MixVPR: Feature Mixing for Visual Place Recognition

Amar Ali-bey, Brahim Chaib-draa, and Philippe Giguère

Université Laval, Québec, Canada

amar.ali-bey.1@ulaval.ca, {brahim.chaib-draa,philippe.giguere}@ift.ulaval.ca

Abstract

Visual Place Recognition (VPR) is a crucial part of mobile robotics and autonomous driving as well as other computer vision tasks. It refers to the process of identifying a place depicted in a query image using only computer vision. At large scale, repetitive structures, weather and illumination changes pose a real challenge, as appearances can drastically change over time. Along with tackling these challenges, an efficient VPR technique must also be practical in real-world scenarios where latency matters. To address this, we introduce MixVPR, a new holistic feature aggregation technique that takes feature maps from pre-trained backbones as a set of global features. Then, it incorporates a global relationship between elements in each feature map in a cascade of feature mixing, eliminating the need for local or pyramidal aggregation as done in NetVLAD or TransVPR. We demonstrate the effectiveness of our technique through extensive experiments on multiple large-scale benchmarks. Our method outperforms all existing techniques by a large margin while having less than half the number of parameters compared to CosPlace and NetVLAD. We achieve a new all-time high recall@1 score of 94.6% *on Pitts250k-test,* 88.0% *on MapillarySLS, and more importantly,* 58.4% *on Nordland. Finally, our method outperforms two-stage retrieval techniques such as Patch-NetVLAD, TransVPR and SuperGLUE all while being orders of magnitude faster.*

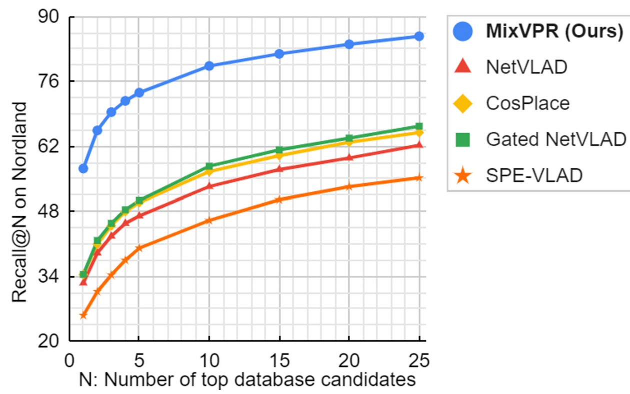

Figure 1. Comparison of performance on the challenging Nordland benchmark. All methods have been trained on the exact same dataset, using the same backbone architecture.

1. Introduction

Visual place recognition (VPR) is an essential part of many robotics [11, 9, 10, 15, 18, 22] and computer vision tasks [2, 23, 27, 16, 17, 45, 6] such as autonomous driving [12], SLAM [49], image geo-localization [38, 7], virtual reality [31] and 3D reconstruction [29]. A visual place recognition system retrieves the location of a given query image by first gathering its visual information into a compact descriptor (image representation), then matching it against a database of references with known geolocations. This task can be extremely challenging due to short term appearance changes (e.g., illumination, occlusion and weather) as well as long term variations (e.g., seasonal changes, construction and vegetation). Therefore, a robust VPR technique should be capable of producing descriptors that are invariant to these changes.

Traditionally, VPR technique used hand-crafted local features such as SIFT [30] and SURF [5] which can be further aggregated into a global descriptor that represents the entire image such as Fisher Vectors [20, 34], Bag of Words [35, 44, 14] and Vector of Locally Aggregated Descriptor (VLAD) [21, 3]. Following the growth of deep learning, where convolutional neural networks (CNNs) have shown outstanding performance in several computer vision tasks, including image classification [19], object detection [28] and semantic segmentation [25], many researchers have proposed to use CNNs for VPR. For instance, Sünderhauf *et al.* [40] showed that features extracted from intermediate layers of CNNs trained for image classification can perform better than hand-crafted features. As a result, many have proposed to train CNNs directly for the task of place recognition [2, 39, 23, 27, 16], by designing end-to-end trainable layers that can be plugged into pre-trained networks (backbones) to aggregate their rich feature maps into robust representations. These approaches demonstrated great success at large scale benchmarks [44, 47] thanks to the availability of pre-trained networks and the VPR-specific datasets for fine-tuning.

Despite all the progress in the field of visual place recognition, most existing state-of-the-art techniques either use NetVLAD [2, 47, 17, 50] or provide a variant that incorporates attention [52], context [23], semantics [33] or multi-scale [17]. These techniques emphasize on the aggregation of local features which have proved to be invariant to viewpoint changes. However, local features are notoriously known to fail under severe illumination and seasonal changes [31].

Alternative approaches to NetVLAD focus on regions of interests instead of local features, by spatially pooling from the feature maps of the backbone. Such techniques include MAC (i.e., max pooling), R-MAC [42] and Generalized Mean (GeM) [36]. Despite their performance in image retrieval [8] these methods have been repeatedly shown to underperform NetVLAD in the task of VPR. Most recently, Berton *et al*. [6] proposed CosPlace, which is a variant that builds on GeM aggregator, showing strong performance on multiple VPR benchmarks.

