Optimal Transport Aggregation for Visual Place Recognition

Sergio Izquierdo Javier Civera
I3A, University of Zaragoza, Spain
{izquierdo, jcivera}@unizar.es

Abstract

The task of Visual Place Recognition (VPR) aims to match a query image against references from an extensive database of images from different places, relying solely on visual cues. State-of-the-art pipelines focus on the aggregation of features extracted from a deep backbone, in order to form a global descriptor for each image. In this context, we introduce SALAD (Sinkhorn Algorithm for Locally Aggregated Descriptors), which reformulates NetVLAD's soft-assignment of local features to clusters as an optimal transport problem. In SALAD, we consider both feature-to-cluster and cluster-to-feature relations and we also introduce a 'dustbin' cluster, designed to selectively discard features deemed non-informative, enhancing the overall descriptor quality. Additionally, we leverage and fine-tune DINOv2 as a backbone, which provides enhanced description power for the local features, and dramatically reduces the required training time. As a result, our single-stage method not only surpasses single-stage baselines in public VPR datasets, but also surpasses two-stage methods that add a re-ranking with significantly higher cost. Code and models are available at https://github.com/serizba/salad.

1. Introduction

Recognizing a place solely from images becomes a challenging task when scenes undergo substantial changes in their structure or appearance. Such capability is referred to in the scientific and technical literature as visual place recognition (and by its acronym VPR), and is essential for agents to navigate and understand their surroundings autonomously in a wide array of applications, such as robotics [12–14, 22, 29] or augmented reality [19]. Specifically, it is present in simultaneous localization and mapping [9, 10] and absolute pose estimation [25, 44] pipelines.

In practice, VPR is framed as an image retrieval problem, wherein typically a query image serves as the input and the goal is to obtain an ordered list of top-k matches against a pre-existing database of geo-localized reference images. Images are represented as an aggregation of appearance pattern descriptors, which are subsequently compared via nearest neighbour. The effectiveness of this matching relies on generating discriminative per-image descriptors that exhibit robust performance even for challenging variations such as fluctuating illumination, structural transformations, temporal changes, weather and seasonal shifts. Most recent research on VPR have thus focused on the two key compo-

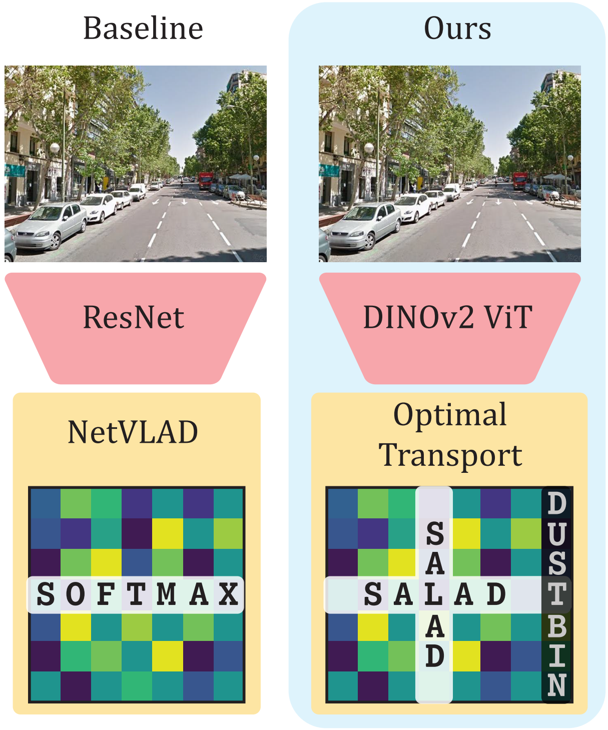

Figure 1. **Illustration of a VPR baseline (left) and our contribution (right).** The left column outlines a typical VPR baseline, a ResNet backbone followed by NetVLAD aggregation [4]. On the right column, we replace ResNet with a partially fine-tuned DINOv2 [41] backbone, and incorporate SALAD, our novel optimal transport aggregation using the Sinkhorn Algorithm. Our model achieves unprecedented state-of-the-art results on common VPR benchmarks.

nents of this general pipeline, namely the deep neural backbones for feature extraction and methods for aggregating such features.

For years, ResNet-based neural networks have been the predominant backbones for feature extraction [4, 23, 45]. Recently, given the success of Vision Transformer (ViT) for different computer vision tasks [17, 21, 30, 33], some methods have introduced ViT in the field of VPR [58, 65]. AnyLoc [28] proposed to leverage foundation models, using DINOv2 [41] as a feature extractor for VPR. However, AnyLoc uses DINOv2 'as is', while we show in this paper that fine-tuning the model for VPR brings a significant increase in performance.

Regarding aggregation, NetVLAD [4], the learned counterpart to the traditional handcrafted VLAD [26], is among the most popular choices. Alternative methods include pooling layers like GeM [45] or learned global aggregation, like the recent MixVPR [2]. In this paper, we propose optimal transport aggregation, setting a new state of the art in VPR.

As a summary, in this work, we present a single-stage approach to VPR that obtains state-of-the-art results in the most common benchmarks. To achieve this, we present two key contributions:

- First, we propose SALAD (Sinkhorn Algorithm for Locally Aggregated Descriptors), a reformulation of the feature-to-cluster assignment problem through the lens of optimal transport, allowing more effective distribution of local features into the global descriptor bins. To further improve the discriminative power of the aggregated descriptor, we let the network discard uninformative features by introducing a 'dustbin' mechanism.
- Secondly, we integrate the representational power of foundation models into VPR, using DINOv2 as the backbone for feature extraction. Unlike previous approaches that utilized DINOv2 in its pre-trained form, our method involves fine-tuning the model specifically for the task. This fine-tuning process converges extremely fast, in just four epochs, and allows DINOv2 to capture more relevant and distinctive features pertinent to place recognition tasks.

The fusion of these two novel components results in DINOv2 SALAD, which can be efficiently trained in less than one hour and sets unprecedented recall in VPR benchmarks, with 75.0% Recall@1 in MSLS Challenge and 76.0% in Nordland. All of this with a single-stage pipeline, without requiring expensive post-processing steps and with an inference speed of less than 3 ms per image.

2. Related Work

The significant research efforts on VPR have been exhaustively compiled in a number of surveys and tutorials over the years [19, 36, 37, 49, 64]. Current research addresses a wide variety of topics, such as novel loss functions [5, 31], image sequences [20, 60], extreme viewpoint changes [32] or text features [24]. In this section, we focus on work related to feature extraction and aggregation, as there lie our contributions.

