MGGA: Make GeM Great Again via Regularization Branch to Mitigate Channel Vanishing in Visual Place Recognition *

Qǐxī Zhào*
Independent Researcher
Shenyang, China
qixiii1219@outlook.com

Jiwei Nie*
State Key Laboratory of Massive Personalized Customization System and Technology
No.1 Haier Road, Laoshan District, Qingdao, China
niejiwei@haier.com

Zuotao Ning
Independent Researcher
Shenyang, China
ningzuotao@ed-alumni.net

Joe-Mei Feng✉
Tamkang University
No.151, Yingzhuan Rd., Tamsui Dist.,
New Taipei City 251301, Taiwan
168984@o365.tku.edu.tw

Abstract

Deep-learning-based methods have achieved significant success in the Visual Place Recognition (VPR) task, which is important for autonomous driving and robotics systems. Recent advancements primarily focus on the sophisticated feature aggregation module. This paper argues for a shift in emphasis toward the backbone features. Through an in-depth analysis of GeM, one of the simplest pooling aggregator based VPR method, we identify a prevalent issue, termed 'Channel vanishing'. The issue manifests as a substantial proportion of channels in both the final GeM descriptor and the backbone output local features turning zero-valued and inactive during training, thereby drastically diminishing the representational capacity of the model and undermining its VPR performance. In order to solve this problem, we propose a regularization branch with a fully connected layer for the GeM pipeline. This branch successfully mitigates Channel vanishing and further enriches the diversity and representation of the backbone output features. During inference, our streamlined model, using only the GeM aggregator, achieves state-of-the-art performance among backbones that are not transformer-based. Notably, when utilizing the DINOv2-B backbone, our method derives 99.1% recall@1 and 100% recall@5 VPR scores on the Tokyo24/7 dataset. This result suggests that strengthening backbone features can substantially narrow the gap between simple GeM pooling and more complex aggregators; assessing how broadly this observation transfers to other aggregators is an interesting direction.

***Keywords:** Deep-learning, Visual Place Recognition, Autonomous Navigation, Robotics, GeM, Channel vanishing*

1. Introduction

Visual Place Recognition (VPR) has emerged as a cornerstone in the field of robotics and autonomous systems. It endows robots and autonomous vehicles with the ability to navigate and interpret their environments with remarkable precision, akin to human place recognition capabilities. VPR operates by extracting features from visual data and matching them against a database of known locations, thus determining the system's current position. However, VPR faces considerable challenges. It must contend with constantly changing conditions, such as shifts in illumination, weather, and seasons, which can drastically alter the appearance of scenes. This requires robust and adaptable algorithms capable of consistent place recognition despite these variations.

In recent years, Deep Neural Networks (DNNs) have demonstrated superior performance in VPR tasks [3, 18, 4, 2, 10]. DNN-based VPR approaches typically com-

*Contribute Equally

prise a backbone for feature extraction and an aggregator for synthesizing these features into discriminative descriptors. Current research predominantly concentrates on devising intricate aggregators, such as MixVPR [2], GeM [18], CosPlace [4], NetVLAD [3], et al. As reported in MixVPR, under the premise of using the same backbone network, although MixVPR demonstrated dominant performance, NetVLAD and CosPlace also showed relatively advanced performance, whereas GeM's performance lagged significantly behind the compared methods. Among these methods, the backbone network occupies most of the parameters of the entire VPR network, with MixVPR accounting for 90%, NetVLAD for 99%, and GeM nearly 100%. Moreover, the versatility of the backbone network has been proven in various vision tasks, reflecting its potent feature extraction capacity. We have reason to believe that GeM, using the same backbone and a minimalistic aggregator design, also has the potential to achieve performance comparable to other methods, although it may be difficult to surpass them.

Therefore, in this paper, we studied GeM to investigate the reasons for its extremely poor performance. During the numerical statistical analysis of its final descriptors, we discovered a critical yet under-discussed issue, as illustrated in Fig. 1. Specifically, during training, numerous channels in the final descriptors vanish to zero. This makes GeM significantly hinder the model's ability to use only a small fraction of descriptor channels to learn varied features, consequently impairing its overall efficacy. Moreover, our further investigation revealed that the same phenomenon appeared in the local features output by the backbone after training. We define this issue as 'Channel Vanishing'. We believe this is the reason for GeM's poor performance. We trace this issue to the use of the ReLU activation function preceding the GeM pooling aggregator, which confines the GeM descriptor to the non-negative section of the high-dimensional hypersphere. As training progresses, descriptors of distinct scenes diverge from each other, causing both the GeM descriptor and local features to converge on the non-negative section's border, where most dimensions are zero. This will be elaborated further in Section 3.

To address this challenge, we propose an innovative training pipeline for GeM, integrating a regularization branch. Experimental results demonstrate that this branch prevents the local features from being constrained to the border during training, thereby rectifying the Channel Vanishing issue in GeM descriptors. As a result, the 'liberated' local features exhibit increased diversity, enhancing the GeM model's performance. Moreover, through experiments, we found that this branch does not contribute highly distinctive information to the final global descriptor during inference. Therefore, despite considering the inconsistency between the training and inference pipelines, we still use only the pure GeM branch for inference to achieve higher computational efficiency. Furthermore, we attribute the role of this regularization branch to assisting the backbone in training highly significant features, enabling the pure GeM to aggregate and produce high-performance descriptors.

Empirical evidence confirms that incorporating the regularization branch into the training regimen significantly improves the pure GeM descriptor's performance, achieving competitive results on various public benchmarks, even with a relatively low descriptor dimension. This performance not only underscores the efficacy of our method but also its capacity for information consolidation and compression, offering viable solutions for memory-constrained applications. Further, leveraging the robust DINOv2-B backbone [14], our approach achieves recall@1 (**99.1%**) and recall@5 (**100%**) VPR scores on the Tokyo24/7 datasets. The exceptional performance of our model suggests strong robustness in complex urban environments. Therefore, we conclude that our findings suggest a shift in emphasis from the aggregator to the backbone in VPR models.

2. Related Works

VPR techniques can be broadly categorized into one-stage and two-stage methods.

One-stage VPR methods generate a global descriptor for each scene image, enabling image comparisons through descriptor similarity. The emergence of Convolutional Neural Networks (CNN) has promoted the development of VPR techniques. Some VLAD-centric methods [3, 17, 16, 7, 13, 12, 26] learn a trainable VLAD layer to softly cluster and aggregate local features with modification including multi-resolution images [12], pyramidal feature maps [26], attention mechanisms [17], and novel loss functions [13, 7]. However, these methods pose storage challenges because of their high-dimensional descriptors. In contrast, simpler pooling-based method GeM [18] utilizes generalized-mean pooling for lower-dimensional descriptor embedding. However, GeM is quickly surpassed by methods such as MixVPR [2], which leverages spatial fully connected layers for feature embedding, and CosPlace [4] and EigenPlaces [5], which share the same model structure and both trained on the SF-XL dataset [4] but with distinct training strategies.