Currently, all existing state-of-the-art techniques propose shallow aggregation layers that are plugged into very deep pre-trained backbones cropped at the last feature-rich layer. By contrast, Wang *et al*. [45] proposed TransVPR, a place recognition architecture that builds on the success of vision Transformers [13] and fuse multi-level attentions to generate global and local descriptors. TransVPR achieved strong results for local feature matching. However, its global representation performance did not surpass that of NetVLAD or CosPlace. With recent advances in isotropic architectures, it has been shown that self-attention is not critical to Vision Transformers [26]. For instance, Tolstikhin *et al*. [43] introduced MLP-Mixer, an architecture based exclusively on multi-level perceptrons, achieving competitive results on multiple vision tasks.

In this paper, we present MixVPR, a new holistic aggregation technique that uses feature maps extracted from a pre-entrained backbone, and iteratively incorporates global relationships into each individual feature map. It does this through a stack of isotropic blocks that we call Feature-Mixer, which consists solely of multi-layer perceptrons (MLPs). The effectiveness of MixVPR is demonstrated by several qualitative and quantitative results where it achieves a new state-of-the-art performance on multiple benchmarks, surpassing existing techniques by a wide margin all while being extremely lightweight.

2. Related Works

The task of visual place recognition has long been approached as an image retrieval problem, where the location of a query image is determined according to the geotags of the most relevant images retrieved from a reference database. With the success of deep learning, almost all recent VPR techniques make use of learned representations. This usually involves using features extracted from a backbone network pretrained on image classification datasets [24], followed by a trainable aggregation layer that transforms these features into robust compact representations. One notable aggregation technique is NetVLAD [2], which is a trainable variant of the VLAD descriptor, where local features are softly assigned to a learned set of clusters. As a result of the success of NetVLAD, many variants have been proposed in literature. Kim *et al*. [23] introduced Contextual Reweighting Network (CRN) which estimates a weight for each local feature from the backbone before feeding it into a NetVLAD layer; their approach introduced a slight but consistent performance boost. Further on, SPE-VLAD [50] has been proposed, to enhance NetVLAD with spatial and regional features, by incorporating pyramid structure. More recently, Zhan *et al*. [52] proposed Gated NetVLAD, which uses a gating mechanism that incorporates attention in the computation of NetVLAD residuals.

Other techniques focus on regions of interest in the feature maps. Among the first techniques is MAC [4], a simple aggregation method that applies max-pooling on each individual feature map, selecting only the most activated neurons. Building on that, Tolias *et al*. [42] introduced R-MAC (Regional Maximum Activations of Convolutions) that consists of extracting multiple Region of Interest (RoI) directly from the CNN feature maps to form representations. These techniques showed impressive performance on the task of image retrieval and have since been used in VPR. Another notable aggregation technique is the Generalized Mean (GeM) [36] which is a learnable generalized form of global pooling. Building on GeM, Berton *et al*. [6] recently proposed CosPlace, a lightweight aggregation technique that combines GeM with a linear projection layer. Their method showed impressive performance on the task of VPR, outperforming GeM and NetVLAD and achieving state-of-the-art results on multiple benchmarks.

Another trend in recent VPR works [17, 45] is to consider using a two-stage retrieval strategy, which consists of running a first global retrieval step to retrieve, for each query, the top k candidates from the reference database. This step is generally more efficient because it uses k-NN on the global descriptors. Then, a second computationally heavy step is performed where the candidates are re-ranked according to their local features [41, 37, 38]. For instance, Patch-NetVLAD [17] uses NetVLAD descriptor for global

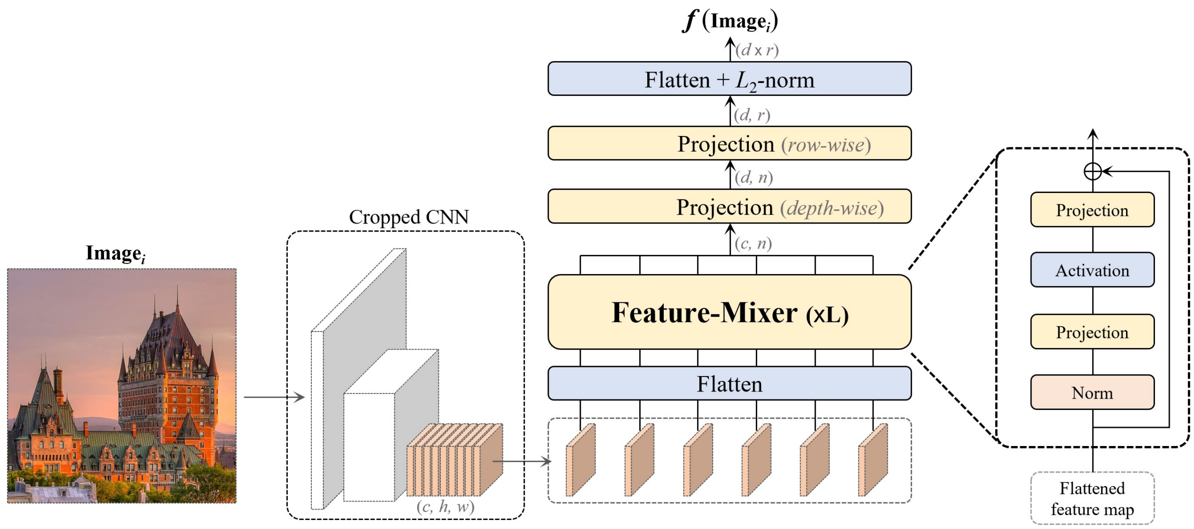

Figure 2. Overview of our newly proposed architecture for place recognition. MixVPR takes as input flattened feature maps from intermediate layers of a pretrained backbone. It incorporates spatial relationship in each individual feature map through a succession of Feature-Mixer blocks. The resulting output is then projected into a compact representation space and used as global descriptor.