Early approaches to VPR used either aggregations of handcrafted local features [3, 15, 26] or global descriptors [39, 53]. In both cases, geometric [18] and temporal [18, 38] consistency was sometimes enforced for enhanced performance. With the emergence of deep neural networks, features pre-trained for recognition tasks, without fine-tuning, showed a significant performance boost over handcrafted ones [54]. However, training or fine-tuning specifically for VPR tasks using contrastive or triplet losses [40] offers an additional improvement and is standard nowadays.

NetVLAD [4] is the most popular architecture explicitly designed for VPR, mimicking the VLAD aggregation [26] but jointly learning from data both convolutional features and cluster centroids. Radenović et al. [45] proposed the Generalized Mean Pooling (GeM) to aggregate feature activations, also a popular baseline due to its simplicity and competitive performance. In addition to these, several other alternatives have been proposed in the literature. For example, Teichmann et al. [56] aggregates regions instead of local features. Recently, MixVPR [2] has presented the best results in the literature by combining deep features with a MLP layer.

A notable trend in VPR has been the adoption of a two-stage approach to enhance retrieval accuracy [11, 23, 47, 50, 55, 65]. After a first stage with any of the methods presented in the previous paragraph, the top retrieved candidates are re-ranked attending to the un-aggregated local features, either assessing the geometric consistency to the query image or predicting their similarity. This re-ranking stage adds a considerable overhead, which is why it is only applied to a few candidates, but generally improves the performance. Re-ranking is out of the scope of our research but, notably, we outperform all baselines that employ re-ranking even if our model does not include such stage (and hence it is substantially faster).

Optimal transport has found a significant number of applications in graphics and computer vision [8]. Specifically, related to our research, it has been used for image retrieval [43], image matching [61] and feature matching [48, 52]. Recently, Zhang et al. [63] used optimal transport at the re-ranking stage in a retrieval pipeline. However, ours is the first work that proposes the formulation of local feature aggregation from an optimal transport perspective.

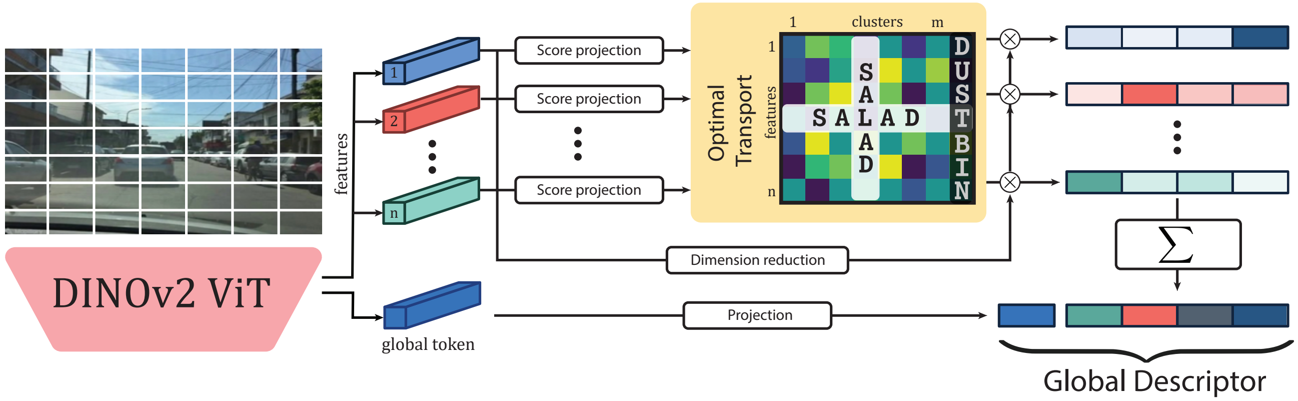

Figure 2. **Overview of our method**. First, the DINOv2 backbone extracts local features and a global token from an input image. Then, a small MLP, score projection, computes a score matrix for feature-to-cluster and dustbin relationships. The optimal transport module uses the Sinkhorn algorithm to transform this matrix into an assignment, and subsequently, dimensionality-reduced features are aggregated into the final descriptor based on this assignment and concatenated with the global token.

3. Method

DINOv2 SALAD is based on NetVLAD, but we propose to use and fine-tune the DINOv2 backbone (Sec. 3.1) and propose a novel module (SALAD) for the assignment (Sec. 3.2) and aggregation (Sec. 3.3) of features.

3.1. Local Feature Extraction

Effective local feature extraction lies in striking a balance: features must be robust enough to withstand substantial changes in appearance, such as those between seasons or from day to night, yet they should retain sufficient information on local structure to enable accurate matching.

Inspired by the success of ViT architectures in many computer vision tasks and by AnyLoc [28], that leverages the exceptional representational capabilities of foundation models [7], we adopt DINOv2 [41] as our backbone. However, differently from AnyLoc, we use a supervised pipeline and include the backbone in the end-to-end training for the specific task, yielding improved performance.

DINOv2 adopts a ViT architecture that initially divides an input image $\mathbf{I} \in \mathbb{R}^{h \times w \times c}$ into $p \times p \times c$ patches, with $p = 14$. These patches are sequentially projected with transformer blocks, resulting in the output tokens $\{\mathbf{t}_1, \ldots, \mathbf{t}_n, \mathbf{t}_{n+1}\}, \mathbf{t}_i \in \mathbb{R}^d$, where $n = hw/p^2$ is the number of input patches and there is an additional global token $\mathbf{t}_{n+1}$ that aggregates class information. Although the DINOv2's authors reported that fine-tuning the model only brings dim improvements, we found that at least for VPR there are substantial gains in selectively unfreezing and training the last blocks of the encoder.

3.2. Assignment

In NetVLAD, a global descriptor is formed by assigning a set of features to a set of clusters, $\{C_1, \ldots, C_j, \ldots, C_m\}$, and then aggregating all features that belong to each cluster. For the assignment, NetVLAD computes a score matrix $\mathbf{S} \in \mathbb{R}_{>0}^{n \times m}$, where the element in its i^{th} row and j^{th} column, $s_{i,j} \in \mathbb{R}_{>0}$, represents the cost of assigning a feature to a cluster C_j. In other words, $\mathbf{S}$ quantifies the affinity of each feature to each clusters. While SALAD draws inspiration from NetVLAD, we identify several crucial aspects in their assignment and propose alternatives to address these.