Except for GeM[18], all of the above methods focus on developing an effective aggregator or novelty training strategies to address the VPR problem. Nevertheless, this paper is dedicated to diversifying the local features derived from the backbone, demonstrating that even with the simple GeM, exceptional performance can still be achieved.

Two-stage VPR approaches [8, 19, 23, 27, 6, 28, 15], which characterize by an initial coarse retrieval phase followed by a re-ranking process using local features. Patch-NetVLAD [8] extends the re-ranking stage for NetVLAD

[3]. CAHIR [15] further develops the APPSVR [17] by incorporating the re-ranking stage. ETR [27] leverages attention mechanisms with the Superpoint [19] or DELG[6] features, achieving remarkable performance. Although this two-stage strategy significantly enhances accuracy and robustness, especially in environments with diverse and challenging conditions, the re-ranking stage entails substantial computational overhead due to extensive local feature matching.

Latest developments utilizing Vision transformers have exhibited exceptional efficacy in this area. For instance, Salad [10], reformulates the clustering process of NetVLAD [3] as an optimal transport problem, leveraging the robust DINOv2 backbone [14] to set new benchmarks in VPR performance. TransVPR [23] and R^{2}former [28] are transformer-based VPR method with outstanding performance, with R^{2}former proposing a comprehensive training pipeline for the global retrieval and re-ranking stages. Pair-VPR [9] introduces a place-aware pre-training strategy combined with a contrastive pair classification framework, achieving state-of-the-art results on benchmarks such as Tokyo24/7 (100% Recall@1 with ViT-G [9]), Pitts30k (95.4% Recall@1), MSLS-Val (95.4% Recall@1), and MSLS-Challenge (81.7% Recall@1), outperforming prior methods like SALAD and EigenPlaces by significant margins, particularly with larger encoders. Similarly, EffoVPR [22] harnesses foundation models for efficient, low-dimensional zero-shot or single-stage retrieval, yielding impressive metrics including 97.5% Recall@1 on Tokyo24/7 (1024-dim), 94.8% on Pitts30k, 90.9% on MSLS-Val, and 78.2% on MSLS-Challenge, while maintaining competitiveness in compact representations (e.g., 94.6% Recall@1 at 128-dim on Tokyo24/7). These Transformer-based approaches underscore the potential of large-scale pre-trained models to enhance VPR robustness, but these methods significantly increase the computational demands.

Therefore, we aim to improve the performance of the one-stage VPR method, GeM, as much as possible to circumvent the storage and computation issues.

3. Channel Vanishing

3.1. Phenomenon of Channel Vanishing

Before describing the phenomenon, here we briefly introduce the GeM-pooling function as:

$$\mathcal{D}_{\text{GeM}} = \mathcal{N}\left(\left(\frac{\sum_{i=1}^{H\times W}(\mathcal{D}_{\text{local}_i})^p}{H\times W}\right)^{\frac{1}{p}}\right). \tag{1}$$

In this equation, $\mathcal{D}_{\text{local}} = \{\mathcal{D}_{\text{local}_i} \in R^{1\times D}\}_{i=1}^{H\times W}$ represent the the local features which are derived by sequentializing the feature map $F \in R^{D\times H\times W}$ after the ReLU function, where D is the feature dimension, H and W are the height and width of F, p is a learn-able exponent that modulates the pooling behavior, and $\mathcal{N}(\cdot)$ is the L2-normalizing operation. It is noteworthy that the ReLU function before the GeM pooling layer is crucial because p and $\frac{1}{p}$ change as floating-point numbers that cannot be applied to negative values.

When training the GeM model [1], we notice an interesting phenomenon: in $\mathcal{D}_{\text{GeM}}$, a notable portion of the embedding channels are with zero value. For convenience, we denote these channels as 'zero-valued channels'. For a deeper investigation, we track the number of these channels throughout the entire training process, as shown in Fig. 1. We see that after a few epochs, the zero-valued channel number exhibits a fluctuating increase, then remains stable at around 87.5% of the total channel number. Simultaneously, we can observe the same pattern on $\mathcal{D}_{\text{local}}$ for $\mathcal{D}_{\text{GeM}}$. Consequently, after training, nearly 90% channel in both $\mathcal{D}_{\text{local}}$ and $\mathcal{D}_{\text{GeM}}$ become non-contributory, severely weakening the representation ability of the model. We refer to this phenomenon as 'Channel Vanishing', and present a comprehensive analysis to explain the cause in the following subsection.

3.2. Analysis on Channel Vanishing

In this section, we analyse the phenomenon of Channel vanishing by illustrating the movement of $\mathcal{D}_{\text{GeM}}$ in feature space during optimization, as shown in the Fig. 2. In this figure, $\mathcal{D}_{\text{GeM}}$ are represented as 'o' in various colors, symbolizing descriptors from the same scene (identical color) or different scenes (distinct colors). For demostration, the high-dimensional space is projected onto a 3-dimensional space. Next, we begin with the introduction of the figure elements in the Fig. 2.

Initially, it is important to note that all $\mathcal{D}_{\text{GeM}}$ are located in the non-negative quadrant of the 3-dimensional spherical space. This positioning is due to two factors:

- The activation of a ReLU function prior to the GeM pooling layer ensures that each channel value of the local features, $\mathcal{C}_{\text{local}}$ is non-negative. Consequently, based on Eq. 1, the channel values in $\mathcal{D}_{\text{GeM}}$, are also non-negative, restricting them to the non-negative sector of the sphere.

- The L2-normalizing function confines $\mathcal{D}_{\text{GeM}}$ to the surface of the 3-dimensional sphere.

Furthermore, the arrows in the Fig. 2 symbolize the forces generated by optimizing the loss function. Here, we use multi-similarity loss $\mathcal{L}_{ms}$ as a demonstration for illustrating these forces. The loss function is formulated as Eq. 2.

$$\mathcal{L}_{ms} = \frac{1}{m}\sum_{i=1}^{m}\left\{\frac{1}{\alpha}\log\left[1+\sum_{k\in P_i}e^{-\alpha(S_{ik}-\lambda)}\right]+\frac{1}{\beta}\log\left[1+\sum_{k\in N_i}e^{\beta(S_{ik}-\lambda)}\right]\right\} \tag{2}$$

[1] Training details can be found in the supplementary material.

