of both geometric validation and deep homography to achieve better performance and higher efficiency, and is also applicable to various scenarios in VPR.

Deep Homography Estimation: Homography estimation is a basic problem in computer vision. Most traditional methods used Direct Linear Transform (Hartley et al. 2003) with RANSAC outlier rejection for homography estimation. In recent years, there has been a surge of interest in developing deep neural networks for this task. DeTone et al. (DeTone et al. 2016) designed a VGG-style network to estimate homography and demonstrated its effectiveness. Nguyen et al. (Nguyen et al. 2018) proposed an unsuper-

vised approach that optimizes the DHE network by minimizing the pixel-wise intensity error between a warped input image and the other image. Further, Zhang et al. (Zhang et al. 2020) began to calculate loss (i.e. image distance) in feature space instead of pixel space. Koguciuk et al. (Koguciuk et al. 2021) presented a bidirectional implicit homography estimation loss for unsupervised training. Le et al. (Le et al. 2020) developed a model that can jointly estimate the homography and dynamics masks to handle dynamic scenes. However, all these works are fitting homography to align images. To the best of our knowledge, our work is among the first to use neural networks to fit homography for geometric check. The homography matrices used for alignment and geometric verification are not necessarily the same. And the loss in our method is based on the re-projection error, which is different from the loss based on image distance in the above methods.

Methodology

Problem Formulation

Given a query image I_q of a previously visited place, the task of a VPR system is to find its best match from a database of geo-tagged place images $\mathcal{D} = \{I_i\}$. The two-stage VPR methods typically first perform a similarity retrieval over $\mathcal{D}$ in the space of the global features to yield a set of candidate images $\mathcal{C} = \{I_c\}$ ($\mathcal{C} \subset \mathcal{D}$), i.e. search top-k candidates. Then, the local matching (and geometric verification) algorithms are used to re-rank the candidate images in $\mathcal{C}$ based on local features. Specifically, we use a neural network as unified feature extractor to get the feature map $\boldsymbol{f} \in \mathbb{R}^{W \times H \times C}$ ($weight \times height \times channel$) of each place image. In the first stage, $\boldsymbol{f}$ is aggregated/pooled into a compact vector as global feature. In the second stage, $\boldsymbol{f}$ is directly viewed as a dense $W \times H$ grid of C-dimensional local features for re-ranking. The local features are L2-normalized.

Architecture Overview

Due to the superiority of Vision Transformer (Dosovitskiy et al. 2020) in capturing feature dependencies over long distances, the Compact Convolutional Transformer (CCT) (Hassani et al. 2021) is used as the unified feature extractor (i.e. backbone) in this work. Its output is a $M \times C$-dimensional tensor (M means the number of patch tokens), which can be reshaped into the feature map $\boldsymbol{f} \in \mathbb{R}^{W \times H \times C} (W \times H = M)$ to restore spatial position. As shown in Fig. 1, the proposed architecture consists of two branches. In the above branch, GeM pooling (Radenović, Tolias, and Chum 2018) is utilized to aggregate the feature map into C-dimensional vector, i.e., global feature. Then L2 distance is used to measure the global feature distance between the query image I_q and each reference image I_i in $\mathcal{D}$ to get candidate image set $\mathcal{C}$. The bottom branch is primarily composed of our proposed DHE network, the inputs of which are the feature maps (i.e. dense local features) of the query and candidate images. It uses the homography estimated by the DHE network to check the geometric consistency of matched local feature pairs between the query I_q and each candidate I_c in $\mathcal{C}$. The number of inliers is used as the similarity of image pairs to re-rank candidate images.

Deep Homography Estimation Network

Based on the multi-view geometry theory, we can use a homography matrix to associate two images presenting a same planar scene or captured by a rotational camera. When homogeneous coordinates are used to denote points, the point $(u, v)^T$ can be represented as $(u, v, 1)^T$. Meanwhile, the homogeneous coordinates $(x, y, z)^T$ and $(x/z, y/z, 1)^T$ denote a same point. Given two points $\mathbf{x} = (u, v, 1)^T$ and $\mathbf{x}' = (u', v', 1)^T$, we can use a non-singular 3×3 matrix $\mathbf{H}$ to express the homography transformation mapping $\mathbf{x} \leftrightarrow \mathbf{x}'$:

$$\begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} \simeq \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \quad \text{or} \quad \mathbf{x}' \simeq \mathbf{H}\mathbf{x}. \tag{1}$$

Even when multiplied by any non-zero scale factor, $\mathbf{H}$ does not change the projective transformation, and only the ratio of matrix elements is meaningful. We can set $h_{33} = 1$ and keep 8 independent ratios in $\mathbf{H}$ as 8 degrees of freedom. So the homography matrix $\mathbf{H}$ can be solved using 4 pairs (non-collinear) of corresponding points in two images.

Previous deep homography works utilize deep neural networks to output the homography for warping an image to another. In VPR, the building facades in urban environments are typically planes, so GeoWarp (Berton et al. 2021) employed deep homography to warp two place images taken from different viewpoints of the same place to a closer geometrical space. However, in more general scenes, the content of the whole image or a large region of the image is not exactly in a plane, so aligning two place images with homography is ineffective. Here, we regress a homography matrix to check spatial consistency instead of aligning images. The purpose of this work is different from previous deep homography works. However, we use a network to regress homography, which can learn from previous works.

