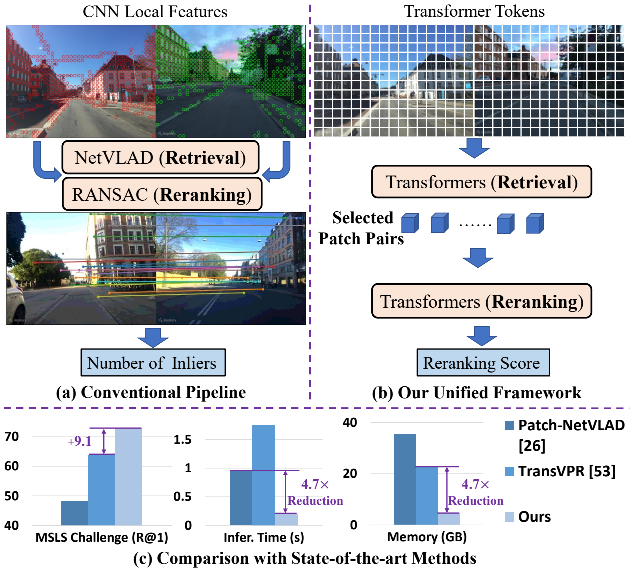
R^2Former: Unified Retrieval and Reranking Transformer for Place Recognition

Sijie Zhu[1,2,†], Linjie Yang[1], Chen Chen[2], Mubarak Shah[2], Xiaohui Shen[1], Heng Wang[1]
[1] ByteDance [2]Center for Research in Computer Vision, University of Central Florida
{sijiezhu,linjie.yang,heng.wang,shenxiaohui.kevin}@bytedance.com, {chen.chen,shah}@crcv.ucf.edu

Abstract

Visual Place Recognition (VPR) estimates the location of query images by matching them with images in a reference database. Conventional methods generally adopt aggregated CNN features for global retrieval and RANSAC-based geometric verification for reranking. However, RANSAC only employs geometric information but ignores other possible information that could be useful for reranking, e.g. *local feature correlations, and attention values. In this paper, we propose a unified place recognition framework that handles both retrieval and reranking with a novel transformer model, named R^2Former. The proposed reranking module takes feature correlation, attention value, and xy coordinates into account, and learns to determine whether the image pair is from the same location. The whole pipeline is end-to-end trainable and the reranking module alone can also be adopted on other CNN or transformer backbones as a generic component. Remarkably, R^2Former significantly outperforms state-of-the-art methods on major VPR datasets with much less inference time and memory consumption. It also achieves the state-of-the-art on the hold-out MSLS challenge set and could serve as a simple yet strong solution for real-world large-scale applications. Experiments also show vision transformer tokens are comparable and sometimes better than CNN local features on local matching. The code is released at* https://github.com/Jeff-Zilence/R2Former.



(a) Conventional Pipeline

(b) Our Unified Framework

(c) Comparison with State-of-the-art Methods

Figure 1. An overview of conventional pipeline and our unified framework. Both our global retrieval and reranking modules are end-to-end trainable transformer layers, which achieve state-of-the-art performance with much less computational cost.

1. Introduction

Visual Place Recognition (VPR) aims to localize query images from unknown locations by matching them with a set of reference images from known locations. It has great potential for robotics [47], navigation [41], autonomous driving [9], augmented reality (AR) applications. Previous works [26, 53] generally formulate VPR as a retrieval problem with two stages, *i.e.* global retrieval, and reranking. The global retrieval stage usually applies aggregation methods (*e.g.* NetVLAD [3] and GeM [43]) on top of CNN (Convolutional Neural Network) to retrieve top candidates from a large reference database. While some works [3, 6] only adopt global retrieval, current state-of-the-art methods [26, 53] conduct reranking (*i.e.* geometric verification with RANSAC [19]) on the top-k (*e.g.* $k = 100$) candidates to further confirm the matches, typically leading to a significant performance boost. *However, geometric information is not the only information that could be useful for reranking, and task-relevant information could be learned with a data-driven module to further boost performance.* Besides, the current reranking process requires a relatively large inference time and memory footprint (typically over 1s and 1MB per image), which cannot scale to real-world applications with large QPS (Queries Per Second) and reference database ($>$ 1M images).

Recently vision transformer [17] has achieved significant performance on a wide range of vision tasks, and has been shown to have a high potential for VPR [53] (Sec. 2).