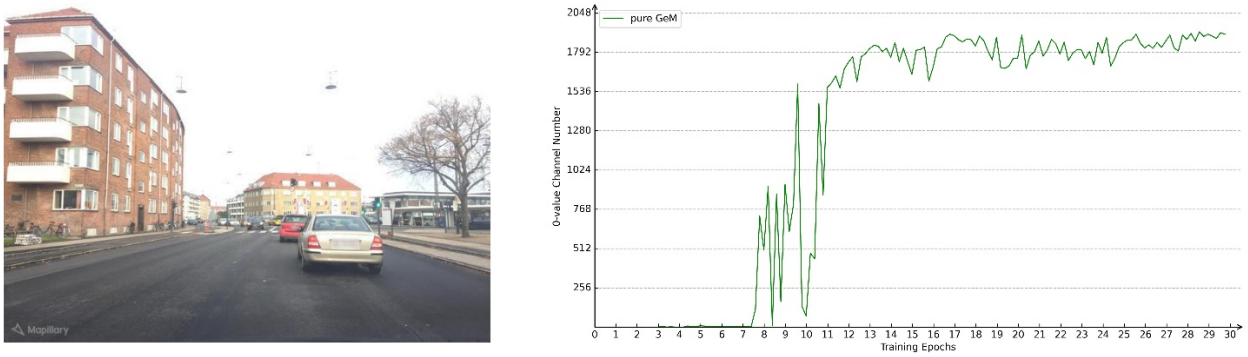


Figure 1. Evolution of zero-valued channel number of the final descriptor when training on pure GeM with ResNet50 on GSV-Cities [1]. We select the left image from the MSLS dataset as a qualitative reference. The right plot shows the training process within 30 epochs. It can be observed that the zero-valued channel first occurs at the 7^{th} epoch. Then after a severe fluctuation, beginning at around the 13^{th} epoch, the number of those channels reaches and stabilizes at 87.5% of the total channel number 2048.

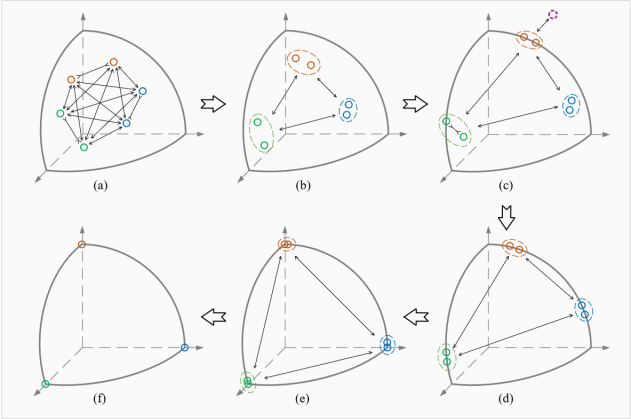


Figure 2. The formation process of Channel vanishing.

where m is the number of training samples per batch, P_i and N_i are the mined positive and negative candidates for the i^{th} anchor image, and k is the indices of the candidate image, $[\alpha, \beta, \lambda]$ are three hyper-parameters, and S represents the similarity between two image descriptors. The convergence of $\mathcal{L}_{ms}$ hinges on the reduction of two factors F_1 and F_2:

$$F_1 = \sum_{k \in P_i} e^{-\alpha(S_{ik}-\lambda)}, \qquad F_2 = \sum_{k \in N_i} e^{\beta(S_{ik}-\lambda)} \tag{3}$$

Given that α and β are positive, F_1 is monotonically decreasing, and F_2 is monotonically increasing. To minimize $\mathcal{L}_{ms}$, particularly when S represents cosine similarity, $\mathcal{L}_{ms}$ continually encourages positive pairs to become more alike $[(S_{ik}|k \in P_i) \rightarrow 1]$ and negative pairs more distinct by decreasing their similarity $[(S_{ik}|k \in N_i) \downarrow]$. Under the ReLU-induced non-negativity, the cosine similarity satisfies $S_{ik} \in [0, 1]$, so the practical lower bound is 0 (orthogonality). This process is driven by two types of forces: 1) Pulling the positive pair closer. 2) Pushing the negative pair further away. The first force narrows the distribution of descriptors within the same scene, whereas the second force broadens the distribution of descriptors across different scenes.

Subsequent to detailing the elements in the Fig. 2, we illustrate the formation process of Channel vanishing. Initially, $\mathcal{D}_{\text{GeM}}$ are dispersed as depicted in Fig. 2(a), influenced by attractive and repulsive forces among the same and different scenes $\mathcal{D}_{\text{GeM}}$, respectively. After a few training iterations, the same-scene $\mathcal{D}_{\text{GeM}}$ are gradually grouped together as shown in Fig. 2(b). In this phase, the repulsive forces between different-scene $\mathcal{D}_{\text{GeM}}$ intensify, causing the overall distribution to become more disparate. As the distribution widens, some $\mathcal{D}_{\text{GeM}}$ inevitably touch the borders of the non-negative section of the sphere where at least one channel equals zero, as shown in Fig. 2(c). It can be seen that one $\mathcal{D}_{\text{GeM}}$ in the green group reaches the border. Although individual $\mathcal{D}_{\text{GeM}}$ might temporarily escape this border, over time, entire groups tend to converge at this boundary, exemplified by the red group in Fig. 2(c). This entrapment is a consequence of the ReLU function's inhibition of negative values, thereby nullifying any counterforce that could displace $\mathcal{D}_{\text{GeM}}$ back into the sphere's interior, such as the hypothetical purple dashed 'o' outside the non-negative sector. We consider Fig. 2(c) illustrates the reason of the fluctuating zero-valued channel number in Fig. 1 between the 7^{th} and 11^{th} epochs. As more $\mathcal{D}_{\text{GeM}}$ accumulate at the border, they exert mutual forces, steering themselves towards the areas with a higher density of zero channels, as shown in Fig. 2(d) and (e). Eventually, all $\mathcal{D}_{\text{GeM}}$ converge to the state depicted in Fig. 2(f) with most of the channel values equaling zero, with the majority of channel values being zero and only a minor fraction remaining non-zero to maintain force equilibrium. This phenomenon

is known as Channel Vanishing. Moreover, according to the Eq. 1, when this phenomenon occurs and most channels of $\mathcal{D}_{\text{GeM}} = 0$, the corresponding channels of $\mathcal{D}_{\text{local}}$ must also be zero, highlighting the occurrence of Channel Vanishing at the local feature level as well. In the experiments employing different loss functions (triplet loss and contrastive loss), the Channel Vanishing issue persists.

3.3. Method

3.3.1 Early Convergence.

In addressing the Channel Vanishing issue inherent in the GeM method, we initially propose to reduce the learning-rate before Channel Vanishing to lead an early convergence of the model. This approach effectively mitigates the Channel Vanishing problem. Connected to the analyse in Sec. 3.2, we consider that the model is converged prematurely at a local optimum, at which most of the descriptors have not been reaching the restricted border. However, the VPR scores are still undesirable, as reported in Tab. 1.