description, then in a later stage, uses the local features composing NetVLAD in order to refine the retrieved candidates. This approach demonstrated good performance when re-ranking is used. Recently, TransVPR [45] used a combination of CNN and Transformer by using multi-head self-attention (Transformer encoder) on top of a shallow CNN backbone. Their aim is to incorporate attention in the resulting tokens of the Transformer network. While their local feature demonstrated great performance for re-ranking, the global descriptors generated by the transformer network were not as powerful as NetVLAD or CosPlace.

In this paper, we follow recent advances in isotropic all-MLP architectures such as MLP-Mixer [43] and gMLP [26], and propose MixVPR, a novel all-MLP aggregation technique, which in contrast to TransVPR [45] and Patch-NetVLAD [17], does not incorporate self-attention or regional feature pooling. Although our method, MixVPR, generates global descriptors and does not perform re-ranking, it outperofms two-stage techniques such as TransVPR [45], Patch-NetVLAD [17] and SuperGlue [38], while being at least $500\times$ faster in terms of latency.

3. Methodology

Our aim is to learn global compact representations that integrate features in a holistic way. Given an image $\mathcal{I}$, we first extract its feature maps $\mathbf{F} \in \mathbb{R}^{c\times h\times w}$ from the intermediate layers of a CNN backbone, $\mathbf{F} = \text{CNN}(\mathcal{I})$. Existing techniques, such as TransVPR [45], Patch-NetVLAD [17], NetVLAD [2], consider $\mathbf{F}$ as a set of c-dimensional spatial descriptors, where each descriptor corresponds to a receptive field in the input image. In contrast, we consider the 3D tensor $\mathbf{F}$ as a set of 2D features of size $h\times w$ such as:

$$\mathbf{F} = \{X^i\}, \quad i = \{1, \ldots, c\} \tag{1}$$

where X^i corresponds to the i^{th} activation map in $\mathbf{F}$ and sweeps across all the image (each feature map carries a certain amount of information regarding the whole image). We reshape each 2D feature X^i into a 1D representation (flattening), resulting in flattened feature maps $\mathbf{F} \in \mathbb{R}^{c\times n}$, where $n = h\times w$.

Then, we feed them to what we call *Feature-Mixer*, a cascade of L MLP blocks of identical structure, as illustrated in Fig. 2. Feature-Mixer takes as input a set of flattened feature maps, and incorporates global relationships into each $X^i \in \mathbf{F}$ as follows (omitting Normalization layer):

$$X^i \leftarrow \mathbf{W}_2(\sigma(\mathbf{W}_1 X^i)) + X^i, \quad i = \{1, \ldots, c\} \tag{2}$$

where $\mathbf{W}_1$ and $\mathbf{W}_2$ are the weights of two fully-connected layers that compose the MLP, and σ is a nonlinearity (ReLU in our case). The input to the MLP is added back to the resulting projection in a skip connection. This is proven to help the flow of gradients and further improve performance [19].

The intuition behind Feature-Mixer is that, instead of focusing on local features, and forcing the network to go through attention mechanism, we take advantage of the capacity of fully connected layers to automatically aggregate features in a holistic way. Feature-Mixer replaces hierarchical (pyramidal) aggregation thanks to its full receptive field, where each neuron has a glimpse into the entire input image. We use a cascade of Feature-Mixer blocks as shown in Fig. 2 in order to iteratively incorporate relationships between spatial features in each individual feature map.

For a given input $\mathbf{F} \in \mathbb{R}^{c \times n}$, Feature Mixer (FM) generates an output $\mathbf{Z} \in \mathbb{R}^{c \times n}$ of the same shape (due to its isotropic architecture), which we feed into a second Feature-Mixer block, and so on until we reach L consecutive blocks, as follows:

$$\mathbf{Z} = FM_L(FM_{L-1}(\dots FM_1(\mathbf{F}))) \tag{3}$$

$\mathbf{Z}$ is usually highly dimensional (as it has the same dimensionality as the extracted feature maps $\mathbf{F}$). To further reduce its dimensionality, we follow it by two fully connected layers that reduce its dimension depth-wise (channel-wise) then row-wise, successively. This can be seen as a weighted pooling operation that enables control of the size of the final global descriptor. First, we apply a depth-wise projection that maps $\mathbf{Z}$ from $\mathbb{R}^{c \times n}$ to $\mathbb{R}^{d \times n}$ as follows:

$$\mathbf{Z}' = \mathbf{W}_d(Transpose(\mathbf{Z})) \tag{4}$$

where $\mathbf{W}_d$ are the weights of a fully-connected layer. We then apply a row-wise projection that maps the output $\mathbf{Z}'$ from $\mathbb{R}^{d \times n}$ to $\mathbb{R}^{d \times r}$ such as:

$$\mathbf{O} = \mathbf{W}_r(Transpose(\mathbf{Z}')) \tag{5}$$

where $\mathbf{W}_r$ are the weights of another fully-connected layer. The final output $\mathbf{O}$ has a dimensionality of $d \times r$, which is flattened and L_2-normalized as usually done in VPR [2, 16, 6].