[†] This work was done during the first author's internship at ByteDance

However, the predominant local features [26, 53] are still based on CNN backbones, due to the built-in feature locality with limited receptive fields. Although vision transformer [17] considers each patch as an input token and naturally encodes local information, the local information might be overwritten with global information by the strong global correlation between all tokens in every layer. *Therefore, it is still unclear how vision transformer tokens perform in local matching as compared to CNN local features in this field.*

In this paper, we integrate the global retrieval and reranking into one unified framework employing only transformers (Fig. 1), abbreviated as R^2Former, which is simple, efficient, and effective. The global retrieval is tackled based on the class token without additional aggregation modules [3, 43] and the other image tokens are adopted as local features. Different from geometric verification which focuses on geometric information between local feature pairs, we feed the correlation between the tokens (local features) of a pair of images, xy coordinates of tokens, and their attention information to transformer modules, so that the module can learn task-relevant information that could be useful for reranking. The global retrieval and reranking parts can be either trained in an end-to-end manner or tuned alternatively with a more stable convergence. The proposed reranking module can also be adopted on other CNN or transformer backbones, and *our comparison shows that vision transformer tokens are comparable to CNN local features in terms of reranking performance (Table 6).*

Without bells and whistles, the proposed method outperforms both retrieval-only and retrieval+reranking state-of-the-art methods on a wide range of VPR datasets. The proposed method follows a very efficient design and applies linear layers for dimension reduction: *i.e.* only 256 and 500×131 for global and local feature dimensions and only 32 for transformer dimension of reranking module, thus is significantly faster ($> 4.7\times$ QPS) with much less ($< 22\%$) memory consumption than previous methods [26, 53]. Both the global and local features are extracted from the same backbone model only once, and the reranking of top-k candidates is finished with only one forward pass by computing the reranking scores of all candidate pairs in parallel within one batch. The reranking speed can be further boosted by parallel computing on multiple GPUs with $> 20\times$ speedup over previous methods [26, 53]. We demonstrate that the proposed reranking module also learns to focus on good local matches like RANSAC [19]. We summarize our contributions as follows:

- A unified retrieval and reranking framework for place recognition employing pure transformers, which demonstrates that vision transformer tokens are comparable and sometimes better than CNN local features in terms of reranking or local matching.
- A novel transformer-based reranking module that learns to attend to the correlation of informative local feature pairs. It can be combined with either CNN or transformer backbones with better performance and efficiency than other reranking methods, *e.g.* RANSAC.
- Extensive experiments showing state-of-the-art performance on a wide range of place recognition datasets with significantly less ($< 22\%$) inference latency and memory consumption than previous reranking-based methods.

2. Related Work

Visual Place Recognition. Visual place recognition (VPR) [2, 21, 38, 39, 47, 58, 60] is traditionally addressed with nearest neighbor search on aggregated [28] hand-crafted features [12, 37]. The current predominant methods [3–6, 26, 53] are based on CNN [27, 46] feature extractors with trainable aggregation layer, *e.g.* NetVLAD [3], CRN [30], or light-weighted pooling layer, *e.g.* GeM [43], R-MAC [24]. Numerous works [7, 20, 22, 23, 34, 54, 55] follow NetVLAD [3] to further improve the global representation for retrieval. Recently, Berton [6] *et al.* introduce a benchmark for VPR and implement a wide range of global-retrieval-based methods in the same framework. CosPlace [4] *et al.* propose an extremely large dataset and formulate the problem differently as classification using orientation information.

In addition to global retrieval, recent state-of-the-art VPR methods [26, 53] apply reranking on the top retrieved candidates. Patch-NetVLAD [26] adopts NetVLAD [3] model as their backbone for global retrieval and applies RANSAC [19] based geometric verification on multi-scale patch descriptors. TransVPR [53] adds multiple transformer layers on the top of CNN backbones to extract both global and local features. An additional attention module is then deployed to select important local features as the input of RANSAC [19]. Other local matching methods from related tasks, *e.g.* SuperGlue [15, 44] and DELG [10], are also evaluated for VPR. However, they both have a lower accuracy with a much slower inference speed [53]. Recently, there are several recent learning-based reranking methods from related tasks, *e.g.* landmark matching [31, 48] and local matching [29]. While they either fail to generalize [31, 48] on VPR (Table 4) or have a very different problem setting [29], *i.e.* computing point-to-point correspondence between two given images without global retrieval. *To summarize, current state-of-the-art VPR methods rely on RANSAC, and methods from related tasks do not generalize well when directly applied to VPR. Our method takes one step forward to integrate retrieval and reranking into one unified VPR framework with pure transformers.*