Reduce assignment priors. When building the score matrix $\mathbf{S}$, NetVLAD introduces certain priors. Specifically, it initializes the linear layer that computes $\mathbf{S}$ with centroids derived from k-means. While this may accelerate the training, it introduces inductive bias and potentially makes the model more susceptible to local minima. In contrast, we propose to learn each row $\mathbf{s}_i$ of the score matrix from scratch with two fully connected layers initialized randomly:

$$\mathbf{s}_i = \mathbf{W}_{s_2}(\sigma(\mathbf{W}_{s_1}(\mathbf{t}_i) + \mathbf{b}_{s_1})) + \mathbf{b}_{s_2} \qquad (1)$$

where $\mathbf{W}_{s_1}$, $\mathbf{W}_{s_2}$ and $\mathbf{b}_{s_1}$, $\mathbf{b}_{s_2}$ are the weights and biases of the layers, and σ is a non-linear activation function.

Discard uninformative features. Some features, such as those representing the sky, might contain negligible information for VPR. NetVLAD does not account for this, and the contribution of all features is preserved in the final descriptor. Contrary, we follow recent works on keypoint matching and introduce a 'dustbin' where non-informative features are assigned to. For that, we augment the score matrix, from $\mathbf{S}$ to $\bar{\mathbf{S}} = [\mathbf{S}, \ \bar{\mathbf{s}}_{i,m+1}] \in \mathbb{R}_{>0}^{n \times m+1}$, by appending the column $\bar{\mathbf{s}}_{i,m+1}$ representing the feature-to-dustbin relation. As in SuperGlue [48], this score is modeled with a single learnable parameter $z \in \mathbb{R}$:

$$\bar{\mathbf{s}}_{i,m+1} = z\mathbf{1}_n \tag{2}$$

being $\mathbf{1}_n = [1, \dots, 1]^\top \in \mathbb{R}^n$ a n-dimensional vector of ones.

Optimal assignment. The original NetVLAD assignment computes a per-row softmax over $\mathbf{S}$ to obtain the distribution of each feature's mass across the clusters. However, this approach only considers the feature-to-cluster relationship and overlooks the reverse –the cluster-to-feature relation. For this reason, we reformulate the assignment as an optimal transport problem where the features' mass, $\boldsymbol{\mu} = \mathbf{1}_n$, must be effectively distributed among the clusters or the 'dustbin', $\boldsymbol{\kappa} = [\mathbf{1}_m^\top, n - m]^\top$. We follow SuperGlue [48] and use the Sinkhorn Algorithm [16, 51] to obtain the assignment $\bar{\mathbf{P}} \in \mathbb{R}^{n \times (m+1)}$ such that

$$\bar{\mathbf{P}}\mathbf{1}_{m+1} = \boldsymbol{\mu} \quad \text{and} \quad \bar{\mathbf{P}}^\top \mathbf{1}_n = \boldsymbol{\kappa}. \tag{3}$$

This algorithm finds the optimal transport assignment between distributions $\boldsymbol{\mu}$ and $\boldsymbol{\kappa}$ iteratively normalizing rows and columns from $\exp\left(\bar{\mathbf{S}}\right)$. Finally, we drop the dustbin column to obtain the assignment $\mathbf{P} = \left[\mathbf{p}_{*,1}, \dots, \mathbf{p}_{*,m}\right]$, where $\mathbf{p}_{*,j}$ stands for the j^{th} column of $\mathbf{P}$.

3.3. Aggregation

Once the feature assignment in our SALAD framework is computed as detailed in Sec. 3.2, we focus on the aggregation of these assigned features to form the final global descriptor. The aggregation process in NetVLAD involves combining all features assigned to each cluster C_j. However, we introduce three variations:

Dimensionality reduction. To efficiently manage the final descriptor size, we first reduce the dimensionality of the tokens from $\mathbb{R}^d$ to $\mathbb{R}^l$. This is achieved by processing the features through two fully connected layers, precisely adjusting the size of the feature vectors while retaining the essential information from the task.

$$\mathbf{f}_i = \mathbf{W}_{f_2}(\sigma(\mathbf{W}_{f_1}(\mathbf{t}_i) + \mathbf{b}_{f_1})) + \mathbf{b}_{f_1} \tag{4}$$

Aggregation. Based on the assignment matrix derived using the Sinkhorn Algorithm, each feature is aggregated into its assigned cluster. Differently from NetVLAD, we do not subtract the centroids to get the residuals. We directly aggregate these features with a summation, reducing the incorporated priors about the aggregation. Viewing the resulting VLAD vector as a matrix $\mathbf{V} \in \mathbb{R}^{m \times l}$, each element $V_{j,k} \in \mathbb{R}$ is computed as follows:

$$V_{j,k} = \sum_{i=1}^{n} P_{i,k} \cdot f_{i,k} \tag{5}$$

where $f_{i,k}$ corresponds to the k^{th} dimension of $\mathbf{f}_i$, with $k \in \{1, \dots, l\}$.

Global token. To include global information about the scene not easily incorporated into local features, we also incorporate a scene descriptor g computed as:

$$\mathbf{g} = \mathbf{W}_{g_2}(\sigma(\mathbf{W}_{g_1}(\mathbf{t}_{n+1}) + \mathbf{b}_{g_1})) + \mathbf{b}_{g_1} \tag{6}$$

where $\mathbf{t}_{n+1}$ is the global token from DINOv2. We then concatenate $\mathbf{g}$ with $\mathbf{V}$ flattened. Following NetVLAD, we do an L2 intra-normalization and an entire L2 normalization of this vector, which yields the final global descriptor.

4. Experiments

To rigorously evaluate the effectiveness of our proposed contributions, we conducted exhaustive experiments following standard evaluation protocols.

4.1. Implementation Details

We ground our training and evaluation setups on the publicly provided framework by MixVPR[1].

For the **architecture**, we opt for a pretrained DINOv2-B backbone, targeting a balance between computational efficiency and representational capacity. We only fine-tune the final 4 layers of the encoder, which significantly enhances the performance without markedly increasing training time. For the fully connected layers, the weights of the hidden layers $\mathbf{W}_{s_1}$, $\mathbf{W}_{f_1}$ and $\mathbf{W}_{g_1}$ have 512 neurons and use ReLU for the activation function σ. To optimize feature handling, we employ a dimensionality reduction, compressing feature token dimensions from $d = 768$ to $l = 128$, and the global to 256. We use $m = 64$ clusters, resulting in a global descriptor of size $128 \times 64 + 256$. We also report results with smaller descriptors, with size $512 + 32$ ($m = 15,\ l = 32$), and $2048 + 64$ ($m = 32,\ l = 64$).