Connection to existing architectures. Our technique is related to MLP-Mixer [43] where a token mixing operation is applied on spatial non-overlapping image patches. We, on the other hand, use features from CNNs that incorporate inductive bias and regard the resulting activation maps as *global features*. Also, MLP-Mixer performs channel-mixing that is shared across individual spatial descriptors, which we do not employ.

Overall, MixVPR computations are mostly matrix multiplications (of fully-connected layers) which are efficient in terms of computation compared to self-attention where the complexity scales quadratically [43]. Also, in MixVPR we extract feature maps from the intermediate layers (instead of the last layer) of the backbone, which reduces the number of parameters by more than half, as most parameters of a pre-trained backbone are present in the last layers.

4. Experiments

In this section, we run extensive experiments to show the effectiveness of the proposed MixVPR compared to existing state-of-the-art techniques by evaluating on multiple challenging benchmarks. In what follows, we present implementation details, datasets, evaluation metrics, performance comparisons and ablation studies.

4.1. Implementation details

Architecture. We implement MixVPR in PyTorch framework [32] and use existing implementations of GeM [36], NetVLAD [2] and CosPlace [6]. However, for techniques without existing implementation, such as SPE-NetVLAD [50] and Gated NetVLAD [52], we do our best to faithfully reimplement them following their respective papers. For all techniques, the CNN backbone is cropped at the last convolutional layer as recommended by their authors. MixVPR uses a backbone cropped in the middle (i.e., at the second last ResNet residual block) so that the Feature Mixer receives feature maps with a spatial dimension of 20×20. For maximum fairness, we use the exact same CNN backbone for all compared techniques (i.e., ResNet-50 [19]). The projection operation in Feature-Mixer is the Linear layer of PyTorch which we follow by a relu nonlinearity. As for the normalization layer we use LayerNorm. Finally, the output of the Feature-Mixer is projected into a smaller representation space using two consecutive fully-connected layer as described in 3, which makes MixVPR an all-MLP architecture. Unless otherwise stated, we fix $L = 4$ the number of stacked Feature-Mixer blocks.

Training. Using a ResNet [19] backbone pre-trained on ImageNet [24], we train all techniques on the same dataset, following the standard framework of GSV-Cities [1], which proposes a highly accurate dataset of 67k places depicted by 560k images. For the loss function, we use Multi-Similarity loss [46] as it has been shown to perform best for visual place recognition [1].We use batches containing $P = 120$ places, each depicted by 4 images resulting in mini-batches of 480 images. We use Stochastic Gradient Descent (SGD) for optimization, with momentum 0.9 and weight decay of 0.001. The initial learning rate of 0.05 is divided by 3 after each 5 epochs. Finally, we train for a maximum of 30 epochs using images resized to 320×320.

Evaluation. For evaluation we use the following 5 benchmarks. Pitts250k-test [44], which contains 8k queries and 83k reference images, collected from Google Street View and Pitts30k-test [44] which is a subset of Pitts250k and comprises 8k queries and 8k references. Both Pittsburgh datasets show significant viewpoint changes. SPED [51] benchmark contains 607 queries and 607 references from surveillance cameras presenting significant seasonal and illumination variations. MSLS [47] benchmark has been collected using car dashcams and presents a wide range of viewpoint and illumination changes. Finally, Nordland [51] is an extremely challenging benchmark which has been collected in 4 seasons using a camera mounted in front of a train, it comprises scenes ranging from snowy winter to sunny summer with extreme appearance changes. We follow the same evaluation metric of [2, 23, 47, 51, 45, 6], where the recall@k is measured. The query image is determined to be successfully retrieved if at least one of the top-k

retrieved reference images is located within $d = 25$ meters from the query one.

4.2. Comparison to the state of the art

In this section, we compare the performance of MixVPR against existing methods in visual place recognition on 4 challenging benchmarks. We compare against AVG [2], GeM [36], NetVLAD [2] and two of its recent variants SPE-VLAD [50] and Gated NetVLAD [52], and CosPlace which recently demonstrated state-of-the-art performance. Results are shown in Table 1. The lines with the sign † are performance of AVG, GeM and NetVLAD trained on Pitts30k-train dataset. For fair comparison, we re-train them using the same backbone and dataset as our technique. Results are shown in the rest of the table. As can be seen, our method convincingly outperforms all other techniques on all benchmarks with a large margin. For instance, MixVPR achieves a new all-time high recall@1 of $\mathbf{94.6}\%$ on Pitts250k-test which is 3.1 percentage points increase over the recent CosPlace technique and over 4.1 points increase compared to NetVLAD.

On MSLS, performance is even more interesting, where we achieve $\mathbf{88.0}\%$ recall@1, which, to the best of our knowledge, is the best score ever achieved. This is 3.5 and 5.4 percentage points increase over CosPlace and NetVLAD which achieved 84.5% and 82.6% recall@1 respectively. This showcases the effectiveness of our technique on datasets presenting a lot of viewpoint variations.

On SPED benchmark, where places exhibit drastic appearance change due to seasonal changes and day-night illumination, our technique surpasses all other techniques achieving $\mathbf{85.2}\%$ recall@1, which is 7.5 points more than NetVLAD, the second best performing technique on SPED.