Vision Transformer. Transformer [52] is first proposed for NLP (Natural Language Processing) tasks as a generic architecture without inductive bias. It is recently introduced to vision tasks as vision transformer [17] (ViT) by simply considering each image patch as a token. The vanilla ViT

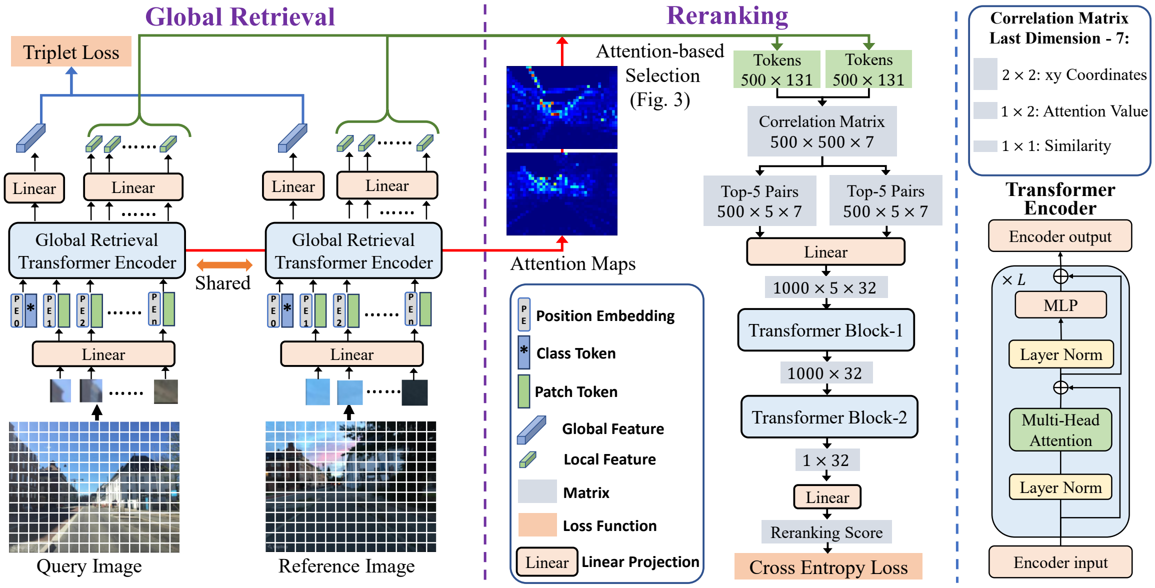


Figure 2. An overview of the proposed framework. Both the global and local features are extracted as the class token and patch tokens, followed by dimension reduction using linear layers. The important local features are then selected based on the attention map (Fig. 3) and fed into the reranking module. Both the global retrieval and reranking modules consist of only transformer and linear layers.

[17] requires large-scale training datasets (*e.g.* ImageNet-21k [14]) to achieve comparable results as CNN [27], and Deit [51] proposes a data-efficient training strategy for ViT which outperforms CNN on standard ImageNet-1k. The recent VG benchmark [6] adopts vanilla ViT without modifying the input resolution which could be suboptimal, while it already shows competitive performance on global retrieval. TransVPR [53] adopts multiple transformer layers, while the backbone feature extractor is still based on CNN. *We hypothesize the power of vision transformer is not fully exploited for VPR and it is interesting to know how vision transformer tokens perform on reranking/local matching as compared with the predominant CNN local feature.*

3. R^2Former

We first formulate the problem and training objective of the proposed method in Sec. 3.1. Then we describe global retrieval and reranking stages in Sec. 3.2 and Sec. 3.3 respectively. Fig. 2 shows an overview of R^2Former.

3.1. Problem Formulation and Training Objective

The proposed framework consists of two stages, *i.e.* global retrieval, and reranking. Given a set of query images $\{I_q\}$ and reference images $\{I_r\}$, the objective of global retrieval is to learn an embedding space in which each query I_q is close to its corresponding positive reference image I_r. During training, reference images from the same location as the query image are defined as positive samples, and the typical threshold is set as 10 meters. We follow the common practice of previous works [3,6] to find the nearest reference image for each query in the embedding space as the final positive sample. Other reference images with distances greater than 25 meters are considered as negative samples. Partial negative mining [6] is conducted to select the hardest negative samples from a random subset. We denote the global embedding features of query, positive samples, and negative samples as E_q, E_p, E_n and the global retrieval loss is trained with margin triplet loss:

$$\mathcal{L}_{retrieval} = max(||E_q - E_p||^2 - ||E_q - E_n||^2 + m, 0). \quad (1)$$

Here $||.||^2$ denotes squared L2 norm and m is the margin.