3.3.2 Regularization Branch

From the above experimental results, in this part, we shift the researching focus to the approach of enabling the descriptors to escape when they reach the restricted borders. Meanwhile, we hope to maintain the simplicity of the GeM model structure, ensuring its potential for widespread application. So, we propose a regularization branch to prevent the local features from Channel vanishing. The regularization branch runs parallel to the origin GeM branch and connects after the last ReLU function, as shown in Fig. 3. In this branch, the local features $\mathcal{D}_{\text{local}}$ are firstly projected to $\mathcal{D}_{\text{proj}}$ using a Fully Connected (FC) layer:

$$\mathcal{D}_{\text{proj}_i} = M \times \mathcal{D}_{\text{local}_i} \tag{4}$$

where M denotes the trainable weight matrix of the FC layer (with matching input/output dimensionality to $\mathcal{D}_{\text{local}}$), and i represents the index of the feature. Subsequently, the projected features $\mathcal{D}_{\text{proj}}$ are simply summarized and L2-normalized then fused with the GeM descriptor to yield the final descriptor $\mathcal{D}_{\text{final}}$, as Eq. 5.

$$\mathcal{D}_{\text{final}} = \mathcal{N}\left[\mathcal{N}\left(\sum_{i=1}^{H\times W} \mathcal{D}_{\text{proj}_i}\right) + \mathcal{D}_{\text{GeM}}\right]. \tag{5}$$

During inference, we remove the regularization branch to simplify the model and minimize computational overhead. Similar to standard GeM, our approach incorporates Principal Component Analysis Whitening (PCA-W) for dimension reduction, aiming for more compact descriptors and storage efficiency. The working mechanism of this design is described as Following.

Before adding the regularization branch, in the training process, $\mathcal{D}_{\text{local}}$ are only influenced by $\mathcal{D}_{\text{GeM}}$. Thus, when $\mathcal{D}_{\text{GeM}}$ are trapped at the borders, $\mathcal{D}_{\text{local}}$ would inevitably suffer from Channel vanishing. However, after integrating the branch, both $\mathcal{D}_{\text{GeM}}$ and $\mathcal{D}_{\text{local}}$ are affected by $\mathcal{D}_{\text{proj}}$. Eq. 4 tells that because M, $\mathcal{D}_{\text{proj}}$ move freely in the entire real-number space without a border. So, during the back-propagation optimizing process , $\mathcal{D}_{\text{proj}}$ can offer new converging forces with different directions from Channel vanishing. For this reason, the regularization branch can correct the local features from the Channel Vanishing state. Since $\mathcal{D}_{\text{GeM}}$ are aggregated from $\mathcal{D}_{\text{local}}$ by Eq. 1, our branch also gives $\mathcal{D}_{\text{GeM}}$ an escaping force, because correcting $\mathcal{D}_{\text{local}}$ would also rescue $\mathcal{D}_{\text{GeM}}$ from Channel Vanishing. Moreover, the other function of the branch is demonstrated in Fig. 4.

3.3.3 Visualization on Local Features.

In this subsection, we underscore the benefits of our regularization branch through a visual analysis of $\mathcal{D}_{\text{local}}$, as shown in Fig. 5. Firstly, since the exponent p of the GeM-pooling layer emphasizing the $\mathcal{D}_{\text{local}}$ with higher norm-value, these features prominently influence the $\mathcal{D}_{\text{GeM}}$. The norm values of the local features are calculated and demonstrated by the superimposed heat-maps, as shown in the upper portion of each sub-figure in the Fig. 5. Moreover, we aim to discern the difference of the local features distribution training with and without the proposed regularization branch. Nonetheless, directly observing high-dimensional (high-dim) features poses a challenge. To solve this, we employ the PCA operation and use the foremost three principal components to represent the features in a three-dimensional (3D) space, demonstrated by 3D distribution-maps at the lower portion of each sub-figure in the Fig. 5. Moreover, we paint the thermal colors on the 3D scatter points in order to better clarify the corresponding relationship of the local features between the distribution-maps and the heat-maps.

From the Fig. 5(a), it's evident that on pure GeM, the Channel Vanishing causes numerous local features converging tightly in a dense group, rendering the discrimination of $\mathcal{D}_{\text{GeM}}$ dominated by few outlier local features. Although, from the heat-map we can find that these outliers are with relative higher norm value and fell on the task-relevant elements, the loss of distinctive representation toward different scene elements results the ineffective for the discrimination of $\mathcal{D}_{\text{GeM}}$. Conversely, as Fig. 5(b) demonstrates, integrating our regularization branch results in a more scattered feature distribution, resembling a cone. The features closer to the cone tip have smaller norm values and extensive longer-norm features spread sparsely. The conical shape is the PCA projection of the high-dim features, which are located in the non-negative space and form a high-dim cone.