We **train** on GSV-Cities [1], a large dataset of urban locations collected from Google Street View. Given the impressive representation power of DINOv2, our pipeline achieves training convergence within just 4 epochs. Using a batch size of 60 places, each represented by 4 images, the training is completed in 30 minutes on a single NVIDIA RTX 3090. We use the multi-similarity loss [59] and AdamW [35] for the optimization, with an initial learning rate set to 6e$-$5. To ensure an effective learning rate, we linearly decay the initial rate at every iteration so at the end of the training is 20% of the initial value. We use a dropout rate of 0.3 on the score projection and dimensionality reduction neurons. As our model is agnostic to the image input size (as long as it can be divided in 14×14 patches), we evaluate on images of size 322×322 but train on 224×224 to speedup training time.

[1] https://github.com/amaralibey/MixVPR

Method	Desc. size	Latency (ms)	MSLS Challenge		MSLS Val		NordLand		Pitts250k-test		SPED	
			R@1	R@5	R@1	R@5	R@1	R@5	R@1	R@5	R@1	R@5
NetVLAD [4]	32768	1.41	35.1	47.4	82.6	89.6	32.6	47.1	90.5	96.2	78.7	88.3
GeM [45]†	1024	1.14	49.7	64.2	78.2	86.6	21.6	37.3	87.0	94.4	66.7	83.4
Conv-AP [1]	8192	1.22	54.2	66.6	83.1	90.3	42.7	58.9	92.9	97.7	79.2	88.6
CosPlace [5]	2048	2.59	67.2	78.0	87.4	93.0	44.2	59.7	92.1	97.5	80.1	89.6
MixVPR [2]	4096	1.37	64.0	75.9	88.0	92.7	58.4	74.6	94.6	98.3	85.2	92.1
EigenPlaces [6]	2048	2.65	67.4	77.1	89.3	93.7	54.4	68.8	94.1	98.0	69.9	82.9
DINOv2 SALAD	512 + 32	2.33	70.8	83.6	89.3	94.9	61.2	78.9	93.0	97.4	88.5	94.7
DINOv2 SALAD	2048 + 64	2.35	73.7	85.9	90.5	95.4	70.4	85.7	94.8	98.3	89.5	94.9
DINOv2 SALAD	8192 + 256	2.41	**75.0**	**88.8**	**92.2**	**96.4**	**76.0**	**89.2**	**95.1**	**98.5**	**92.1**	**96.2**

Table 1. **Comparison against single-stage baselines.** We compare DINOv2 SALAD against two popular baselines [4, 45] and the four baselines that show best results in recent literature [1, 2, 5, 6]. Our slim version already obtains state-of-the-art results in all metrics. Our full model outperforms all previous results by a significant margin. Note, in particular, the large improvement in the most challenging benchmarks, MSLS Challenge and NordLand. † We reproduced GeM results training during 80 epochs following MixVPR training pipeline.

Method	Desc. size		Memory (GB)	Latency (ms)		MSLS Challenge			MSLS Val		
	Global	Local		Retrieval	Reranking	R@1	R@5	R@10	R@1	R@5	R@10
Patch-NetVLAD [23]	4096	2826 × 4096	908.30	9.55	8377.17	48.1	57.6	60.5	79.5	86.2	87.7
TransVPR [58]	**256**	1200 × 256	22.72	6.27	1757.70	63.9	74.0	77.5	86.8	91.2	92.4
R2Former [65]	**256**	500 × 131	4.7	8.88	202.37	73.0	85.9	88.8	89.7	95.0	96.2
DINOv2 SALAD (ours)	8192 + 256	**0.0**	**0.63**	**2.41**	**0.0**	**75.0**	**88.8**	**91.3**	**92.2**	**96.4**	**97.0**

Table 2. **Comparison against baselines with re-ranking.** We compare our single-stage DINOv2 SALAD with methods that perform a re-ranking stage to improve performance. Without using re-ranking, our DINOv2 SALAD outperforms all other methods while being orders of magnitude faster and more memory-efficient. Latency metrics obtained from [65] using a RTX A5000. Latency for DINOv2 SALAD was computed using a RTX 3090. Memory footprint is calculated on the MSLS Val dataset, which includes around $18,000$ images.

Method	Desc. size	SF-XL Test v1	SF-XL Test v2
CosPlace [5]	2048	76.4	88.8
EigenPlaces [6]	2048	84.1	90.8
DINOv2 SALAD	8192 + 256	**88.6**	**94.8**

Table 3. **Results on SF-XL. (R@1)** Our DINOv2 SALAD achieves unprecedented results on SF-XL despite never seeing any single image of San Francisco during VPR finetuning.

To **validate** our experiments and select the hyperparameters, we monitored the recall in the Pittsburg30k-test [57]. We observed that, in the long run, most configurations perform similarly, but rapid convergence on a few epochs is more sensitive to the hyperparameters.

4.2. Results

We benchmarked our model against several single-stage baselines, namely NetVLAD [4] and GeM [45] as two representative tradicional baselines, and Conv-AP [1], CosPlace [5], MixVPR [2] and EigenPlaces [6] as the four most recent and best performing baselines in the literature. The evaluation spanned a diverse array of well-established datasets: MSLS Validation and Challenge [60], which are comprised of dashcam images; Pittsburgh250k-test [57], featuring urban scenarios; SPED [14], a collection from surveillance cameras; NordLand, notable for its seasonal variations from images captured from the front of a train traversing Norway; and SF-XL [5], a large urban dataset to evaluate VPR at scale. We use Recall@k (R@k) as the metric for all our experiments, as it is standard in related work. We use evaluation data and code from MixVPR [2], which considers retrieval as correct if an image at less than 25 meters (or two frames for Nordland) from the query is among the top-k predicted candidates.

As shown in Table 1, our model outperforms all previous methods on all datasets and all metrics. Even the smaller $512 + 32$ version already surpasses previous models with bigger descriptors on most datasets. It is worth highlighting the metrics saturation observed in MSLS Val, Pitts250k-test and SPED, and on the other hand the challenging nature of MSLS Challenge and NordLand. The MSLS Challenge dataset, with its diversity, extensive size and closed labels, and NordLand, with its extreme sample similarity and seasonal shifts, emerge then as key benchmarks for assessing VPR performance. Although our DINOv2 SALAD shows a significant improvement on *all* benchmarks, it is precisely in MSLS Challenge and NordLand where we obtain the most substantial recall increases, with $+7.6\%$, $+11.7\%$ and $+17.6\%$, $+14.6\%$ for R@1, R@5 respectively over the second best. For SF-XL, as shown in Table 3, our method also achieves the best results to date. This is remarkable, considering that the previous state of the art was trained on this dataset, whereas our method never used any image of San Francisco when it was fine-tuned.