Finally and most importantly, on the extremely challenging Nordland benchmark, MixVPR achieves 69% and and 79% relative improvement over CosPlace and NetVLAD ($\mathbf{58.4}\%$ vs 34.4% and 32.6% resp.), and more than double compared to the rest of the other techniques.

4.3. Comparing against two-stage techniques

Some techniques use a two-stage retrieval framework, where a first pass is performed to retrieve the best M candidates using global representations, then a second pass (re-ranking) is executed to perform geometric verification on the local features between the query and each one of the M candidates [45]. This is known to increase recall@N performance at the expense of heavy computation and memory overhead. We compare against Patch-NetVLAD [17], DELG [7], SuperGlue [38] and TransVPR [17] which are state-of-the-art techniques that perform two-stage visual place recognition. Table 2 shows performance on the Mapillary Challenge. Although our technique does not perform any re-ranking, it achieves better performance than existing two-stage techniques while being orders of magnitudes more efficient in terms of memory and computation (over $500\times$ faster retrieval time). We believe that MixVPR can replace two-stage techniques in applications where time and resources are of great importance. For instance, MixVPR takes only 6 milliseconds to generate an image representation, while the second fastest method, TransVPR, takes 45 milliseconds. Matching latency does not apply to MixVPR since it is a global technique that does not perform re-ranking. However, it is clear from Table 2 that the re-ranking phase takes a lot of time, making such techniques unusable in real-time applications.

4.4. Ablation studies

We conduct multiple ablation experiments to further validate the design of MixVPR.

4.4.1 Hyperparameters

In order to showcase the effect of Feature-Mixer, we conduct multiple experiments by varying L the number of Feature-Mixer blocks. First, we train a baseline network without Feature-Mixer ($L = 0$), and compare its performance when trained with multiple stacked Feature-Mixer blocks ($L \in \{1, 2, 4, 8\}$). Results are shown in Table 3, where we see that introducing only one Feature-Mixer layer improves recall@1 performance by 1.8 recall@1 points from 89.5% to 91.3% on Pitts30k-test and 4 on MSLS from 82.9% to 86.9%. Overall, the best results are obtained with 4 Feature-Mixer layers, although all configurations achieve similar performance. Feature-Mixer adds 340k parameters to the network, therefore we can refer to Table 3 to choose the best compromise.

4.4.2 Descriptor dimensionality

The architecture of MixVPR allows to configure the dimensionality of the output descriptor, by fixing the size of the last two projection operations. In 3 we show recall@1 performance for different dimensionality configurations on Pitts30k-test. For NetVLAD, GeM and AVG, we used PCA dimensionality reduction learned on a subset of 10k images from the training set. CosPlace, like MixVPR, allows to configure the output dimensionality. Hence, we trained once for each configuration. From the chart in Fig. 3, we can clearly see that MixVPR outperforms all other techniques.

4.4.3 Backbone architecture

In Table 4 we conduct multiple experiments using different backbone architectures. Since we crop the backbone at

Method	dim	Pitts250k-test			MSLS-val			SPED			Nordland		
		R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
AVG [2] †	2048	62.6	82.7	88.4	59.3	71.9	75.5	54.7	72.5	77.1	4.4	8.4	10.4
GeM [36] †	2048	72.3	87.2	91.4	65.1	76.8	81.4	55.0	70.2	76.1	7.4	13.5	16.6
NetVLAD [2] †	32768	86.0	93.2	95.1	59.5	70.4	74.7	71.0	87.1	90.4	4.1	6.6	8.2
AVG [2]	2048	78.3	89.8	92.6	73.5	83.9	85.8	58.8	77.3	82.7	15.3	27.4	33.9
GeM [36]	2048	82.9	92.1	94.3	76.5	85.7	88.2	64.6	79.4	83.5	20.8	33.3	40.0
NetVLAD [2]	32768	90.5	96.2	97.4	82.6	89.6	92.0	78.7	88.3	91.4	32.6	47.1	53.3
SPE-NetVLAD [50]	163840	89.2	95.3	97.0	78.2	86.8	88.8	73.1	85.5	88.7	25.5	40.1	46.1
Gated NetVLAD [52]	32768	89.7	95.9	97.1	82.0	88.9	91.4	75.6	87.1	90.8	34.4	50.4	57.7
CosPlace [6]	2048	91.5	96.9	97.9	84.5	90.1	91.8	75.3	85.9	88.6	34.4	49.9	56.5
MixVPR (Ours)	2048	94.1	98.2	98.8	87.0	92.7	94.2	84.7	92.1	94.4	57.9	73.8	79.0
MixVPR (Ours)	4096	**94.6**	**98.3**	**99.0**	**88.0**	**92.7**	**94.6**	**85.2**	**92.1**	**94.6**	**58.4**	**74.6**	**80.0**

Table 1. **Comparison of different techniques on popular benchmarks.** † are results reported by the authors and confirmed using their trained networks. We however, train all six techniques on the same dataset using the same backbone network (ResNet-50). NetVLAD and its variants obtain third best performance just after the recent CosPlace method. Our technique, MixVPR, obtains by far the best performance on all benchmarks, and with big margins.