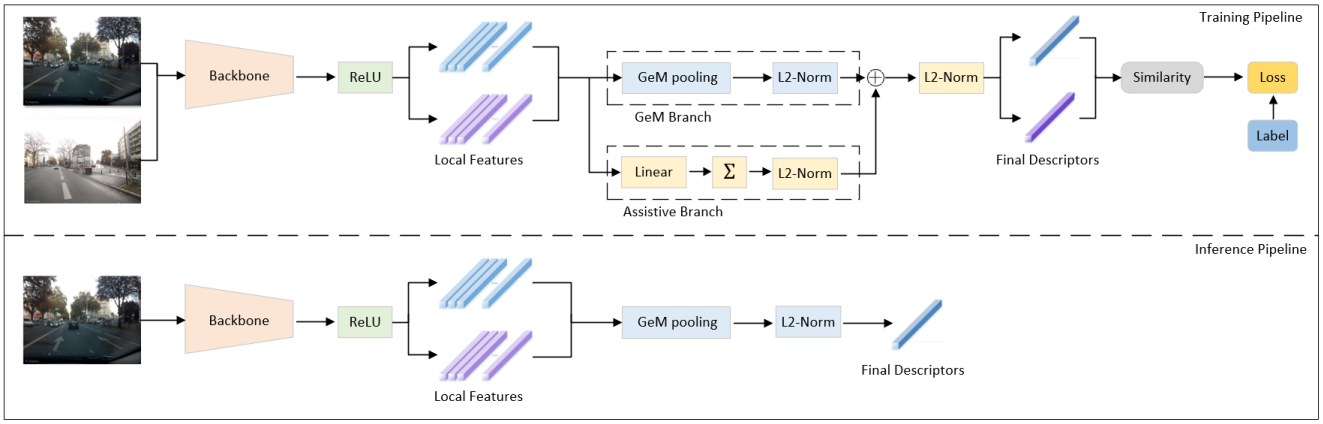


Figure 3. The training and inference pipeline of our proposed method

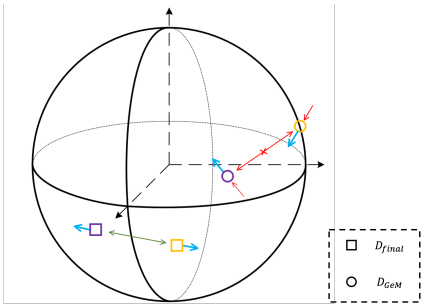


Figure 4. Forces and potential moving directions of the descriptors. The ball is a simplified demonstration of the entire high-dimensional hyper-sphere. The square and circle symbols indicate the final descriptor $\mathcal{D}_{\text{final}}$s and the GeM descriptor $\mathcal{D}_{\text{GeM}}$s, on which the different colors represent different-scene images. The black arrows point out the force conditions and the blue arrows show the potential moving directions. Here, it is not $\mathcal{D}_{\text{GeM}}$ but $\mathcal{D}_{\text{final}}$ that directly receive the repulsive force between different-scene descriptors. And with the new force given by the regularization branch, $\mathcal{D}_{\text{GeM}}$ are now able to leave the Channel Vanishing state.

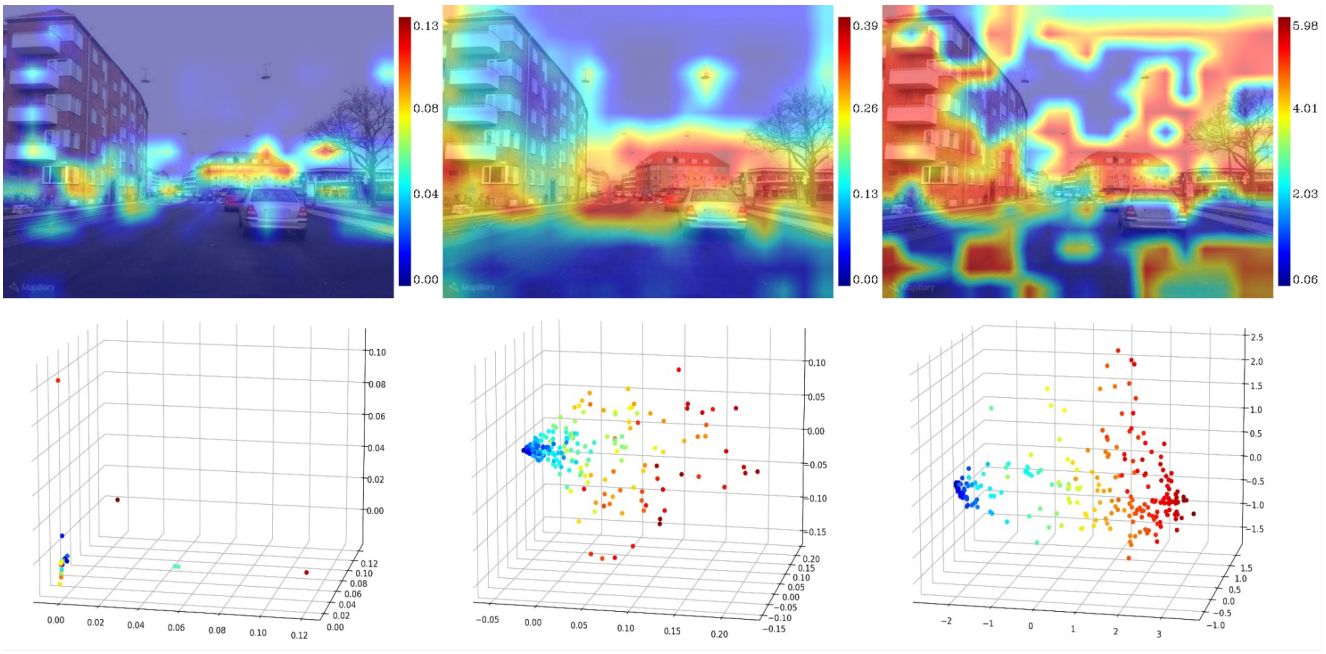
With gem pooling layers, such a distribution ensures that the dispersed, longer-norm features significantly influence $\mathcal{D}_{\text{GeM}}$. Therefore, from the heat-map, we can also find that these feature are typically align with the recognizable VPR task-relevant objects [17], such as buildings and streetlights. While the shorter-norm features are more concentrated, and mostly correspond to the unrecognizable elements, such as sky and roads. This phenomenon accentuates informative elements in $\mathcal{D}_{\text{GeM}}$, thereby enhancing our method's overall descriptiveness. Moreover, The effectiveness of the regularization branch is further confirmed as it offers new converging directions to $\mathcal{D}_{\text{local}}$, not only averting Channel Vanishing, but also fostering more distinctive learning for the VPR task. In addition, Fig. 5(c) also visualizes local features of early-converging GeM, where we indeed see a success in avoiding Channel vanishing. However, the heat-map reveals the non-optimality of the convergence and exposes its inability to effectively distinguish the important objects. The enhanced representation of local features suggests that inferencing with solely the GeM aggregator could suffice to achieve satisfactory performance.

4. Experiment

We train[2] our method on the GSV-Cities dataset [1], a comprehensive collection of urban street-view images, and employ five renowned benchmarks for evaluation: Pitts-30k and Pitts-250k [21], Tokyo24/7 [20], MSLS-val and MSLS challenge [25]. Note again that we remove the regularization branch for inference, using only the pure GeM model structure. Adhering to the VPR community standards [3, 17, 4, 2, 28], we utilize a 25-meter threshold to categorize positive and negative matches, and our primary evaluation metrics are the recall rates at the top 1, 5, and 10 ranks (R@1, R@5, and R@10). Because we agree that these metrics can effectively quantify the ability of a VPR method to correctly identify a location from a set of candidates, reflecting both accuracy and reliability in real-world applications.

Firstly, we compare our method against the Channel Vanishing and early-converging versions of GeM, with the same backbone ResNet50 [11].The statistics are reported in Tab. 1. These VPR scores clearly exhibit the advantage brought by the regularization branch. Especially on Tokyo24/7, our method surpasses the Channel vanishing version of GeM by 37.1% on the R@1 performance (all the '%' in the comparisons represent the absolute percentage point), which strongly proves the effectiveness of our design. Although the early-converging GeM also shows a certain performance enhancement, its VPR scores are notably lower than ours. Therefore, we confirm it is the Channel vanishing issue that results in the poor VPR scores of GeM,

[2]Implementation details can be found in the supplementary material.