As shown in Fig. 2, feature maps $\boldsymbol{f}_q$ and $\boldsymbol{f}_c$ of the query image I_q and a candidate image I_c are first extracted using feature extractor (i.e. the backbone), and then are fed to our DHE network. The DHE network consists of two modules: Similarity Matching Module and Homography Regression Module. The former is a parameterless and differentiable operation. The latter is a learnable neural network.

The Similarity Matching Module S utilizes the cosine similarity to compute the similarity map $\boldsymbol{s}_{qc} \in \mathbb{R}^{M \times M}$ between local features pairs of $\boldsymbol{f}_q$ and $\boldsymbol{f}_c$. The M is the number of local features (i.e. patch tokens for Transformer) in each image. The $\boldsymbol{s}_{qc}$ can also be expressed as $S(\boldsymbol{f}_q, \boldsymbol{f}_c)$. Since the local features have been L2-normalized, the cosine similarity can be replaced by the inner product. Formally,

$$\boldsymbol{s}_{qc}(i, j) = \boldsymbol{f}_q(i)^T \boldsymbol{f}_c(j) \quad i, j \in M. \tag{2}$$

The Homography Regression Module applies the similarity map $\boldsymbol{s}_{qc}$ plus a learnable position embedding $\boldsymbol{E}_{pos} \in \mathbb{R}^{M \times M}$ as input and yields the homography matrix $\mathbf{H}_{qc}$ for following geometric verification. However, it is difficult to directly regress the elements in $\mathbf{H}$ due to the high variance in their magnitude. As a result, it is common to use networks to regress the 4 pairs of corresponding points (or 4-point offsets) in two images (DeTone et al. 2016; Nguyen et al. 2018;

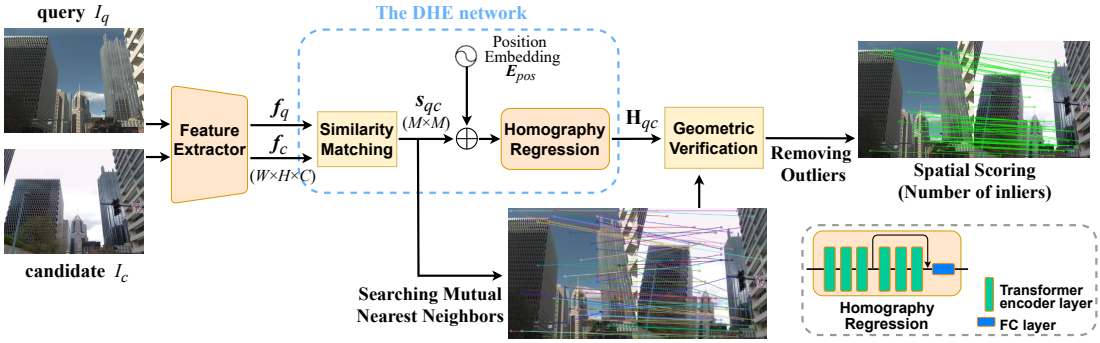

Figure 2: Diagram of our re-ranking process with the DHE network. The feature maps $\boldsymbol{f}_q$ and $\boldsymbol{f}_c$ of the query image I_q and a candidate image I_c are fed into the Similarity Matching Module to compute the similarity map $\boldsymbol{s}_{qc}$. Then the Homography Regression Module uses the $\boldsymbol{s}_{qc}$ to yield the homography matrix $\mathbf{H}_{qc}$ for geometric verification of mutual matches.

Berton et al. 2021). Inspired by this, this module first computes the 4 pairs of points (4-point correspondence) on the images I_q and I_c, i.e., a 16D vector. This process is implemented by using 6 stacked transformer encoder layers followed by a fully connected layer, and a shortcut connection is used between the third encoder layer and the sixth encoder layer. The transformer encoder can provide a larger effective perceptual field on the similarity map. And it is denoted as

$$\begin{aligned} TransE(\boldsymbol{s}_{qc}+\boldsymbol{E}_{pos}) &= [\underbrace{\boldsymbol{p}_{q1},\ldots,\boldsymbol{p}_{q4}}_{\boldsymbol{P}_q},\underbrace{\boldsymbol{p}_{c1},\ldots,\boldsymbol{p}_{c4}}_{\boldsymbol{P}_c}] \\ &= [\boldsymbol{P}_q,\boldsymbol{P}_c], \end{aligned} \tag{3}$$

where $\boldsymbol{p}_{q1},\ldots,\boldsymbol{p}_{q4}$ and $\boldsymbol{p}_{c1},\ldots,\boldsymbol{p}_{c4}$ are four points on I_q and I_c, respectively. $[\boldsymbol{P}_q,\boldsymbol{P}_c]$ is their concatenation.

After obtaining 4 pairs of points on two images, we use the Direct Linear Transform (Hartley et al. 2003) to yield the homography $\mathbf{H}_{qc}$, which is represented as

$$\mathbf{H}_{qc} = DLT(\mathbf{P}_q,\mathbf{P}_c). \tag{4}$$

We succinctly denote the whole process of this module as

$$\mathbf{H}_{qc} = R(\boldsymbol{s}_{qc}). \tag{5}$$

The whole process of the DHE network is summarized as

$$\mathbf{H}_{qc} = DHE(\boldsymbol{f}_q,\boldsymbol{f}_c) = R\big(S(\boldsymbol{f}_q,\boldsymbol{f}_c)\big). \tag{6}$$

Re-ranking with Geometric Verification

In the re-ranking stage, we first calculate the mutual nearest neighbor matching of local features through exhaustive cross-matching. Then the homography matrix yielded by the DHE network is used to check geometric verification and remove outliers. Finally, we use the number of inliers as the similarities of image pairs to re-rank the candidates.