The reranking module takes the local features of two images as input and generates two-logit scores $\mathbb{L}$ as the output of a binary classification, representing the likelihoods for True or False matches. We feed both positive and negative query-reference pairs to the reranking module during training and the cross entropy (CE) loss is formulated as:

$$\mathcal{L}_{reranking} = CE(Softmax(\mathbb{L}_{qr}), \mathbb{I}_{qr}). \quad (2)$$

$\mathbb{L}$ and $\mathbb{I}$ denote the logits scores and ground-truth labels for the query-reference pairs. Although the partial negative mining [6] is shown to perform better than full negative mining for global retrieval in [6], the objective of the reranking module is to distinguish top-k retrieved candidates which are harder than partial negative samples [6]. To make sure the reranking module sees top-k hardest samples during training, we first freeze the global retrieval module and train the reranking module with randomly selected negative samples from the top-k hardest samples of the full database. The retrieval and reranking modules are then fine-tuned together with partial negative mining for better performance. Details are provided in Sec. 4.2.

3.2. Global Retrieval Module

We describe the components of our vision transformer backbone for VPR and how the final global/local features are generated. As shown in Fig. 2, the query and reference images share the same backbone transformer encoder and there is no additional aggregation [3, 43] or key-point [15] extraction module for global and local feature generation. In other words, all the features and intermediate data are directly generated using only transformers.

ViT for Place Recognition. An input image $I \in \mathbb{R}^{h \times w \times c}$ is first divided into small $p \times p$ patches ($p = 16$ by default) and converted into a number of tokens $T \in \mathbb{R}^{n \times d}$ by linear projection as the input of transformer encoders. Here n, d denote the number and dimension of tokens, and h, w, c denote the height, width, and number of channels in input images. In addition to the n patch tokens, ViT [17] adds an additional learnable class token to aggregate classification information from each layer, which serves as a simple alternative for feature aggregation. Then we adopt the learnable position embedding $PE \in \mathbb{R}^{(n+1) \times d}$ from ViT [17] and add it to each token to provide positional information. *Different from previous work [6] using the fixed training resolution of ViT [17], we conduct 2D interpolation on the positional embedding so that the input resolution can be arbitrary and we use the most widely used resolution* (640×480) *for our method.*

Global Attention. On the bottom right of Fig. 2, we provide the detailed inner architecture of the transformer encoder, which has L cascaded basic transformer [52] layers. The key component is the multi-head attention module, which adopts three linear projections to convert the input token into query, key and value, denoted as Q, K, V with dimension d. The basic attention is computed as $softmax(QK^{\top}/d)V$ ($\top$ means transpose), and a multi-head attention module performs this attention procedure in parallel for multiple heads with their final outputs concatenated. *The multi-head attention module can model stronger global correlation than CNN with a limited receptive field, which helps information aggregation for global representation. However, it might reduce the locality of each token which could be harmful to local matching, we demonstrate that transformer tokens perform well as local features compared to CNN as shown in Table 6.*

Dimension Reduction. We feed the output class token from the last transformer layer into a linear layer to generate the global feature with only 256 dimensions. The local features are generated by applying a linear head on the output patch tokens of the penultimate transformer layer and the dimension is reduced to 128 to achieve a small memory consumption. Both global and local features are L2 normalized after reduction. *The dimension reduction ensures a much smaller (Table 3) total feature dimensions than previous reranking-based methods.*

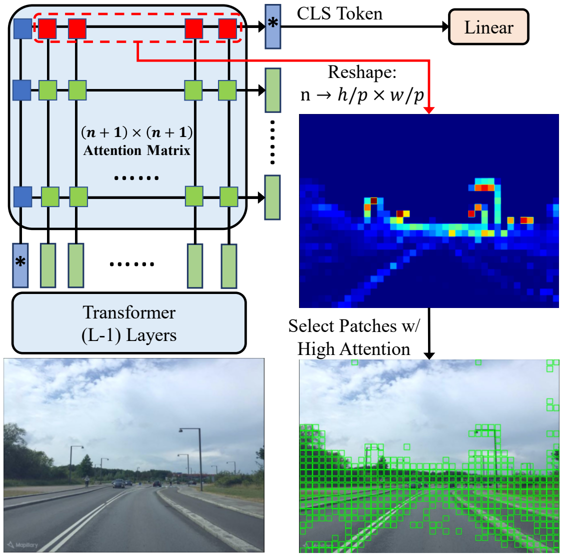


Figure 3. Illustration of attention generation and token selection.