(a) (b) (c)

Figure 5. Visualization on local features. The three groups from left to right correspond to three versions of GeM: pure GeM, ours, and early-converging GeM. All the three version are with ResNet50 and trained on GSV-Cities. The upper row shows the heat-maps of the norm value, and the color bar indicates the numerical level. The lower row exhibits the feature distribution in the 3D space, where the colors are aligned with those on the heat-maps. The original image can be found in Fig. 1, and examples on more images are provided in the supplementary material.

Table 1. Performance of the three versions of GeM with the backbone ResNet50.

2*Methods	Pitts250k			Tokyo24/7			MSLS-val			MSLS challenge		
	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
GeM (Channel Collapse)	81.1	91.4	93.4	50.8	67.3	74.3	76.6	86.1	88.5	-	-	-
GeM (Early Converging)	86.5	94.2	96.0	59.7	72.7	78.7	83.4	89.7	91.8	55.6	69.4	74.9
GeM (Assitive Branch)	**92.4**	**97.5**	**98.3**	**87.9**	**92.7**	**94.3**	**89.1**	**93.5**	**95.7**	**63.9**	**76.4**	**80.0**

because after solving this issue, no matter with the branch or early-converging strategy, we all see an obvious performance improvement. In addition, compared with the early-converging version, our more rational local feature representation does bring a stronger VPR capability for the final descriptors of GeM.

Furthermore, in order to more thoroughly evaluate our method, we compare it with the recent state-of-the-art VPR technologies, including both the one-stage and two-stage methods.

4.1. Comparison with One-stage Methods

For the one-stage methods, we select the recent benchmark methods for comparison, namely EigenPlaces [5], MixVPR [2], Salad [10], and our baseline GeM [18]. Firstly, we compare our model with these methods with the same backbone ResNet50 [11]. Secondly, we extend the experiments on the models with the backbone DINOv2-B [14]. This decision is informed by SALAD [10], which verifies that fine-tuning the potent backbone DINOv2 can significantly enhance the performance of the VPR methods. The experimental results are shown in Tab. 2. The public-available codes of above methods enable us to re-train the models on the same dataset with an identical hardware, ensuring a fair comparison. All the methods are trained on GSV-Cities [1] with Multi-Similarity-Loss [24], which have been proven as a powerful combination for the VPR task [2, 10]. The model architectures of the methods adhere to the corresponding papers [5, 2, 18, 10].

4.1.1 With ResNet50

In the comparisons outlined in the upper section of Table 2, our model with a 1024-dim descriptor outperforms all other methods in average VPR scores across the four datasets, with 84.1% R@1 score, 1.4% higher than the second-best MixVPR. When further reducing the descriptor dimension

Table 2. Comparison of our method with the one-stage SoTAs. Note that here the GeM model uses the early-converging version.

2*Methods	2*Backbone	2*Dim.	Pitts250k			Tokyo24/7			MSLS-val			MSLS challenge			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
EigenPlaces	ResNet50	512	91.7	97.3	97.9	74.0	84.2	88.8	85.7	91.9	92.9	61.3	71.5	76.5	78.2	86.2	89.3
EigenPlaces	ResNet50	2048	93.5	97.3	98.6	75.4	87.2	89.3	84.8	91.8	92.9	62.0	74.5	77.3	78.5	87.6	89.8
MixVPR	ResNet50	4096	94.3	98.0	99.0	84.7	92.1	94.1	87.7	92.7	94.6	63.5	75.6	79.9	82.6	89.6	91.9
Ours	ResNet50	512	91.6	97.0	97.9	84.1	92.7	95.2	87.8	93.5	94.7	63.9	76.4	80.0	81.9	89.9	92.0
Ours	ResNet50	1024	92.4	97.5	98.3	87.9	92.7	94.3	89.1	93.5	95.7	66.8	79.2	82.7	84.1	90.7	92.8
GeM	DINOv2-B	384	89.3	96.3	97.7	73.0	86.7	91.8	84.9	92.0	93.4	65.2	79.8	83.9	78.1	88.7	91.7
EigenPlaces	DINOv2-B	512	95.0	98.4	99.1	94.6	98.1	98.4	90.4	96.1	96.6	75.2	86.7	89.8	88.8	94.8	96.0
EigenPlaces	DINOv2-B	768	95.1	98.4	99.0	95.6	98.4	98.4	91.2	95.8	96.5	75.2	87.5	90.4	89.3	95.0	96.1
MixVPR	DINOv2-B	3072	94.9	98.6	99.3	94.0	96.5	96.5	90.5	95.4	96.4	72.5	85.2	88.2	88.0	93.9	95.1
Salad	DINOv2-B	8448	94.9	98.6	99.2	92.7	96.8	97.5	92.0	95.7	96.2	74.9	86.9	89.7	88.6	94.5	95.7
Ours	DINOv2-B	384	95.0	98.6	99.1	97.1	98.4	98.7	91.1	96.1	96.5	75.0	88.6	90.8	89.6	95.4	96.3
Ours	DINOv2-B	2048	96.0	98.7	99.3	99.1	100	100	92.7	96.8	97.4	80.1	90.4	92.4	92.0	96.5	97.3

to 512, our method still maintains a competitive average performance, with the highest R@5 and R@10 scores for 89.9% and 92.0% on average. Besides, although surpassed by the 4096-dim MixVPR and 2048-dim EigenPlaces on Pitts250k, our 1024-dim version achieves the best performance on all the other datasets, with the R@1 scores of 87.9%, 89.1%, and 66.8% on Tokyo24/7, MSLS-val, and MSLS challenge. These scores indicate a significant margin over MixVPR and EigenPlaces, with differences of more than 3%, 1%, and 3%.

4.1.2 With DINOv2-B

In experiments utilizing DINOv2-B as the backbone, our method achieves outstanding performance, leading the average score of the four datasets. Our 384-dim descriptor yields VPR scores comparable to the 768-dim EigenPlaces and 8448-dim Salad on Pitts250k, MSLS-val, and MSLS challenge, while on Tokyo24/7, our method outperforms these two methods by a notable gap of no less than 1.5% R@1 score. Furthermore, we also conduct experiments with raising our final descriptor dimension to 2048 [3]. This results in a further improvement in VPR performance. One point is worth mentioning that, with 2048-dim final descriptors, our method surprisingly achieves 99.1% R@1 and 100% R@5 scores on the Tokyo24/7 dataset. Such high scores imply that our method is robust against the variabilities and complexities inherent in real-world urban scenes, such as changes in illumination, weather, and urban dynamics. With these results, we believe that our model demonstrates a level of accuracy in place recognition that may approach strong practical performance in rapidly identifying and distinguishing complex urban scenes.