The similarity map $\boldsymbol{s}_{qc}$ between the query I_q and a candidate I_c has been computed. We apply it to search mutual nearest neighbor matches set $\mathcal{MN}$, which is defined as

$$\begin{aligned} \mathcal{MN} = \{(x,y):\ & x = \arg\max_i \boldsymbol{s}_{qc}(i,y), \\ & y = \arg\max_j \boldsymbol{s}_{qc}(x,j)\}. \end{aligned} \tag{7}$$

That is, if the most similar local feature in image I_q of the feature $\boldsymbol{f}_c(y)$ is $\boldsymbol{f}_q(x)$, and the most similar feature in image I_c of the feature $\boldsymbol{f}_q(x)$ is $\boldsymbol{f}_c(y)$, then $(x,y)\in\mathcal{MN}$.

The x and y are the indices of corresponding patches. Assume that each patch corresponds to the Cartesian coordinates of a 2D image point in the center of the patch. We denote the homogeneous coordinates of a pair of matched patches as $\mathbf{p}_q$ and $\mathbf{p}_c$. If the re-projection error is less than a threshold θ, then it is an inlier. That is, an inlier satisfies

$$\|C(\mathbf{p}_c) - C(\mathbf{H}_{qc}\mathbf{p}_q)\| \le \theta, \tag{8}$$

where $C(\cdot)$ means converting homogeneous coordinates to Cartesian (inhomogeneous) coordinates. We re-rank the candidate images using the number of inliers between the query and candidates.

Training Strategy

In common deep homography works, an image can be aligned with another image after using the homography transformation because the two images present a (approximately) same planar scene. So, they can train networks by minimizing the distance between two aligned images. But this prior condition does not hold for place images in many scenes. Meanwhile, we also cannot directly obtain homography labels to supervise the training of our DHE network.

One possible solution is to train the network with the label provided by RANSAC. However, we found that the network is difficult to converge whether using the homography parameters yielded by RANSAC for direct supervision, or using the selected inliers and their re-projection errors for supervision. We propose to use only the number of inliers provided by RANSAC as supervision information, let the network autonomously decide which matched pairs are inliers, and optimize the network by minimizing the average re-projection error of these "inliers". This makes it easier for networks to converge. Meanwhile, the loss produced by this process has the potential to optimize the feature extractor through back-propagation, so that the extracted feature maps are more suitable for local matching and outperform the method directly using RANSAC.

We first initialize the backbone and the DHE network individually. The initialization of the backbone follows the

Dataset	Description	Variation	
		Condition	Viewpoint
MSLS	long-term, urban, suburban	✓	✓
Pitts30k	urban, panorama	✓	✓
Nordland	suburban, natural, seasonal	✓	×
St. Lucia	suburban	✓	✓

Table 1: Summary of the datasets.

Method	Pitts30k			MSLS val		
	R@1	R@5	R@10	R@1	R@5	R@10
GeM	83.1	92.8	95.2	80.5	90.4	92.3
GeM+DHE	87.8	94.3	95.9	81.2	90.1	92.8
GeM+ransac	88.3	94.4	95.8	81.8	**91.6**	**93.1**
GeM*	83.9	93.3	95.4	80.3	90.0	92.3
GeM+DHE*	**89.4**	**95.1**	**96.2**	**84.1**	91.4	**93.1**

Table 2: Ablations on geometric check and training strategy.

training procedure of NetVLAD using triplet loss. That is

$$L_g = \sum_j l\left(d_G\left(q, p^q\right) + m - d_G\left(q, n_j^q\right)\right), \tag{9}$$

where $l(x) = \max(x, 0)$. m is the margin. d_G is the L2 distance between global features of two images. q, p^q, and n_j^q are the query, positive sample, and hard negative sample.

The feature maps extracted from the trained backbone are fed to the DHE network to yield homography $\mathbf{H}$. Meanwhile, we use Eq. 7 to compute the mutual matches of local features in the positive image pair, and apply RANSAC to decide the number of inliers N. For all mutual matches, $\mathbf{H}$ is used for transformation. N pairs of matched points with the smallest re-projection error are regarded as "inliers". The set of the homogeneous coordinates of these "inliers" is denoted as $\mathcal{IN}$. We aim to minimize their re-projection errors. So we design a re-projection error of inliers (REI) loss for the DHE network training. That is

$$L_r = \frac{\sum_{(\hat{\mathbf{p}}_c, \hat{\mathbf{p}}_q) \in \mathcal{IN}} \|C(\hat{\mathbf{p}}_c) - C(\mathbf{H}_{qc}\hat{\mathbf{p}}_q)\|}{N}. \tag{10}$$

After the initialization of the backbone and DHE network, they are fine-tuned together using the joint loss:

$$L = L_g + \lambda L_r, \tag{11}$$

where λ is a weight. Note that when we initialize the DHE network, the backbone is frozen. But when we combine them for fine-tuning, the parameters of the last few layers of the backbone are updatable.