3.3. Reranking Transformer Module

We describe the attention-based local feature selection and the workflow inside our reranking transformer module.

Attention-based Selection. For large-scale scenarios, local features occupy the majority of memory, thus reducing the number of local features is important for real-world deployment. Previous works [3, 26, 53] usually use 640×480 as image resolution, resulting in 1200 patch tokens. However, only a small portion of them are informative to determine the location of the image, *e.g.* buildings, trees, roads, *etc*. Previous works either leverage all the local features from multiple scales [26] or adopt additional modules [53] to generate attention maps, which filter local features with a certain threshold during local matching. *On the contrary, we leverage the natural attention map within our transformer module and only save a fixed number of local features* (e.g. 500) *with top attention values, resulting in a much lower memory cost and simpler extraction pipeline.*

The $(n+1) \times (n+1)$ attention matrix of the last transformer layer in Fig. 3 is formulated as $softmax(QK^{\top}/d)$ in Sec. 3.2. It represents the contribution from each input token to each output token. Since the output class (CLS) token is the only one that connects to the global embedding feature, the CLS token output channel of the matrix ($(n+1)$-dimensional vector for n patches and CLS token) represents the contribution from input tokens to the global feature, which corresponds to the importance of each patch. We reshape the n-dimensional vector corresponding to n patches as $h/p \times w/p$ attention map to sort all the tokens and we select the top-500 tokens that are likely to cover the informative regions (bottom right of Fig. 3). The attention value and x,y coordinates are saved along with each local feature, resulting in $128 + 3 = 131$ dimensions.

Correlation-based Reranking. RANSAC-based [19] geometric verification only leverages the top-1 matched pairs for all local features, but ignores other important information that could be useful for reranking, *i.e.* the correlation, and attention value. Local feature pairs with higher correlation/similarity have higher probabilities to be correct local matches, and an image pair with more correct local matches is more likely to represent the same place. Also, high attention values of the local patches could indicate the importance of the local feature pairs. Therefore, our reranking module is designed to maintain the correlation, attention, and positional information for all the local feature pairs in a correlation matrix, resulting in 7 dimensions (Fig. 2) denoted as (x, y, A, x', y', A', S). x, y, x', y' denote the coordinates of the two patches in the query and reference images. A, A' denotes the attention values of the two patches. S is the cosine similarity between the 128-dimensional local features. Since each image has 500 selected features/tokens, there are 500×500 pairs, resulting in a $500 \times 500 \times 7$ correlation matrix. *The correlation matrix contains the major information for all the feature pairs and allows the model to learn whatever is useful to determine whether the two images are from the same location.*

To reduce the computation, we select the 5 nearest neighbors of each token in the feature space to produce two $500 \times 5 \times 7$ matrices, as the other feature pairs with large distances are likely to be wrong matches. They are concatenated together and fed to a linear layer, resulting in a $1000 \times 5 \times 32$ matrix. *We then leverage the strong global correlation modeling of transformer [52] to aggregate the large matrix as a reranking score to determine whether the input pair is a correct match.* First, we adopt "Transformer Block-1" to extract important information from the top-5 pairs as one output class token. Then "Transformer Block-2" extracts aggregates the information from the 1000 tokens as a single 32-dimension vector (the class token). The two transformer blocks are multiple transformer layers with linear projection and standard Sinusoidal positional embedding [52]. Finally, the vector is converted into 2 channels by a linear head as a binary classification (*i.e.* True vs False).