In Table 2, we compare our DINOv2 SALAD method,

Method	Desc. size	MSLS Challenge			MSLS Val			NordLand			Pitts250k-test			SPED		
		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
ResNet NetVLAD [4]	32768	35.1	47.4	51.7	82.6	89.6	92.0	32.6	47.1	53.3	90.5	96.2	97.4	78.7	88.3	91.4
DINOv2 AnyLoc [28]	49152	42.2	53.5	58.1	68.7	78.2	81.8	16.1	25.4	30.4	87.2	94.4	96.5	85.3	94.4	95.4
ResNet SALAD	8192	57.4	70.8	74.9	83.2	89.5	91.8	33.3	49.6	55.8	91.4	96.9	97.9	75.0	86.7	89.8
ConvNext [34] SALAD	8192	63.9	75.2	80.1	85.5	92.4	94.5	47.8	64.3	70.3	93.9	97.9	98.8	83.5	90.9	92.9
DINOv2 GeM	4096	62.6	78.3	83.0	85.4	93.9	95.0	35.4	52.5	59.6	89.5	96.5	98.0	83.0	92.1	93.9
DINOv2 MixVPR	4096	72.1	85.0	88.3	90.0	95.1	96.0	63.6	80.1	84.6	94.6	98.3	**99.3**	89.8	94.9	96.1
DINOv2 NetVLAD	24576	**75.8**	86.5	89.8	**92.4**	95.9	96.9	71.8	86.5	90.1	**95.6**	**98.7**	**99.3**	90.8	95.7	**96.7**
DINOv2 NetVLAD (dim. red.)	8192	73.3	85.6	88.3	90.1	95.4	96.8	70.1	86.5	90.2	95.4	98.4	99.1	90.6	95.4	**96.7**
DINOv2 SALAD (ours)	8192 + 256	75.0	**88.8**	**91.3**	92.2	**96.4**	**97.0**	**76.0**	**89.2**	**92.0**	95.1	98.5	99.1	**92.1**	**96.2**	96.5

Table 4. **Ablations**. The first two rows correspond to two baselines in the literature [4, 28], the rest to different aggregations appended to DINOv2 including our DINOv2 SALAD. Note that only DINO NetVLAD, with a significantly bigger descriptor size than ours, is able to show competitive results. We outperform all the rest DINOv2 baselines of similar descriptor sizes by a large margin.

which solely operates on a single retrieval stage, against the leading two-stage Visual Place Recognition (VPR) techniques. In this comparison, we include the best performing models in the literature, namely R2Former [65], TransVPR [58], and Patch-NetVLAD [23], which incorporate a re-ranking refinement. Note how our DINOv2 SALAD, despite being orders of magnitude faster and smaller in memory, significantly outperforms all these two-stage methods on all benchmarks. This finding not only highlights the efficiency of our model but also demonstrates the effectiveness of global retrieval using our novel SALAD aggregation. Additionally, considering our method's reliance on local features, we believe that a re-ranking stage could also be applied, potentially increasing our recall metrics but at the price of a higher computational footprint.

4.3. Ablation Studies

Effect of DINOv2. We assess the impact of the DINOv2 backbone and our optimal transport aggregation SALAD separately. For this, we compare with the existing baselines of ResNet NetVLAD or AnyLoc, this last one applying a VLAD on top of a pretrained DINOv2 encoder. We integrate the DINOv2 backbone with various aggregation modules, obtaining a handful of performant techniques that improve their respective previous results. As shown in Table 4, all of these outperform the baselines, even though AnyLoc already uses DINOv2. This validates the DINOv2's integration in end-to-end fine-tuning to refine its capabilities.

Effect of SALAD. Our experiments in Table 4 show that aggregation also matters. Even the recent MixVPR aggregation coupled with DINOv2 does not match the performance of DINOv2 NetVLAD and DINOv2 SALAD. We believe that the DINOv2 backbone is especially suitable for local feature aggregation, as its features work remarkably well in dense visual perception tasks [27, 41, 62]. Although DINOv2 NetVLAD achieves comparable performance to SALAD, it employs a descriptor almost three times as big. Besides, the generalization performance of DINOv2 NetVLAD is limited, as observed in NordLand results. We attribute this to NetVLAD's priors initialization with urban scenarios, which constrain the convergence of the system. In our experiments we also trained a slimmer DINOv2 NetVLAD version, whose features are dimensionally reduced as described in Section 3.3, targetting a final descriptor of roughly the same size as SALAD. In this fairer setup, DINOv2 SALAD clearly outperforms DINOv2 NetVLAD. We also evaluate SALAD on top of ResNet and ConvNext backbones, which improves over baseline ResNet NetVLAD but is significantly worse than using DINOv2. This indicates that SALAD is specially suited for high spatial resolution features, like the ones from DINOv2.

Model	Dim. size	# Params.	Latency (ms)	MSLS Val R@1
S	384	21M	1.30	90.5
B	768	86M	2.41	92.2
L	1024	300M	7.82	92.6
G	1536	1100M	24.93	91.7

Table 5. **DINOv2 configurations and performances.**

Effect of hyperparameters. DINOv2 comes in different sizes that affect the number of parameters, inference speed, and representation capabilities. As shown in Table 5, more parameters do not always result in better performance. Excessively big models might be harder to train or prone to overfitting the training set. From these results, we chose the DINOv2-B backbone, which exhibits a great balance between performance and size and speed. Regarding descriptor size, we observed (Table 1) that changing m and l allows to get slimmer versions with competitive performance. For the number of blocks to train, as shown in Table 6, fine-tuning two or four block report the best results without significant computation overhead.

Effect of SALAD components. In Table 7, we show how different components of our SALAD pipeline affect the final performance. Both the global token, which appends global information not captured in local features, and the dustbin, which helps in distilling the aggregated features, contribute to the performance of SALAD. We also trained a model using a dual-softmax [46] to solve the optimal transport assignment, following LoFTR and Gluestick [42, 52]. Although dual-softmax achieves only slightly worse performance, the Sinkhorn Algorithm is theoretically sound and provides a better acronym to our method.