Experiments

Datasets and Performance Evaluation

We conduct experiments using multiple VPR datasets: **MSLS** (Warburg et al. 2020), **Pitts30k** (Torii et al. 2013), **Nordland** (downsampled test set with 224x224 image size) (Olid et al. 2018), and **St. Lucia** (Berton et al. 2022). Table 1 summarizes their main information. The Recall@N (R@N) is used to assess model performance, which calculates the percentage of queries that have at least one of the top-N retrieved reference images taken within a certain threshold of ground truth. Following common procedure (Wang et al. 2022; Berton et al. 2022), the threshold is 25m and 40° for MSLS (including MSLS val and MSLS challenge), 25m for Pitts30k and St. Lucia, and ±2 frames for Nordland.

Implementation Details

In DHE-VPR architecture, the CCT-14 model pre-trained on ImageNet (Deng et al. 2009) is used as the backbone. The transformer encoder layers after the 8th layer are removed, before the 3rd layer are frozen. We resize the input image to 384×384 pixels and get 24×24×384-D feature maps (the global features are 384-D). We re-rank the top-32 candidates to yield final results. The re-projection error threshold θ of the inlier is set to 1.5 times the patch size for RANSAC, and 3 times the patch size for geometric verification using DHE (in inference). The margin m in Eq. 9 is set to 0.1, and the weight λ in Eq. 11 is 100. Experiments are implemented using PyTorch on an NVIDIA GeForce RTX 3090 GPU. For the initialization of the DHE network, the Adam optimizer is used with learning rate = 0.0001 (multiplied by 0.8 after every 5 epochs) and batch size = 16. We train the network for 100 epochs (2k iterations per epoch) on MSLS-train. The implementation of the backbone initialization and the fine-tuning of entire model basically follows the benchmark (Berton et al. 2022), with learning rate = 0.00001 and batch size = 4. For the backbone initialization, we train CCT-14 on MSLS-train for MSLS, Nordland, and St. Lucia, and further train it on Pitts30k-train for Pitts30k. For fine-tuning, the DHE network and the last 2 encoder layers in backbone are updatable. The model for Pitts30k is fine-tuned on Pitts30k-train for 40 epochs (5k iterations per epoch), while the model for others is fine-tuned on MSLS-train for 2 epochs (10k iterations per epoch). We use 2 hard negative images in a triplet.

Ablation Study

We conduct several ablation experiments on the Pitts30k and MSLS (val) datasets to validate the design of our DHE network and training strategy. We demonstrate the effectiveness by comparing performance before and after using the DHE network for re-ranking, as well as before and after using our proposed training strategy for fine-tuning. We also use the solution with RANSAC-based re-ranking as a reference.

- **GeM:** Direct retrieval with the GeM global feature.
- **GeM+DHE:** GeM feature is used to retrieve candidates, and the homography estimated by DHE network is used to check spatial consistency for re-ranking. The backbone and DHE network are trained independently.
- **GeM+RANSAC:** GeM and RANSAC are used for candidates retrieving and re-ranking, respectively.
- **GeM*:** Direct retrieval with the GeM feature in our DHE-VPR model. The backbone and DHE network are jointly fine-tuned with our training strategy.
- **GeM+DHE*:** GeM and DHE are used for candidates retrieving and re-ranking, respectively. The backbone and DHE network are jointly fine-tuned with the proposed training strategy. i.e. our complete method.

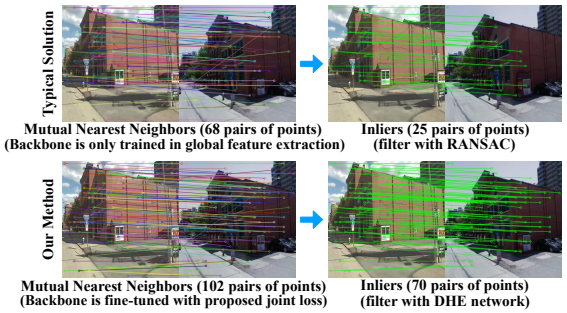

Figure 3: Qualitative comparison of typical solution and our method. The top is the typical solution using RANSAC, which is non-differentiable, i.e. the backbone is only trained in global feature extraction. The bottom is ours, which yields more mutual nearest neighbors and inliers than the top. (When the re-projection threshold θ is set to 1.5× the patch size, ours yields 39 inliers, which is also more than the top.)

Table 2 displays the results of different ablated versions. Before fine-tuning with our training strategy (i.e., GeM, GeM+DHE, GeM+RANSAC), either the solution of using DHE network or RANSAC to check geometric consistency for re-ranking improves the performance on the one-stage retrieval (GeM). This indicates that it is feasible for us to use the DHE network to fit homography for geometric verification. After fine-tuning with our proposed training strategy (i.e., GeM* and GeM+DHE*), there is no significant difference in performance between GeM* and GeM, while GeM+DHE* outperforms GeM+DHE. The former shows that fine-tuning with the joint loss does not affect the performance of global features. Meanwhile, the latter indicates that the joint fine-tuning strategy enables the backbone to output better local features (i.e. the features are more suitable for local matching). GeM+DHE* surpasses GeM+RANSAC, demonstrating the superiority of differentiable deep homography over non-differentiable RANSAC. That is, thanks to the differentiable DHE process, the error can assist the backbone in parameter updating via back-propagation. This leads to that although our proposed REI loss uses the number of inliers yielded by RANSAC as supervision, our method outperforms the typical method directly using RANSAC. Fig. 3 depicts this visually. The (correct) mutual nearest neighbors of our method are significantly more than that of the typical solution, resulting in a larger number of inliers of our method than the typical solution after geometric verification.