4. Experiment

4.1. Datasets and Evaluation Metrics

We train our model on MSLS (Mapillary Street-Level Sequences) [55] dataset, which covers a wide range of real-world scenarios for VPR, *e.g.* different cities, viewpoint variation, day/night, and season changes. The performance in urban scenarios (*e.g.* Pitts30k [50], Tokyo24/7 [49]) can be further improved by finetuning on Pitts30k. For evaluation, we use standard train/val/test split [6, 26, 53, 55] on major datasets, including MSLS Val [55], MSLS Challenge [55], Pitts30k [50], Tokyo 24/7 [49], R-SF (Revisited San Francisco) [6, 11, 33, 45], St Lucia [40]. The MSLS Challenge [55] is a hold-out set whose labels are not released, but researchers can submit the predictions on their challenge server to get the performance. For the other datasets with location labels, we follow previous works [6, 26, 53] to use 25 meters as the threshold for correct localization and report recall@k (k=1,5,10) as evaluation metrics. We also report detailed computational costs, including inference time for extraction/retrieval/matching, feature dimensions, GFLOPs, and memory footprint (Table 3).

4.2. Implementation Details

The proposed method is implemented using PyTorch [42]. The models are trained on 8 Tesla-V100 GPUs. All images are resized to 640×480 for training and evaluation. The reranking is conducted on top-100 candidates and we set margin $m = 0.1$. We use ViT-S [17] (12 layers with 384 dimensions) architecture as our default backbone and the module is initialized with off-the-shelf pre-trained weights [51] on ImageNet-1k [14]. The transformers in the reranking module use a small dimension of 32 with only 2 and 6 layers for Transformer Block-1 and Block-2 respectively, thus resulting in a very small computational cost.

The global retrieval and the reranking module can be trained jointly in an end-to-end manner, but we train them separately (Sec. 3.1) by default to achieve a stable convergence and better accuracy. The global retrieval module is trained following common practice [6]. The reranking module is trained from scratch following the standard pipeline of training transformers, *i.e.* AdamW [36] optimizer with cosine learning rate schedule [35]. The initial learning rate is 0.0005 with 64 triplets per batch. Each triplet only samples one hard negative reference image which is randomly selected from the top-100 hardest samples from the MSLS training set of the corresponding query. Given that the global embedding features do not change during this stage, the global hard negative mining only needs to be conducted once and can be efficiently computed on GPU in 3 hours. We precompute the hard negative list of all queries for the training of the reranking module, which is more efficient than the full mining implementation of the VG benchmark [6]. The reranking module is trained for 50 epochs and we select the model with the highest recall@5 on the validation set. The global retrieval and reranking module are then finetuned together with partial negative mining [6]. More details are included in **supplementary material**.

4.3. Comparison with State-of-the-art

In this section, we compare the proposed method with previous state-of-the-art methods, including NetVLAD [3], SFRS [23], SP-SuperGlue [15, 44], Patch-NetVLAD [26], and TransVPR [53]. We further compare our method with the best configurations in the recent VG benchmark [6].

	MSLS Val [55]			MSLS Challenge [55]			Pitts30k [50]			Tokyo 24/7 [49]		
	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
NetVLAD [3]	60.8	74.3	79.5	35.1	47.4	51.7	81.9	91.2	93.7	64.8	78.4	81.6
SFRS [23]	69.2	80.3	83.1	41.5	52.0	56.3	89.4	94.7	95.9	85.4	91.1	**93.3**
SP-SuperGlue [15,44]	78.1	81.9	84.3	50.6	56.9	58.3	87.2	94.8	**96.4**	88.2	90.2	90.2
Patch-NetVLAD [26]	79.5	86.2	87.7	48.1	57.6	60.5	88.7	94.5	95.9	86.0	88.6	90.5
TransVPR [53]	86.8	91.2	92.4	63.9	74.0	77.5	89.0	94.9	96.2	79.0	82.2	85.1
Ours	**89.7**	**95.0**	**96.2**	**73.0**	**85.9**	**88.8**	**91.1**	**95.2**	96.3	**88.6**	**91.4**	91.7

Table 1. Comparison of our method with previous state-of-the-art results on major VPR datasets. Our model is trained on MSLS and tested on MSLS Val and Challenge set. Our model is further finetuned on Pitts30k for urban scenarios, *i.e.* Pitts30k, Tokyo 24/7.