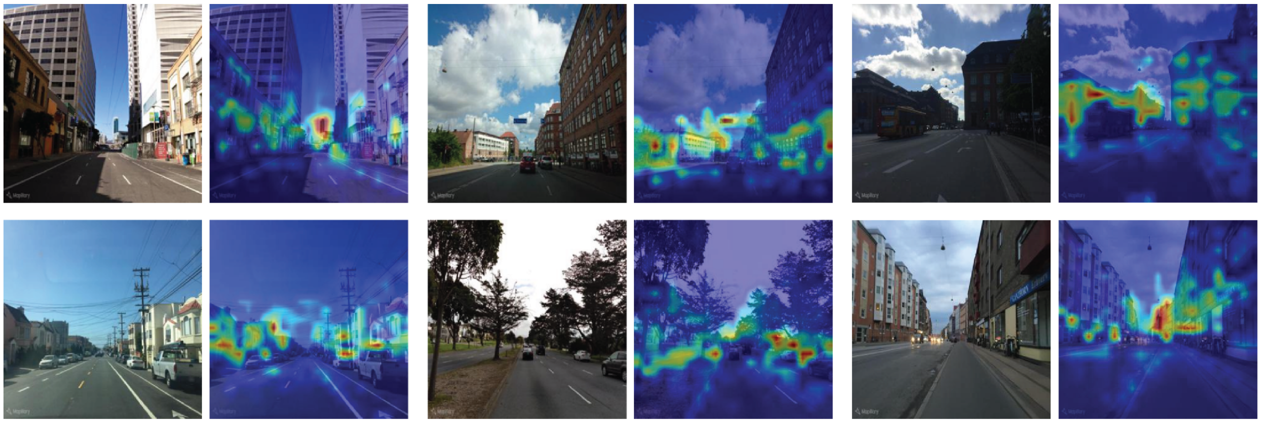

Figure 3. **Heatmap of local features importance**. Left images show the original pictures, their right counterparts represent the weights *not* assigned to the ‘dustbin’. Note how the network learns to discard uninformative regions like skies, roads or dynamic objects, and instead focus on distinctive patterns in buildings and vegetation. We attribute its focus on distant buildings to their invariance to viewpoint change.

Method	MSLS Val		
	R@1	R@5	R@10
DINOv2 SALAD (frozen)	88.5	95.0	96.2
DINOv2 SALAD (train 2 last blocks)	92.0	**96.5**	**97.0**
DINOv2 SALAD (train 4 last blocks)	**92.2**	96.4	**97.0**
DINOv2 SALAD (train 6 last blocks)	91.6	96.2	**97.0**
DINOv2 SALAD (train all blocks)	89.2	95.1	96.1

Table 6. **Fine-tuning different number of DINOv2 blocks.**

Method	MSLS Val		
	R@1	R@5	R@10
DINOv2 SALAD w/o dustbin	91.4	95.8	96.2
DINOv2 SALAD w/o global token	91.8	96.0	96.2
DINOv2 SALAD (Dual Softmax)	91.9	95.7	96.5
DINOv2 SALAD	**92.2**	**96.4**	**97.0**

Table 7. **Ablation study of the SALAD components.**

4.4. Introspective Results

We provide an introspection of our model’s performance through a series of illustrative figures. Figure 3 visualizes the weights that are not assigned to the ‘dustbin’, offering insight into the parts of the input image that the network considers informative. As the ‘dustbin’ assignment is completely learnt by the network, some discarded features might be counter-intuitive. However, we observe that it typically removes dynamic objects and focuses on the most distinctive and invariant parts of the image. In Figure 4, we display the assignment distribution of patches from two different images depicting the same place. It demonstrates the model’s ability to consistently distribute most of the weights into the same bins for patches representing similar regions. Such repeatable and consistent assignment across different images of the same place is crucial for the reliability and performance of the system. Finally, in Figure 5, we showcase various query images alongside their respective top-3 retrievals made by our system. DINOv2 SALAD is able to retrieve correct predictions even under challenging conditions, such as severe changes in illumination or viewpoint.

5. Conclusions and Limitations

In this paper, we have proposed DINOv2 SALAD, a novel model for VPR that outperforms previous baselines by a substantial margin. This achievement is the result of combining two key contributions: a fine-tuned DINOv2 backbone for enhanced feature extraction and our novel SALAD (Sinkhorn Algorithm for Locally Aggregated Descriptors) module for feature aggregation. Our extensive experiments demonstrate the effectiveness of these modules, highlighting the model’s single-stage nature and exceptionally fast training and inference speed.

While our work brings significant improvements in performance, it is not without limitations. Primarily, the adoption of DINOv2 as our backbone results in slower processing speeds compared to ResNet-based methods. Besides, although SALAD is a general aggregation module, its effectiveness is tied to the choice of backbone. It excels with DINOv2, which offers high spatial resolution features, but it is less suited for coarser features. Additionally, in SALAD we use an optimal transport assignment in its simplest form. More sophisticated constraints could improve the resulting assignment, a very relevant aspect for our future work.

Acknowledgments

This work was supported by the Spanish Government (PID2021-127685NB-I00 and TED2021-131150B-I00), the Aragón Government (DGA T45_23R) and the scholarship FPU20/02342.

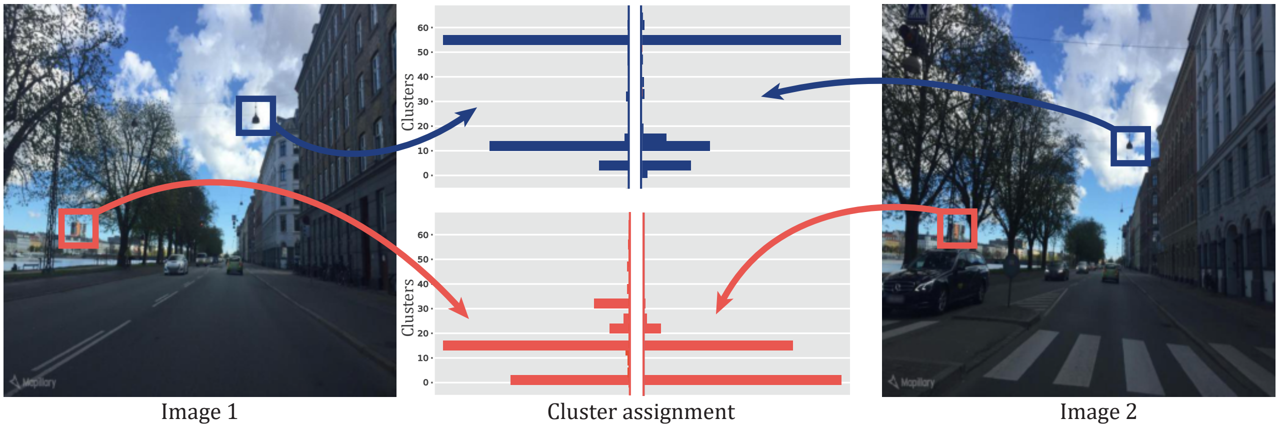

Figure 4. **Illustration of feature-to-cluster assignments.** See at the leftmost and rightmost part of the figure two different views of the same place. Framed by red and blue squares we highlight two corresponding patches in each of the images. The central part of the figure shows the feature-to-cluster assignments for these patches. Note how DINOv2 SALAD correctly assigns the features to the same bins for both views, even with different local texture.