The comparison experiment of different strategies with deep homography for re-ranking in VPR (i.e. our DHE-VPR uses deep homography for spatial verification, while GeoWarp (Berton et al. 2021) uses it for image alignment.) is shown in our arXiv version of this paper, which illustrates that our DHE-VPR can significantly outperform GeoWarp (especially for the suburban scene) with the same settings.

Comparisons with SOTA Methods

We compared our DHE-VPR method against several SOTA VPR algorithms, including three one-stage VPR methods using global features for direct retrieval: NetVLAD (Arandjelovic et al. 2016), SFRS (Ge et al. 2020) and CosPlace (Berton, Masone, and Caputo 2022), as well as four two-stage VPR methods with re-ranking: SP-SuperGlue (DeTone et al. 2018; Sarlin et al. 2020), Patch-NetVLAD (Hausler et al. 2021), TransVPR (Wang et al. 2022) and ETR-D (Zhang et al. 2023). We also show the direct retrieval result using GeM in our pipeline (denoted as GeM*).

Two Patch-NetVLAD versions are used in our experiments, i.e., Patch-NetVLAD-s (speed-focused) and Patch-NetVLAD-p (performance-focused). SP-SuperGlue is first used for VPR in Patch-NetVLAD work, in which NetVLAD is used to retrieve candidates, and the SuperGlue (DeTone et al. 2018) matcher is applied to match the SuperPoint (Sarlin et al. 2020) key-point features for re-ranking. TransVPR and ETR-D employ the transformer model (same as ours). Patch-NetVLAD-p and TransVPR fit homography for geometric check, but they use RANSAC, whereas we use deep homography. Besides, they use the multi-scale fusion (Hausler et al. 2021) or multi-level aggregation (Wang et al. 2022) on features to boost performance, but we do not.

The quantitative results of our DHE-VPR compared with other methods are shown in Table 3. On all datasets, our DHE-VPR achieves the best Recall@5. And the average performance (R@1, R@5, and R@10) of our method on all datasets is also the best among all methods. CosPlace is the SOTA one-stage VPR method because it was trained on the very large-scale SF-XL dataset (Berton, Masone, and Caputo 2022), whereas our method was not trained on such a dataset. Although our model has no advantage when using only global features for retrieval, our complete DHE-VPR outperforms CosPlace thanks to geometric verification for re-ranking. Especially on Nordland, which is prone to perceptual aliasing, our method has a 20.6% absolute increase on R@1 than without re-ranking. TransVPR is an excellent two-stage VPR method, with the best R@1 on MSLS val and MSLS challenge. However, our method outperforms it in most results, achieving absolute improvements of 4.2% and 13.7% on R@5 on the MSLS challenge and Nordland, respectively. Besides, all methods use 640×480 resolution images except our method, which uses the 384×384 resolution. This enables our method to perform well on the low-resolution Nordland dataset as well. The qualitative results in Fig. 4 illustrate our method is highly robust against condition (e.g. light, season) and viewpoint changes, and less susceptible to perceptual aliasing than other methods.

Runtime Analysis

We evaluate the computational efficiency of the VPR systems using the runtime (feature extraction time and matching time) of a single query on the Pitts30k test dataset. We compare the proposed DHE-VPR with the other two-stage VPR methods. The results are shown in Table 4. Since we use the compact backbone (CCT) and the input image is only 384×384 resolution, our method has an advantage in feature extraction runtime. And the matching time of our method is only 0.098s, which is less than 1/30 of TransVPR and Patch-NetVLAD-p. These two methods use RANSAC for geometric verification. Compared with SP-SuperGlue,

Method	Pitts30k			MSLS val			MSLS challenge			Nordland test			St. Lucia			Average		
	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
NetVLAD	84.0	92.8	94.9	52.4	64.7	69.4	31.5	42.1	46.2	10.9	19.2	24.5	49.0	68.1	76.2	45.6	57.4	62.2
SFRS	**89.4**	94.7	95.9	69.2	80.3	83.1	41.6	52.0	56.3	22.0	35.0	41.8	75.4	86.2	90.2	59.5	69.6	73.5
CosPlace	88.4	94.5	95.7	82.8	89.7	92.0	61.4	72.0	76.6	64.1	83.2	**89.2**	94.9	97.5	98.7	78.3	87.4	90.4
GeM*	83.9	93.3	95.4	80.3	90.0	92.3	57.1	74.7	80.1	46.8	72.5	81.3	96.0	98.8	99.5	72.8	85.9	89.7
SP-SuperGlue	87.2	94.8	**96.4**	78.1	81.9	84.3	50.6	56.9	58.3	25.8	35.4	38.2	86.5	92.1	93.4	65.6	72.2	74.1
Patch-NetVLAD-s	87.5	94.5	96.0	77.8	84.3	86.5	48.1	59.4	62.3	44.2	57.5	62.7	90.2	93.6	95.0	69.6	77.9	80.5
Patch-NetVLAD-p	88.7	94.5	95.9	79.5	86.2	87.7	48.1	57.6	60.5	51.6	60.1	62.8	93.9	95.5	96.2	72.4	78.8	80.6
TransVPR	89.0	94.9	96.2	**86.8**	91.2	92.4	**63.9**	74.0	77.5	61.3	71.7	75.6	98.7	99.0	99.2	79.9	86.2	88.2
ETR-D	84.2	91.6	93.8	79.3	88.0	89.6	50.6	62.1	65.8	-	-	-	-	-	-	-	-	-
DHE-VPR (ours)	**89.4**	**95.1**	96.2	84.1	**91.4**	**93.1**	61.7	**78.2**	**82.6**	**67.4**	**85.4**	88.9	**99.1**	**99.6**	**99.7**	**80.3**	**89.9**	**92.1**