Method	Extraction latency (ms)	Matching latency (s)	Mapillary Challenge		
			R@1	R@5	R@10
Super-Glue [38]	160	7.5	50.6	56.9	58.3
DELG [7]	190	35.2	52.2	61.9	65.4
Patch-NetVLAD [17]	1300	7.4	48.1	59.4	62.3
TransVPR [45]	45	3.2	63.9	74.0	77.5
NetVLAD [2]	17	–	35.1	47.4	51.7
MixVPR (Ours)	6	–	**64.0**	**75.9**	**80.6**

Table 2. **Comparison with two-stage retrieval techniques.** The first four techniques use a second refinement pass (matching) to re-rank the top candidates in order to improve retrieval performance. MixVPR (ours) does not use re-ranking, which makes it at least 500× faster all while outperforming existing state-of-the-art. (a NVIDIA Titan Xp has been used to calculate latency).

$\times L$	# params (M)	Latency (ms)	Pitts30k-test			MSLS-val		
			R@1	R@5	R@10	R@1	R@5	R@10
0	9.6	6.3	89.5	95.0	96.2	82.9	90.7	91.9
1	9.9	6.5	91.3	95.6	96.5	86.9	92.8	94.3
2	10.2	6.6	91.3	95.8	96.6	87.6	93.1	94.6
4	10.9	6.6	91.9	**95.9**	**96.7**	**87.6**	**93.5**	**95.0**
8	12.2	7.2	**92.3**	95.9	96.6	87.2	92.6	93.9

Table 3. **Ablation on the number of Feature-Mixer blocks.** The baseline ($L = 0$) does not use Feature-Mixer. We compare it to different configurations by varying L the number of stacked Feature-Mixer blocks. Overall, $L = 4$ stacks of Feature-Mixer performs the best on all benchmarks.

the 4^{th} residual layer (instead of the last) we end up cropping out half the total number of parameters, thus accelerating computation and reducing memory use. As can be seen in Table 4. Using ResNet-18 [19] we end up with only 3.5M parameters, which is 15% the number of parameters in CosPlace or NetVLAD, all while getting competitive results. We believe ResNet-18 can be used in applications where real-time is top priority. Importantly, MixVPR obtains state-of-the-art performance using only ResNet-34 which comprises 70% less parameters compared to CosPlace while outperforming it by 2.3 recall@1 points on MSLS. The best overall results are obtained with ResNet-

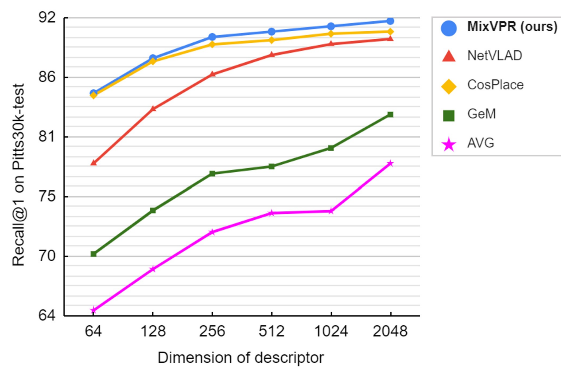

Figure 3. Recall@1 performance on Pitts30k-test with different dimensionality configurations.

50 where the number of parameters (10.9M) is less than half that of NetVLAD or CosPlace. Interestingly, using ResNeXt50 [48] did not increase performance compared to ResNet-50. We believe this is because MixVPR draws much of its performance from the Feature Mixing rather than the backbone network.

Backbone	# param. (M)	Pitts30k-test			MSLS-val		
		R@1	R@5	R@10	R@1	R@5	R@10
ResNet-18	3.5	89.5	95.0	96.2	82.7	89.1	91.8
ResNet-34	8.2	90.5	95.2	96.3	85.3	91.6	93.4
ResNet-50	10.9	91.6	96.0	96.7	88.0	92.8	94.5
ResNeXt-50	10.9	91.7	95.7	96.5	87.0	93.5	94.7

Table 4. **Comparing different backbones.** Each backbone is cropped at the fourth residual block (before the last one), which results in half the number of parameters of the same backbone used in CosPlace or netVLAD. MixVPR only needs intermediate features of the backbone.