	Features Dim ↓	GFLOPs ↓	Trained on MSLS - R@1					
			Pitts30k	MSLS Val	MSLS Chall.	Tok. 24/7	R-SF	St Lucia
ResNet101 + GeM [6,43]	1024	86.29	77.2	77.0^{*}	55.5	51.0	46.9	91.6
ResNet101 + NetVLAD [3,6]	65536	86.06	80.8	81.1^{*}	61.5	59.0	56.1	95.1
CCT384 + NetVLAD [6,25]	24576	**18.53**	85.1	83.8^{*}	61.4	70.3	65.9	98.4
Ours w/o Reranking	**256**	25.90	76.3	79.3	56.2	45.7	47.5	94.3
Ours	$65500^{\dagger}$	$48.50^{\dagger}$	**88.4**	**89.7**	**73.0**	**72.7**	**72.1**	**99.7**

Table 2. Comparison following the protocol of the recent VG benchmark [6] on major datasets in terms of R@1. The models are trained on MSLS and directly tested on all datasets w/o finetuning. † The dimensions and GFLOPs are computed by adding the numbers of global retrieval and reranking together, *i.e.* $Dim = 256 + 500 \times (128 + 3)$ and $GFLOPs = 25.90 + 100 \times 0.226$.

	Feature Dim ↓		Latency per Query (ms) ↓			Memory Footprint (GB) ↓	
	Global	Local	Extraction	Retrieval	Reranking	MSLS Val	1M Images
ResNet101 + NetVLAD [3,6]	65536	N/A	9.60	2.33	N/A	4.79	244.14
Patch-NetVLAD-s [26]	512	936 × 512	9.29	0.08	952.85	37.60	1917.29
Patch-NetVLAD-p [26]	4096	2826 × 4096	9.36	0.19	8377.17	908.30	46315.85
TransVPR [53]	**256**	1200 × 256	**6.20**	**0.07**	1757.70	22.72	1158.53
Ours	**256**	**500 × (128+3)**	8.81	**0.07**	**202.37**	**4.79**	**244.01**

Table 3. Comparison of computational cost in terms of feature dimension, latency, and memory footprint. All the methods are measured on MSLS Val ($18,871$ database images) using the same CPU and GPU (RTX A5000). Reranking is conducted on top-100 candidates. "Patch-NetVLAD-p" and "Patch-NetVLAD-s" denote the performance and speed-oriented versions.

We also provide a detailed comparison with state-of-the-art methods [6,26,53] on computational cost. Other works [1,16,18,57,59,62] on related tasks with different settings are not included.

As shown in Table 1, the proposed method significantly outperforms state-of-the-art methods on MSLS Val and Challenge set with absolute R@1 **improvement of** 2.9% **and** 9.1% **respectively**. Benefiting from large-scale training, our method also achieves state-of-the-art R@1 on Pitts30k and Tokyo24/7. Small-scale dataset [50] with weak supervision is not enough to train a data-driven reranking module, and training on one large-scale dataset with good generalization on all datasets is more desirable for real-world deployment. (Details in supp. material.)

In Table 2, we select the best-performing models in the VG benchmark [6] and compare them with our method on major datasets. Eynsham [13] dataset is not included because it has a different camera type (gray-scale) and resolution. $*$ denotes reproduced results using the official MSLS [55] validation code and pre-trained models [6]. Table 2 shows that our model trained on MSLS generalizes well on all the other datasets and significantly outperforms the best models in the VG benchmark [6] with comparable computational cost.

In Table 3, we compare the computational cost of our method with previous global retrieval methods and retrieval+reranking methods. Apparently, the inference latency of extraction and retrieval is negligible as compared to the reranking latency of previous works. Our method is significantly faster than all the other reranking-based methods because our reranking process is only one network forward pass on GPU and can be easily accelerated by parallel computing, *e.g.* reranking on 4 GPUs takes only 47.17 ms per query (20.2× speedup over previous reranking-based methods [26,53]). However, it is not straightforward to accelerate RANSAC-based geometric verification with multi-processing. Previous works [26,53] generally follow the standard implementation of OpenCV [8]. In addition, the local feature dimension of our method is the smallest among all the reranking-based methods, specifically, we only save 500 important local features with very small dimensions $(128 + 3)$ for each image. In total, it only needs 4.79 GB of memory for the MSLS validation set which is much smaller (189× reduction) than 908.3 GB of Patch-NetVLAD-p.

	R@1	R@5	R@10
No Reranking	79.3	90.8	92.6
RANSAC [19]	84.9	93.0	94.5
RRT [48]	81.2	91.9	93.1
CVNet [32]	73.4	86.8	91.4
Ours	**89.7**	**95.0**	**96.2**

Table 4. Comparison with different reranking methods (top-100 candidates reranked) using our backbone on MSLS Val.