Table 3: Comparison to SOTA methods on benchmark datasets. The best is highlighted in bold and the second is underlined.

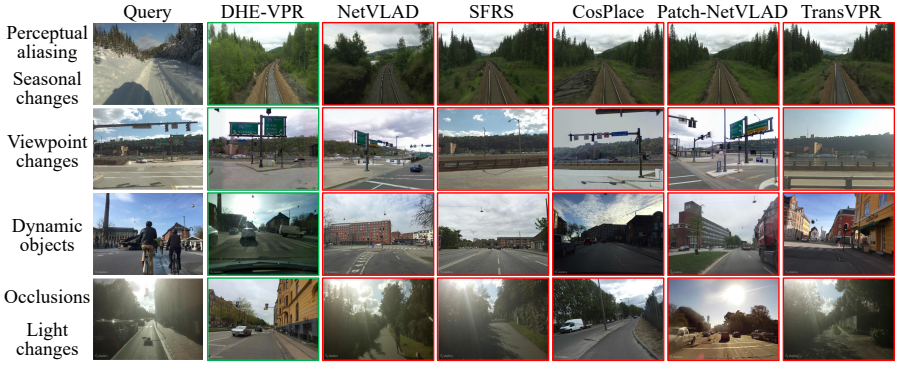

Figure 4: Qualitative results. In these challenging examples, our DHE-VPR gets the correct results, while all other methods yield false places. In the last example, the buildings on the left of the query image are occluded by vehicles and vegetation.

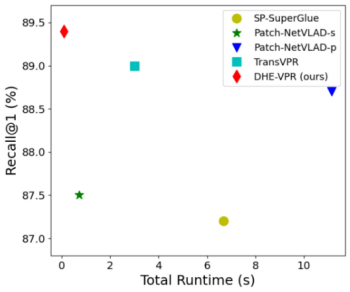

Figure 5: Recall@1-Runtime comparison of two-stage VPR methods on the Pitts30k dataset.

Method	Extraction Time (s)	Matching Time (s)	Total Time (s)
SP-SuperGlue	0.042	6.639	6.681
Patch-NetVLAD-s	0.186	0.551	0.737
Patch-NetVLAD-p	0.412	10.732	11.144
TransVPR	0.008	3.010	3.018
DHE-VPR (Re100)	0.006	0.264	0.270
DHE-VPR (ours)	**0.006**	**0.098**	**0.104**

Table 4: Runtime of two-stage methods on Pitts30k. DHE-VPR(Re100) uses our method to re-rank top-100 candidates.

which also uses neural networks to match local features (but not based on homography), our method takes less than 1/60 of its matching time. Although Patch-NetVLAD-s uses a fast verification algorithm (Rapid Spatial Scoring), it is still more time-consuming than ours. Considering that our method only re-ranks the top-32 candidates while others re-rank top-100, we also provide the runtime of ours to re-rank top-100 candidates for a more fair comparison, denoted as DHE-VPR(Re100). Its matching runtime and total runtime are still one order of magnitude lower than the RANSAC-based methods (TransVPR and Patch-NetVLAD-p). Fig. 5 simultaneously shows total runtime and R@1. Our method has obvious advantages in both performance and efficiency.

Conclusions

In this paper, we presented a novel hierarchical VPR architecture, which uses a DHE network to regress homography to check the geometric consistency in the re-ranking stage. It can break through the time-consuming and non-differentiable limitations of the RANSAC algorithm. Meanwhile, we proposed the REI loss to train the DHE network, which can be jointly optimized with the backbone thus making the learned feature maps more suitable for local matching in re-ranking. The experimental results showed that our architecture can outperform several SOTA methods on VPR benchmark datasets. And the runtime of ours is more than one order of magnitude lower than that of the existing two-stage methods using RANSAC for geometric verification.

Acknowledgments

This work was supported by the National Key R&D Program of China (2022YFB4701400/4701402), SSTIC Grant (JCYJ20190809172201639, WDZC20200820200655001), Shenzhen Key Laboratory (ZDSYS20210623092001004), the Project of Peng Cheng Laboratory (PCL2023A08), and Beijing Key Lab of Networked Multimedia.

References

Ali-bey, A.; et al. 2022. Gsv-cities: Toward appropriate supervised visual place recognition. *Neurocomputing*, 513: 194–203.

Angeli, A.; Filliat, D.; Doncieux, S.; and Meyer, J. 2008. Fast and Incremental Method for Loop-Closure Detection Using Bags of Visual Words. *IEEE Transactions on Robotics*, 24(5): 1027–1037.