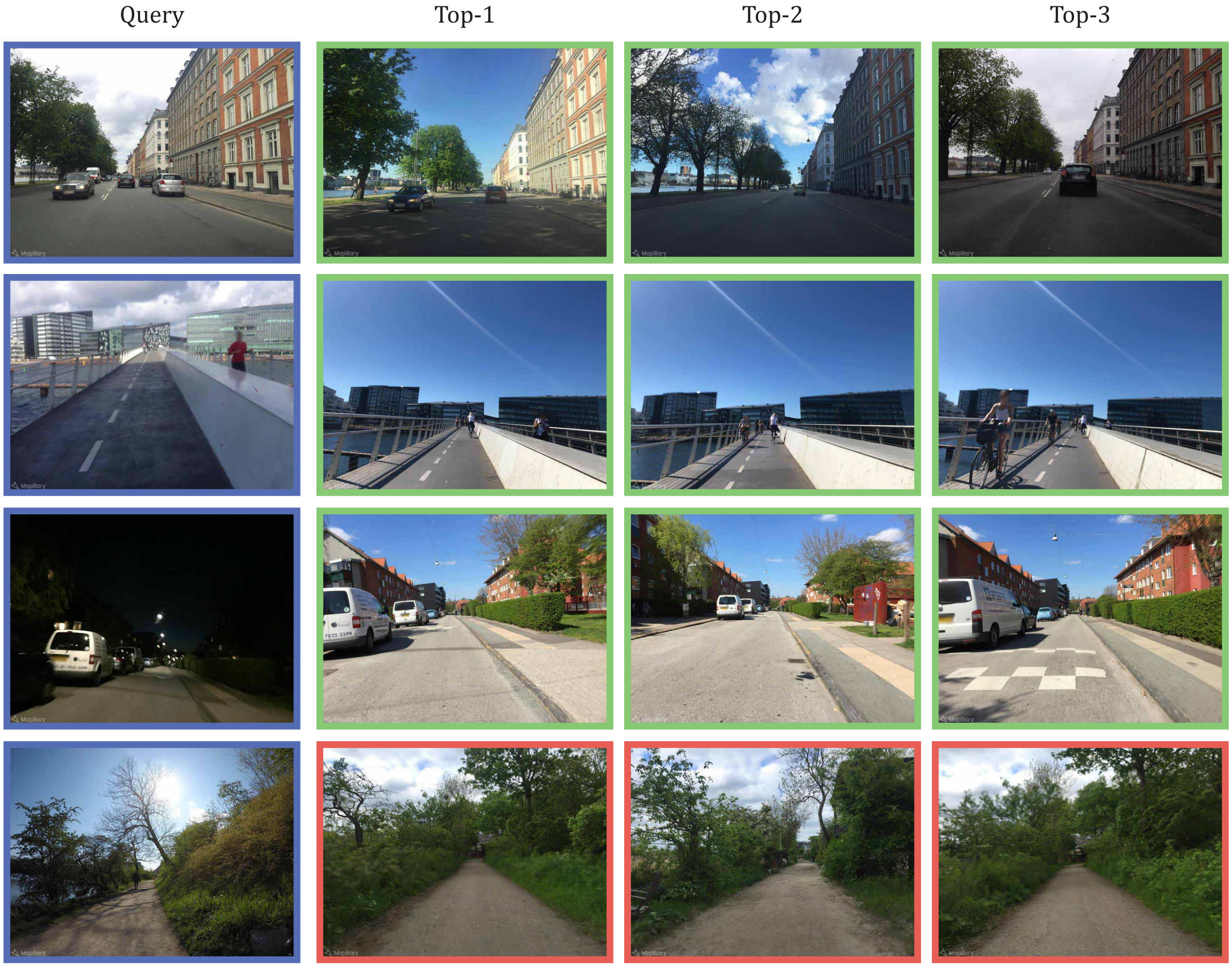

Figure 5. **DINOv2 SALAD qualitative results at MSLS.** The left column shows several queries and the three other ones shows the top-3 candidates retrieved by our DINOv2 SALAD. Candidates are framed in green if they correspond to the same place as the query, and in red if they do not. Note the correct retrievals under seasonal, weather, viewpoint and day-night changes. Note also a challenging failure case in the last row, due to non-discriminative image content.

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] Amar Ali-Bey, Brahim Chaib-Draa, and Philippe 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.

[3] Relja Arandjelovic and Andrew Zisserman. All about vlad. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1578–1585, 2013.

[4] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef 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.

[5] 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.

[6] Gabriele Berton, Gabriele Trivigno, Barbara Caputo, and Carlo 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.

[7] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

[8] Nicolas Bonneel and Julie Digne. A survey of optimal transport for computer graphics and computer vision. In *Computer Graphics Forum*, pages 439–460. Wiley Online Library, 2023.

[9] Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, José Neira, Ian Reid, and John J Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on robotics*, 32(6):1309–1332, 2016.

[10] Carlos Campos, Richard Elvira, Juan J Gómez Rodríguez, José MM Montiel, and Juan D Tardós. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE Transactions on Robotics*, 37(6):1874–1890, 2021.

[11] Bingyi Cao, Andre Araujo, and Jack 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.

[12] 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 *2017 IEEE international conference on robotics and automation (ICRA)*, pages 3223–3230. IEEE, 2017.

[13] 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.

[14] 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.

[15] Mark Cummins and Paul Newman. Fab-map: Probabilistic localization and mapping in the space of appearance. *The International journal of robotics research*, 27(6):647–665, 2008.

[16] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26, 2013.

[17] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

[18] 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.

[19] Sourav Garg, Tobias Fischer, and Michael Milford. Where is your place, visual place recognition? *arXiv preprint arXiv:2103.06443*, 2021.

[20] Sourav Garg, Madhu Vankadari, and Michael Milford. Seqmatchnet: Contrastive learning with sequence matching for place recognition & relocalization. In *Conference on Robot Learning*, pages 429–443. PMLR, 2022.