4.2. Comparison with Two-stage Methods

In Tab. 3, we compare our method against two-stage VPR methods, namely Patch-NetVLAD [8], TransVPR [23], CAHIR [15], SP-SuperGlue [19], ETR [27], DELG [6], and R^2Former [28]. The VPR performance of the comparison methods are as reported in their respective papers [8, 27, 19, 6, 23, 28, 15]. These methods depend on a second stage to re-rank the selected candidates with local features, thus resulting in high VPR performance. Nonetheless, our method still retains a leading position. From the table, we can see that with the ResNet50 backbone, our method shows competitive performance within this comparison group, attaining VPR scores on the MSLS challenge of 66.8%, 79.2%, and 82.7% on R@1, 5, and 10. When switching to DINOv2-B (2048-dim), the proposed method demonstrates an absolute leading advantage. Our R@1 scores surpass the second-best R^2Former by margins of 7.1% and 10.5% on these two datasets. In summary, our method proves to be both simple and effective, outperforming more complex re-ranking methods with just a backbone and a GeM-pooling layer. This may indicate a significant stride in VPR technology, demonstrating that the simple and streamlined approaches can also yield superior results.

4.3. Performance on Dimension Reduction

We also delve into the impact of the dimension reduction on the VPR performance, as depicted in Fig. 6. It can be seen that across all tested dimensions and regardless of the backbone used, our method consistently surpasses the early-converging GeM. We attribute this superiority again to the more targeted local feature representation of our method.

A noteworthy observation is the impressive performance of our method even with significantly reduced descriptor dimensions. For example, with the DINOv2-B backbone, our 192-dimensional descriptor already attains competitive results among the methods listed in Tab. 2, with R@1 scores of 94.2%, 90.7%, and 95.2% on the Pitts250k, MSLS-val, and Tokyo24/7 datasets. These scores are remarkable, considering that the descriptor dimension is 16 $\times$ lower than MixVPR and 44 $\times$ lower than the heaviest SALAD. Such good performance under the drastic dimension reduction underscores the efficiency of our approach in terms of information consolidation and compression, which is particularly advantageous for applications in memory-constrained

[3] Dimension expansion details are provided in the supplementary material.

Table 3. Comparison of our method with the two-stage Methods. The ResNet50 and DINOv2-B versions of our model are with descriptor dimension 1024 and 2048.

2*Methods	Pitts30k			Tokyo24/7			MSLS-Val			MSLS challenge		
	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
Patch-NetVLAD-s	87.5	94.5	94.8	70.2	78.7	82.2	77.8	84.3	86.5	48.1	59.4	62.3
Patch-NetVLAD-p	88.7	94.5	95.9	86.0	88.6	90.5	79.5	86.2	87.7	48.1	57.6	60.5
TransVPR	89.0	94.9	96.2	79.0	82.2	85.1	86.8	91.2	92.7	63.9	74.0	77.5
CAHIR	90.1	95.4	96.0	90.5	92.1	92.4	81.9	88.2	90.3	-	-	-
SP-SuperGlue	87.2	94.8	96.4	88.2	90.2	90.2	78.4	82.8	84.2	50.6	56.9	58.3
ETR-S	83.1	91.1	93.8	90.1	93.0	94.6	80.5	86.5	88.9	53.9	62.8	66.1
ETR-D	84.2	91.6	93.8	89.2	94.3	95.2	79.3	88.0	89.6	50.6	62.1	65.8
DELG	89.8	95.3	96.6	86.4	92.4	93.0	83.2	89.3	89.5	52.2	61.9	65.4
R^2former	91.1	95.2	96.3	88.6	91.4	91.7	89.7	95.0	96.2	73.0	85.9	88.8
Ours (ResNet50)	90.6	95.6	96.8	87.9	92.7	94.3	89.1	93.5	95.7	66.8	79.2	82.7
Ours (DINOv2-B)	**92.5**	**96.6**	**97.6**	**99.1**	**100**	**100**	**92.7**	**96.8**	**97.4**	**80.1**	**90.4**	**92.4**

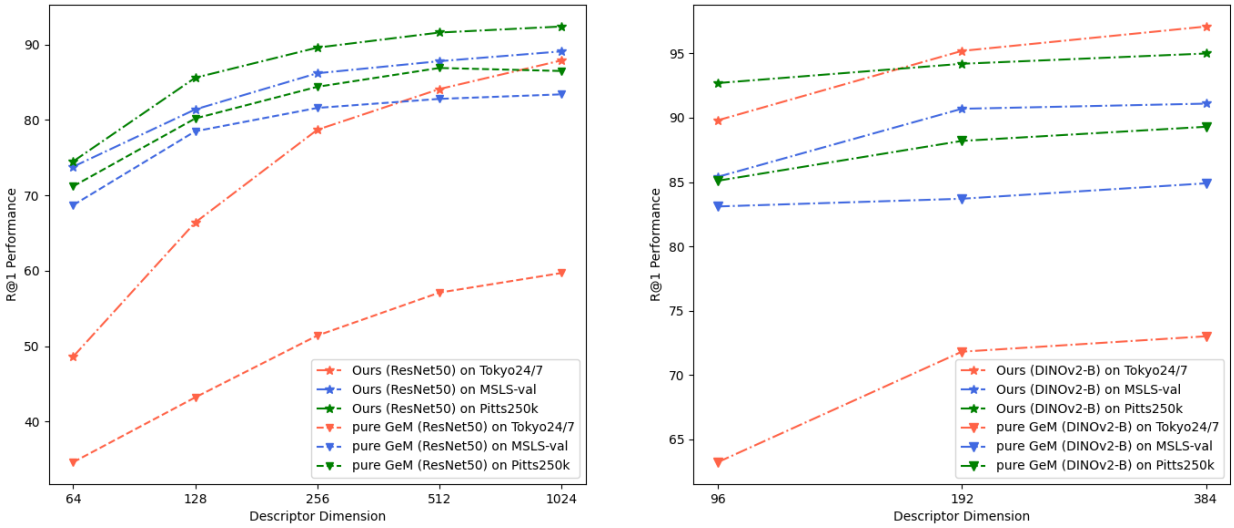


Figure 6. The R@1 performance on ours and early-converging GeM with different dimension reduction levels from the PCA-W operation.

environments like mobile robotics and embedded systems.

5. Discussion

5.1. Integrating the Regularization Branch in the inference process.

As illustrated in the previous section, the regularization branch is introduced as a *training-time* mechanism to prevent Channel vanishing by shaping the intermediate local features and providing additional gradients. During inference, we remove this branch to keep the deployment pipeline identical to standard GeM and to minimize computational overhead. Since the branch is optimized jointly with the backbone as a regularizer, it is not intended to act as an additional inference head; retaining it at test time would change the descriptor computation and complicate deployment. A dedicated quantitative study of the accuracy–efficiency trade-off when integrating the branch at inference is left for future work.