Arandjelovic, R.; Gronat, P.; Torii, A.; Pajdla, T.; and Sivic, J. 2016. NetVLAD: CNN architecture for weakly supervised place recognition. In *CVPR*, 5297–5307.

Bay, H.; Ess, A.; Tuytelaars, T.; and Gool, L. V. 2008. Speeded-up robust features (SURF). *Computer vision and image understanding*, 110(3): 346–359.

Berton, G.; Masone, C.; and Caputo, B. 2022. Rethinking visual geo-localization for large-scale applications. In *CVPR*, 4878–4888.

Berton, G.; Masone, C.; Paolicelli, V.; and Caputo, B. 2021. Viewpoint invariant dense matching for visual geolocalization. In *ICCV*, 12169–12178.

Berton, G.; Mereu, R.; Trivigno, G.; Masone, C.; Csurka, G.; Sattler, T.; and Caputo, B. 2022. Deep visual geo-localization benchmark. In *CVPR*, 5396–5407.

Brachmann, E.; Krull, A.; Nowozin, S.; Shotton, J.; Michel, F.; Gumhold, S.; and Rother, C. 2017. Dsac-differentiable ransac for camera localization. In *CVPR*, 6684–6692.

Brachmann, E.; et al. 2019. Neural-guided RANSAC: Learning where to sample model hypotheses. In *ICCV*, 4322–4331.

Brachmann, E.; et al. 2021. Visual camera re-localization from RGB and RGB-D images using DSAC. *IEEE transactions on pattern analysis and machine intelligence*, 44(9): 5847–5865.

Camara, L. G.; et al. 2020. Highly Robust Visual Place Recognition Through Spatial Matching of CNN Features. In *ICRA*, 3748–3755.

Cao, B.; et al. 2020. Unifying deep local and global features for image search. In *ECCV*, 726–743. Springer.

Chen, Z.; Jacobson, A.; Sunderhauf, N.; Upcroft, B.; and Milford, M. 2017a. Deep learning features at scale for visual place recognition. In *ICRA*, 3223–3230.

Chen, Z.; Maffra, F.; Sa, I.; and Chli, M. 2017b. Only look once, mining distinctive landmarks from convnet for visual place recognition. In *IROS*, 9–16.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Li, F.-F. 2009. ImageNet: A large-scale hierarchical image database. In *CVPR*, 248–255.

DeTone, D.; et al. 2016. Deep image homography estimation. *arXiv preprint arXiv:1606.03798*.

DeTone, D.; et al. 2018. Superpoint: Self-supervised interest point detection and description. In *CVPR Workshops*, 224–236.

Doan, A.-D.; Latif, Y.; Chin, T.-J.; Liu, Y.; Do, T.-T.; and Reid, I. 2019. Scalable place recognition under appearance change for autonomous driving. In *ICCV*, 9319–9328.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *ICLR*.

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

Gao, P.; et al. 2020. Long-term loop closure detection through visual-spatial information preserving multi-order graph matching. In *AAAI*, volume 34, 10369–10376.

Garg, S.; Jacobson, A.; Kumar, S.; and Milford, M. 2017. Improving condition-and environment-invariant place recognition with semantic place categorization. In *IROS*, 6863–6870.

Garg, S.; and Milford, M. 2021. SeqNet: Learning descriptors for sequence-based hierarchical place recognition. *IEEE Robotics and Automation Letters*, 6(3): 4305–4312.

Garg, S.; et al. 2018. Don't look back: Robustifying place categorization for viewpoint-and condition-invariant place recognition. In *ICRA*, 3645–3652.

Ge, Y.; Wang, H.; Zhu, F.; Zhao, R.; and Li, H. 2020. Self-supervising fine-grained region similarities for large-scale image localization. In *ECCV*, 369–386. Springer.

Hansen, P.; et al. 2014. Visual place recognition using HMM sequence matching. In *IROS*, 4549–4555.

Hartley, R.; et al. 2003. *Multiple view geometry in computer vision*. Cambridge university press.

Hassani, A.; Walton, S.; Shah, N.; Abuduweili, A.; Li, J.; and Shi, H. 2021. Escaping the big data paradigm with compact transformers. *arXiv preprint arXiv:2104.05704*.

Hausler, S.; Garg, S.; Xu, M.; Milford, M.; and Fischer, T. 2021. Patch-netvlad: Multi-scale fusion of locally-global descriptors for place recognition. In *CVPR*, 14141–14152.

Hausler, S.; and Milford, M. 2020. Hierarchical multi-process fusion for visual place recognition. In *ICRA*, 3327–3333. IEEE.

Hou, Y.; et al. 2018. BoCNF: efficient image matching with Bag of ConvNet features for scalable and robust visual place recognition. *Autonomous Robots*, 42(6): 1169–1185.

Jin Kim, H.; et al. 2017. Learned contextual feature reweighting for image geo-localization. In *CVPR*, 2136–2145.

Jégou, H.; Douze, M.; Schmid, C.; and Pérez, P. 2010. Aggregating local descriptors into a compact image representation. In *CVPR*, 3304–3311.

Khaliq, A.; Ehsan, S.; Chen, Z.; Milford, M.; and Mcdonald-Maier, K. 2020. A Holistic Visual Place Recognition Approach Using Lightweight CNNs for Significant ViewPoint and Appearance Changes. *IEEE Transactions on Robotics*, 36(2): 561–569.