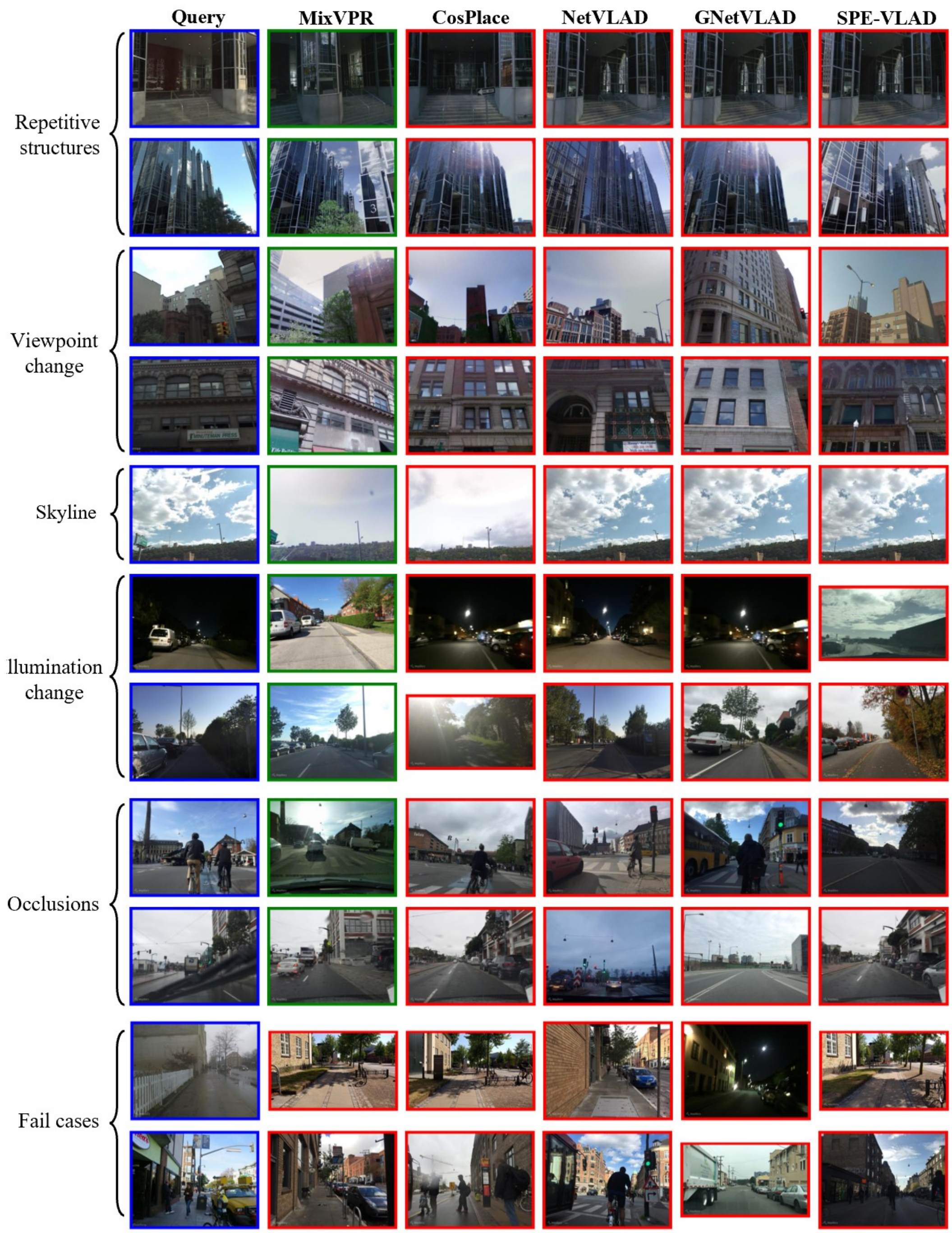

Figure 4. Comparison of challenging retrieval scenarios on MSLS and Pitts30k datasets. MixVPR succeeds the retrieval of all these challenging queries, while all other techniques fail. This qualitative results highlight the robustness of MixVPR to extreme scenarios.

4.5. Qualitative Results

Fig. 4 illustrates qualitative results of the retrieval of some challenging queries. We discuss 5 scenarios where all other techniques struggle retrieving the correct match while MixVPR succeeds. **Repetitive structures:** this is a serious problem for VPR techniques, since different places may contain the same type of building or structure with the same layout or texture, this can fool the recognition system and induce a lot of false positives as we can see in the first two rows of Fig. 4, where only MixVPR succeeded in retrieving the right reference, while all other techniques retrieved images of different places that are overly similar to the query. **Viewpoint change:** for this scenario, techniques that focus on local features, such as NetVLAD, tend to perform better. However, in rows 3-4 of Fig 4, only MixVPR retrieved the right references, which highlights its capacity to deal with extreme viewpoint changes. **Skyline:** some environments contain few static structures such as buildings and poles, making the image lack distinctive textures. In this case, the skyline constitutes an important signature of the place. As we can see in row 5 of Fig 4, only MixVPR succeeded in retrieving the correct reference based most likely on the skyline all while ignoring the cloud texture. **Illumination change:** we believe this to be the most important aspect of a robust VPR system, because illumination variations occur on a daily basis, such an example is illustrated in rows 6-7 of Fig 4 where the query is taken during the night and its reference is taken during the day. CosPlace, NetVLAD and Gated NetVLAD all retrieved images of locations taken at nighttime, in contrast, MixVPR retrieved the correct reference even though it is visually very tricky even for the human eye. This highlights the robustness of our method in extremely challenging situations. **Occlusions:** this can be challenging when part of the image is obstructed with an object that can affect the global semantic of the image. For instance, row 8 of Fig 4 shows a query with two cyclists in the middle of the field of view (FoV), which tricked other techniques to retrieve the wrong references containing cyclists in the middle of the FoV. Only MixVPR ignored the cyclists and successfully retrieved the right reference. Finally, we show two cases where all techniques fail, due to extreme environmental changes and the presence of a lot of occlusions.

4.5.1 Visualizing learned weights

Fig 5 illustrates a subset of learned weights from the first hidden layer of Feature-Mixer (24 neurons out of 400). The weights of each unit have been reshaped to 20×20 to match the spatial size of feature maps coming from the backbone. As we can see, hidden units in Feature-Mixer learned a wide range of regional feature selection. We observe that some neurons focus on one or multiple small spots of the image, while other focus on the entire input. We believe the combination of these neurons can replace attention and pyramidal scheme in deep model for VPR.

5. Conclusion

In this work, we designed a novel all-MLP aggregation technique that employs feature maps from pretrained networks, and learns robust representations in a cascade of feature mixing. MixVPR is composed of a stack of Feature-Mixer blocks, where each block incorporates global relationships between individual feature maps. We demonstrated the effectiveness of the feature mixing through ablation studies, and showed that MixVPR outperforms existing state-of-the-art by a wide margin on every benchmark we tested on. Finally, we also compared performance of MixVPR against two-stage retrieval techniques such as Patch-NetVLAD and TransVPR and showed that our method is superior while being over $500\times$ faster.