5.2. The function of the linear layer in the Regularization Branch.

The initial purpose of the linear layer in the Regularization Branch is to project the local features from the non-negative section to the whole feature space. The experimental results prove its effectiveness. However, the significant improvement of the VPR performance is more than expected. Therefore, we believe that the linear layer also has other positive functions during the training process. In addition to this projection-based regularization, alternative remedies for feature collapse in deep networks include using leaky/non-saturating activations, adding dropout, or adjusting normalization. We focus on the FC projection branch because it is a minimal, plug-and-play modification that directly enlarges the reachable feature space while preserving the GeM inference architecture. A systematic comparison and potential combinations with these alternatives are left for future work. In the future work, we will further investigate the mathematical logic of this feature.

6. Conclusion

We revisited the GeM-based Visual Place Recognition framework and uncovered a critical but overlooked issue, termed Channel Vanishing, where a large portion of descriptor channels become inactive during training. This phenomenon weakens the backbone's representational power and ultimately limits recognition performance. To address it, we introduced a lightweight regularization branch that co-trains with the GeM path to maintain channel activity and prevent feature collapse on the hypersphere boundary. The branch is removed during inference, preserving the model's efficiency while enhancing feature diversity and discriminability. Extensive experiments demonstrate that our approach consistently improves various backbones and achieves state-of-the-art results among non-transformer models, with further gains when integrated with DINOv2. These findings suggest that the main bottleneck of modern VPR may lie in under-activated backbone representations rather than the aggregation design. We hope this work motivates future studies to reconsider backbone optimization as a central direction for advancing robust and generalizable visual place recognition.

References

[1] A. Ali-bey, B. Chaib-draa, and P. Giguère. Gsv-cities: Toward appropriate supervised visual place recognition. *Neurocomputing*, 513:194–203, 2022. 4, 6, 7

[2] A. Ali-Bey, B. Chaib-Draa, and P. Giguere. Mixvpr: Feature mixing for visual place recognition. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2998–3007, 2023. 1, 2, 6, 7

[3] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5297–5307, 2016. 1, 2, 3, 6

[4] G. Berton, C. Masone, and B. Caputo. Rethinking visual geo-localization for large-scale applications. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4878–4888, 2022. 1, 2, 6

[5] G. Berton, G. Trivigno, B. Caputo, and C. Masone. Eigenplaces: Training viewpoint robust models for visual place recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11080–11090, 2023. 2, 7

[6] B. Cao, A. Araujo, and J. Sim. Unifying deep local and global features for image search. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XX 16*, pages 726–743. Springer, 2020. 2, 3, 8

[7] Y. Ge, H. Wang, F. Zhu, R. Zhao, and H. Li. Self-supervising fine-grained region similarities for large-scale image localization. In *European conference on computer vision*, pages 369–386. Springer, 2020. 2

[8] S. Hausler, S. Garg, M. Xu, M. Milford, and T. Fischer. Patch-netvlad: Multi-scale fusion of locally-global descriptors for place recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14141–14152, 2021. 2, 8

[9] S. Hausler and P. Moghadam. Pair-vpr: Place-aware pre-training and contrastive pair classification for visual place recognition with vision transformers. *IEEE Robotics and Automation Letters*, 2025. 3

[10] S. Izquierdo and J. Civera. Optimal transport aggregation for visual place recognition. *arXiv preprint arXiv:2311.15937*, 2023. 1, 3, 7

[11] S. Jian, H. Kaiming, R. Shaoqing, and Z. Xiangyu. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision & Pattern Recognition*, pages 770–778, 2016. 6, 7

[12] A. Khaliq, M. Milford, and S. Garg. Multires-netvlad: Augmenting place recognition training with low-resolution imagery. *IEEE Robotics and Automation Letters*, 7(2):3882–3889, 2022. 2

[13] L. Liu, H. Li, and Y. Dai. Stochastic attraction-repulsion embedding for large scale image localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2570–2579, 2019. 2

[14] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 2, 3, 7

[15] G. Peng, H. Li, Y. Huang, J. Zhang, M. Wen, S. Rahul, and D. Wang. Cahir: Co-attentive hierarchical image representations for visual place recognition. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6087–6094. IEEE, 2023. 2, 3, 8

[16] G. Peng, Y. Yue, J. Zhang, Z. Wu, X. Tang, and D. Wang. Semantic reinforced attention learning for visual place recognition. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13415–13422. IEEE, 2021. 2

[17] G. Peng, J. Zhang, H. Li, and D. Wang. Attentional pyramid pooling of salient visual residuals for place recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 885–894, 2021. 2, 3, 6

[18] F. Radenović, G. Tolias, and O. Chum. Fine-tuning cnn image retrieval with no human annotation. *IEEE transactions on pattern analysis and machine intelligence*, 41(7):1655–1668, 2018. 1, 2, 7

[19] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4938–4947, 2020. 2, 3, 8

[20] A. Torii, R. Arandjelovic, J. Sivic, M. Okutomi, and T. Pajdla. 24/7 place recognition by view synthesis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1808–1817, 2015. 6

[21] A. Torii, J. Sivic, T. Pajdla, and M. Okutomi. Visual place recognition with repetitive structures. In *Proceedings of the*

IEEE conference on computer vision and pattern recognition, pages 883–890, 2013. 6

[22] I. Tzachor, B. Lerner, M. Levy, M. Green, T. B. Shalev, G. Habib, D. Samuel, N. K. Zailer, O. Shimshi, N. Darshan, et al. Effovpr: Effective foundation model utilization for visual place recognition. *arXiv preprint arXiv:2405.18065*, 2024. 3

[23] R. Wang, Y. Shen, W. Zuo, S. Zhou, and N. Zheng. Transvpr: Transformer-based place recognition with multi-level attention aggregation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13648–13657, 2022. 2, 3, 8

[24] X. Wang, X. Han, W. Huang, D. Dong, and M. R. Scott. Multi-similarity loss with general pair weighting for deep metric learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5022–5030, 2019. 7

[25] F. Warburg, S. Hauberg, M. Lopez-Antequera, P. Gargallo, Y. Kuang, and J. Civera. Mapillary street-level sequences: A dataset for lifelong place recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2626–2635, 2020. 6

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

[27] H. Zhang, X. Chen, H. Jing, Y. Zheng, Y. Wu, and C. Jin. Etr: An efficient transformer for re-ranking in visual place recognition. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5665–5674, 2023. 2, 3, 8

[28] S. Zhu, L. Yang, C. Chen, M. Shah, X. Shen, and H. Wang. R2former: Unified retrieval and reranking transformer for place recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19370–19380, 2023. 2, 3, 6, 8