Koguciuk, D.; et al. 2021. Perceptual loss for robust unsupervised homography estimation. In *CVPR*, 4274–4283.

Le, H.; Liu, F.; Zhang, S.; and Agarwala, A. 2020. Deep homography estimation for dynamic scenes. In *CVPR*, 7652–7661.

Liu, D.; Cui, Y.; Yan, L.; Mousas, C.; Yang, B.; and Chen, Y. 2021. Densernet: Weakly supervised visual localization using multi-scale feature aggregation. In *AAAI*, volume 35, 6101–6109.

Liu, L.; Li, H.; and Dai, Y. 2019. Stochastic attraction-repulsion embedding for large scale image localization. In *ICCV*, 2570–2579.

Lowry, S.; and Andreasson, H. 2018. Lightweight, Viewpoint-Invariant Visual Place Recognition in Changing Environments. *IEEE Robotics and Automation Letters*, 3(2): 957–964.

Lowry, S.; Sünderhauf, N.; Newman, P.; Leonard, J. J.; Cox, D.; Corke, P.; and Milford, M. J. 2016. Visual place recognition: A survey. *IEEE Transactions on Robotics*, 32(1): 1–19.

Lu, F.; Chen, B.; Zhou, X.-D.; and Song, D. 2021. STA-VPR: Spatio-temporal alignment for visual place recognition. *IEEE Robotics and Automation Letters*, 6(3): 4297–4304.

Lu, F.; Zhang, L.; Dong, S.; Chen, B.; and Yuan, C. 2023. AANet: Aggregation and Alignment Network with Semi-hard Positive Sample Mining for Hierarchical Place Recognition. In *ICRA*, 11771–11778. IEEE.

Milford, M. J.; and Wyeth, G. F. 2012. SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights. In *ICRA*, 1643–1649. IEEE.

Naseer, T.; Oliveira, G. L.; Brox, T.; and Burgard, W. 2017. Semantics-aware visual localization under challenging perceptual conditions. In *ICRA*, 2614–2620.

Naseer, T.; Spinello, L.; Burgard, W.; and Stachniss, C. 2014. Robust visual robot localization across seasons using network flows. In *AAAI*, volume 28.

Nguyen, T.; Chen, S. W.; Shivakumar, S. S.; Taylor, C. J.; and Kumar, V. 2018. Unsupervised deep homography: A fast and robust homography estimation model. *IEEE Robotics and Automation Letters*, 3(3): 2346–2353.

Olid, D.; et al. 2018. Single-view place recognition under seasonal changes. *arXiv preprint arXiv:1808.06516*.

Peng, G.; Zhang, J.; Li, H.; and Wang, D. 2021. Attentional pyramid pooling of salient visual residuals for place recognition. In *ICCV*, 885–894.

Radenović, F.; Tolias, G.; and Chum, O. 2018. Fine-tuning CNN image retrieval with no human annotation. *IEEE transactions on pattern analysis and machine intelligence*, 41(7): 1655–1668.

Sarlin, P.-E.; DeTone, D.; Malisiewicz, T.; and Rabinovich, A. 2020. Superglue: Learning feature matching with graph neural networks. In *CVPR*, 4938–4947.

Sünderhauf, N.; Shirazi, S.; Jacobson, A.; Pepperell, E.; and Milford, M. 2015. Place Recognition With ConvNet Landmarks: Viewpoint-Robust, Condition-Robust, Training-Free. In *RSS*.

Torii, A.; Sivic, J.; Pajdla, T.; and Okutomi, M. 2013. Visual place recognition with repetitive structures. In *CVPR*, 883–890.

Wang, R.; Shen, Y.; Zuo, W.; Zhou, S.; and Zheng, N. 2022. TransVPR: Transformer-based place recognition with multi-level attention aggregation. In *CVPR*, 13648–13657.

Warburg, F.; Hauberg, S.; Lopez-Antequera, M.; Gargallo, P.; Kuang, Y.; and Civera, J. 2020. Mapillary street-level sequences: A dataset for lifelong place recognition. In *CVPR*, 2626–2635.

Xin, Z.; Cai, Y.; Lu, T.; Xing, X.; Cai, S.; Zhang, J.; Yang, Y.; and Wang, Y. 2019. Localizing Discriminative Visual Landmarks for Place Recognition. In *ICRA*, 5979–5985.

Yin, P.; Xu, L.; Li, X.; Yin, C.; Li, Y.; Srivatsan, R. A.; Li, L.; Ji, J.; and He, Y. 2019. A multi-domain feature learning method for visual place recognition. In *ICRA*, 319–324.

Yu, J.; Zhu, C.; Zhang, J.; Huang, Q.; and Tao, D. 2019. Spatial pyramid-enhanced NetVLAD with weighted triplet loss for place recognition. *IEEE transactions on neural networks and learning systems*, 31(2): 661–674.

Zhang, H.; Chen, X.; Jing, H.; Zheng, Y.; Wu, Y.; and Jin, C. 2023. ETR: An Efficient Transformer for Re-ranking in Visual Place Recognition. In *WACV*, 5665–5674.

Zhang, J.; Wang, C.; Liu, S.; Jia, L.; Ye, N.; Wang, J.; Zhou, J.; and Sun, J. 2020. Content-aware unsupervised deep homography estimation. In *ECCV 2020*, 653–669. Springer.