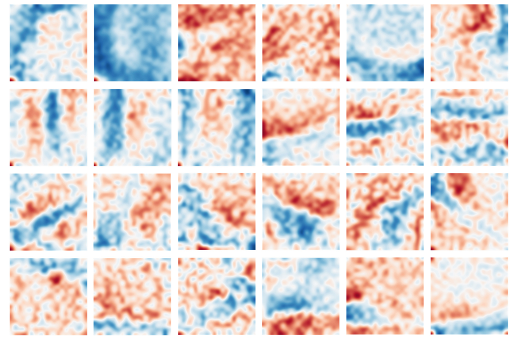

Figure 5. Illustration of learned weights from a subset of 24 neurons from the first Feature-Mixer block. Blue color corresponds to positive weights and Red corresponds to negative weights.

Acknowledgement: This work has been supported by The Fonds de Recherche du Québec Nature et technologies (FRQNT). We gratefully acknowledge the support of NVIDIA Corporation with the donation of a Quadro RTX 8000 GPU used for our experiments.

References

[1] Amar Ali-bey, Brahim Chaib-draa, and Philippe Giguère. GSV-Cities: Toward Appropriate Supervised Visual Place Recognition. *Neurocomputing*, 513:194–203, 2022.

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

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

[4] Artem Babenko and Victor Lempitsky. Aggregating local deep features for image retrieval. In *Proceedings of the IEEE international conference on computer vision*, pages 1269–1277, 2015.

[5] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *European conference on computer vision*, pages 404–417. Springer, 2006.

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

[7] Bingyi Cao, Andre Araujo, and Jack Sim. Unifying deep local and global features for image search. In *European Conference on Computer Vision*, pages 726–743. Springer, 2020.

[8] Wei Chen, Yu Liu, Weiping Wang, Erwin Bakker, Theodoros Georgiou, Paul Fieguth, Li Liu, and Michael S Lew. Deep learning for instance retrieval: A survey. *arXiv preprint arXiv:2101.11282*, 2021.

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

[10] Zetao Chen, Lingqiao Liu, Inkyu Sa, Zongyuan Ge, and Margarita Chli. Learning context flexible attention model for long-term visual place recognition. *IEEE Robotics and Automation Letters*, 3(4):4015–4022, 2018.

[11] Zetao Chen, Fabiola Maffra, Inkyu Sa, and Margarita Chli. Only look once, mining distinctive landmarks from convnet for visual place recognition. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 9–16. IEEE, 2017.

[12] Anh-Dzung Doan, Yasir Latif, Tat-Jun Chin, Yu Liu, Thanh-Toan Do, and Ian Reid. Scalable place recognition under appearance change for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9319–9328, 2019.

[13] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

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

[15] Sourav Garg, Niko Suenderhauf, and Michael Milford. Semantic–geometric visual place recognition: a new perspective for reconciling opposing views. *The International Journal of Robotics Research*, page 0278364919839761, 2019.

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

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

[18] Stephen Hausler, Adam Jacobson, and Michael Milford. Multi-process fusion: Visual place recognition using multiple image processing methods. *IEEE Robotics and Automation Letters*, 4(2):1924–1931, 2019.

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

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

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

[22] Ahmad Khaliq, Shoaib Ehsan, Zetao Chen, Michael Milford, and Klaus McDonald-Maier. A holistic visual place recognition approach using lightweight cnns for significant viewpoint and appearance changes. *IEEE transactions on robotics*, 36(2):561–569, 2019.

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

[24] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

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

[26] Hanxiao Liu, Zihang Dai, David So, and Quoc V Le. Pay attention to mlps. *Advances in Neural Information Processing Systems*, 34:9204–9215, 2021.

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

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

[29] Zhe Liu, Shunbo Zhou, Chuanzhe Suo, Peng Yin, Wen Chen, Hesheng Wang, Haoang Li, and Yun-Hui Liu. Lpd-net: 3d point cloud learning for large-scale place recognition and environment analysis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2831–2840, 2019.

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

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

[32] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

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

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

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

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

[37] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12716–12725, 2019.

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

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

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

[41] Hajime Taira, Masatoshi Okutomi, Torsten Sattler, Mircea Cimpoi, Marc Pollefeys, Josef Sivic, Tomas Pajdla, and Akihiko Torii. Inloc: Indoor visual localization with dense matching and view synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7199–7209, 2018.

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

[43] Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, et al. Mlp-mixer: An all-mlp architecture for vision. *Advances in Neural Information Processing Systems*, 34:24261–24272, 2021.

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

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

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

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

[48] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1492–1500, 2017.

[49] Rohit Yadav and Rahul Kala. Fusion of visual odometry and place recognition for slam in extreme conditions. *Applied Intelligence*, pages 1–20, 2022.

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

[51] Mubariz Zaffar, Sourav Garg, Michael Milford, Julian Kooij, David Flynn, Klaus McDonald-Maier, and Shoaib Ehsan. Vpr-bench: An open-source visual place recognition evaluation framework with quantifiable viewpoint and appearance change. *International Journal of Computer Vision*, pages 1–39, 2021.

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