[21] Kai Han, Yunhe Wang, Hanting Chen, Xinghao Chen, Jianyuan Guo, Zhenhua Liu, Yehui Tang, An Xiao, Chunjing Xu, Yixing Xu, et al. A survey on vision transformer. *IEEE transactions on pattern analysis and machine intelligence*, 45(1):87–110, 2022.

[22] 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.

[23] 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.

[24] Ziyang Hong, Yvan Petillot, David Lane, Yishu Miao, and Sen Wang. Textplace: Visual place recognition and topological localization through reading scene texts. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2861–2870, 2019.

[25] Arnold Irschara, Christopher Zach, Jan-Michael Frahm, and Horst Bischof. From structure-from-motion point clouds to fast location recognition. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2599–2606. IEEE, 2009.

[26] Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez. Aggregating local descriptors into a compact image

representation. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 3304–3311. IEEE, 2010.

[27] Markus Käppeler, Kürsat Petek, Niclas Vödisch, Wolfram Burgard, and Abhinav Valada. Few-shot panoptic segmentation with foundation models. *arXiv preprint arXiv:2309.10726*, 2023.

[28] Nikhil Keetha, Avneesh Mishra, Jay Karhade, Krishna Murthy Jatavallabhula, Sebastian Scherer, Madhava Krishna, and Sourav Garg. Anyloc: Towards universal visual place recognition. *arXiv preprint arXiv:2308.00688*, 2023.

[29] 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.

[30] Youngwan Lee, Jonghee Kim, Jeffrey Willette, and Sung Ju Hwang. Mpvit: Multi-path vision transformer for dense prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7287–7296, 2022.

[31] María Leyva-Vallina, Nicola Strisciuglio, and Nicolai Petkov. Data-efficient large scale place recognition with graded similarity supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23487–23496, 2023.

[32] Tsung-Yi Lin, Yin Cui, Serge Belongie, and James Hays. Learning deep representations for ground-to-aerial geolocalization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5007–5015, 2015.

[33] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, et al. Swin transformer v2: Scaling up capacity and resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12009–12019, 2022.

[34] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11976–11986, 2022.

[35] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[36] Stephanie Lowry, Niko Sünderhauf, Paul Newman, John J Leonard, David Cox, Peter Corke, and Michael J Milford. Visual place recognition: A survey. *ieee transactions on robotics*, 32(1):1–19, 2015.

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

[38] Michael J Milford and Gordon F Wyeth. Seqslam: Visual route-based navigation for sunny summer days and stormy winter nights. In *2012 IEEE international conference on robotics and automation*, pages 1643–1649. IEEE, 2012.

[39] Ana C Murillo, Gautam Singh, Jana Kosecka, and José Jesús Guerrero. Localization in urban environments using a panoramic gist descriptor. *IEEE Transactions on Robotics*, 29(1):146–160, 2012.

[40] Kevin Musgrave, Serge Belongie, and Ser-Nam Lim. A metric learning reality check. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXV 16*, pages 681–699. Springer, 2020.

[41] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[42] Rémi Pautrat, Iago Suárez, Yifan Yu, Marc Pollefeys, and Viktor Larsson. Gluestick: Robust image matching by sticking points and lines together. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9706–9716, 2023.

[43] Ofir Pele and Michael Werman. Fast and robust earth mover's distances. In *2009 IEEE 12th international conference on computer vision*, pages 460–467. IEEE, 2009.

[44] Noé Pion, Martin Humenberger, Gabriela Csurka, Yohann Cabon, and Torsten Sattler. Benchmarking image retrieval for visual localization. In *2020 International Conference on 3D Vision (3DV)*, pages 483–494. IEEE, 2020.

[45] 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.

[46] Ignacio Rocco, Mircea Cimpoi, Relja Arandjelović, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Neighbourhood consensus networks. *Advances in neural information processing systems*, 31, 2018.

[47] 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.

[48] 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.

[49] Stefan Schubert, Peer Neubert, Sourav Garg, Michael Milford, and Tobias Fischer. Visual Place Recognition: A Tutorial. *IEEE Robotics & Automation Magazine*, 2023.

[50] Shihao Shao, Kaifeng Chen, Arjun Karpur, Qinghua Cui, André Araujo, and Bingyi Cao. Global features are all you need for image retrieval and reranking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11036–11046, 2023.

[51] Richard Sinkhorn and Paul Knopp. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics*, 21(2):343–348, 1967.

[52] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8922–8931, 2021.

[53] Niko Sünderhauf and Peter Protzel. Brief-gist-closing the loop by simple means. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1234–1241. IEEE, 2011.

[54] Niko Sünderhauf, Sareh Shirazi, Feras Dayoub, Ben Upcroft, and Michael Milford. On the performance of convnet features for place recognition. In *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 4297–4304. IEEE, 2015.

[55] 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.

[56] Marvin Teichmann, Andre Araujo, Menglong Zhu, and Jack Sim. Detect-to-retrieve: Efficient regional aggregation for image search. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5109–5118, 2019.

[57] Akihiko Torii, Josef Sivic, Tomas Pajdla, and Masatoshi Okutomi. Visual place recognition with repetitive structures. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 883–890, 2013.

[58] 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.

[59] Xun Wang, Xintong Han, Weilin Huang, Dengke Dong, and Matthew R Scott. Multi-similarity loss with general pair weighting for deep metric learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5022–5030, 2019.

[60] Frederik Warburg, Soren Hauberg, Manuel Lopez-Antequera, Pau Gargallo, Yubin Kuang, and Javier Civera. Mapillary street-level sequences: A dataset for lifelong place recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2626–2635, 2020.

[61] Jiankai Xing, Fujun Luan, Ling-Qi Yan, Xuejun Hu, Houde Qian, and Kun Xu. Differentiable rendering using rgbxy derivatives and optimal transport. *ACM Transactions on Graphics (TOG)*, 41(6):1–13, 2022.

[62] Jingfeng Yao, Xinggang Wang, Shusheng Yang, and Baoyuan Wang. Vitmatte: Boosting image matting with pretrained plain vision transformers. *Information Fusion*, 103: 102091, 2024.

[63] Chao Zhang, Stephan Liwicki, and Roberto Cipolla. Beyond the cls token: Image reranking using pretrained vision transformers. In *BMVC*, 2022.

[64] Xiwu Zhang, Lei Wang, and Yan Su. Visual place recognition: A survey from deep learning perspective. *Pattern Recognition*, 113:107760, 2021.

[65] Sijie Zhu, Linjie Yang, Chen Chen, Mubarak Shah, Xiaohui Shen, and Heng 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.