	R@1	R@5	R@10
No Reranking	79.3	90.8	92.6
Reranking	86.6	94.1	95.0
Reranking + Mining	88.4	93.4	94.9
Reranking + Mining + Finetune	**89.7**	**95.0**	**96.2**
End-to-end Training	87.3	93.5	95.4

Table 5. Ablation study on training strategy. We freeze the global retrieval module by default when training the reranking module.

	Architecture	R@1	R@5	R@10
Ours w/o Reranking	ViT-Small	79.3	90.8	92.6
	ResNet50 + GeM	79.6	90.9	92.6
	ViT-Base	84.9	92.7	94.5
Ours w/ RANSAC	ViT-Small	84.9	93.0	94.5
	ResNet50 + GeM	84.3	91.4	93.0
	ViT-Base	87.0	93.0	94.6
Ours	ViT-Small	89.7	95.0	96.2
	ResNet50 + GeM	88.4	93.6	95.3
	ViT-Base	**90.0**	**95.1**	**96.9**

Table 6. Ablation study on MSLS Val with different backbone architectures, including ResNet50+GeM [27,43], ViT Small [17], ViT Base [17]. Transformer tokens perform on par with CNN local features in terms of reranking/local matching.

	R@1	R@5	R@10
No Reranking	79.3	90.8	92.6
Reranking	**86.6**	**94.1**	**95.0**
Reranking Module			
Remove Positional Embedding	86.4	93.2	94.7
Remove Transformer Block-1	85.5	93.8	95.0
Remove Transformer Block-2	81.9	92.8	94.6
Replace Top-5 w/ Top-1	86.6	93.8	94.7
Reranking Input			
Remove Attention Selection	83.6	92.8	94.1
Remove xy Coordinates Value	85.0	93.5	94.7
Remove Attention Value	86.1	93.5	94.3
Remove Correlation Value	31.8	59.2	72.8

Table 7. Ablation study on model components. We freeze the global retrieval module so that the global and local features are the same for different ablations. All ablations are trained with partial negative mining [6]. The components are defined in Sec. 3.

Even for datasets with over 1M reference images (*e.g.* R-SF dataset), the memory footprint is only 244.01 GB which can fit in a typical server or high-end desktop. It can be further reduced to 122 GB by using float16 instead of float32, with which we observe no performance drop on R-SF. However, other reranking-based methods typically require much larger ($> 4\times$) memory, which cannot scale to large-scale real-world scenarios with 1M reference images. Given the unified design, outstanding performance, strong efficiency, and scalability of our method, it could serve as a well-balanced solution for real-world large-scale applications.

4.4. Ablation Study

Reranking Methods. In Table 4, we adopt the same global and local features from our default backbone and compare different reranking methods. "No Reranking" denotes our model with only global retrieval. "RANSAC" [19] follows the pipeline of state-of-the-art methods [26, 53] using $1.5\times$ of the patch size. "RRT" [48] is not proposed for VPR, nevertheless, we train it on the top of our backbone for comparison. "CVNet" [31] does not provide the training code, we thus adopt their pre-trained model trained on Google Landmark [56]. Remarkably, our reranking module significantly outperforms RANSAC [19] based geometric verification using much less ($< 22\%$) inference time (Table 3). It also outperforms two recent state-of-the-art reranking methods on landmark retrieval, indicating the superiority of our reranking module.

Training Strategies. In Table 5, we compare our method based on different training strategies. "No Reranking" uses only global retrieval, and "Reranking" denotes training reranking module with partial negative mining on a fixed global retrieval module. "Reranking+Mining" means training the reranking module with our global negative mining by randomly selecting one negative sample from the 100 global hardest samples. "Reranking+Mining+Finetune" further finetunes retrieval and reranking module together. "End-to-end Training" directly trains global retrieval and reranking module from the beginning in an end-to-end manner with partial negative mining. All the strategies with reranking outperform previous state-of-the-art methods on R@5, and training the global retrieval and reranking module separately with a carefully designed strategy achieves the best performance. The results indicate that global hard negative mining is helpful for training the reranking module. End-to-end training is also a good trade-off between simplicity and performance.

Different Backbones. In Table 6, we combine our reranking module with different backbone retrieval modules, including ViT-Small (default), ResNet50+GeM [27, 43] and ViT-Base. We follow [6] to add L2 normalization before the pooling layer but do not freeze or remove any convolutional layer. The attention map of ResNet50+GeM is generated following CAM [61]. The ResNet50+GeM model achieves competitive results for both global retrieval and reranking. It also significantly outperforms previous

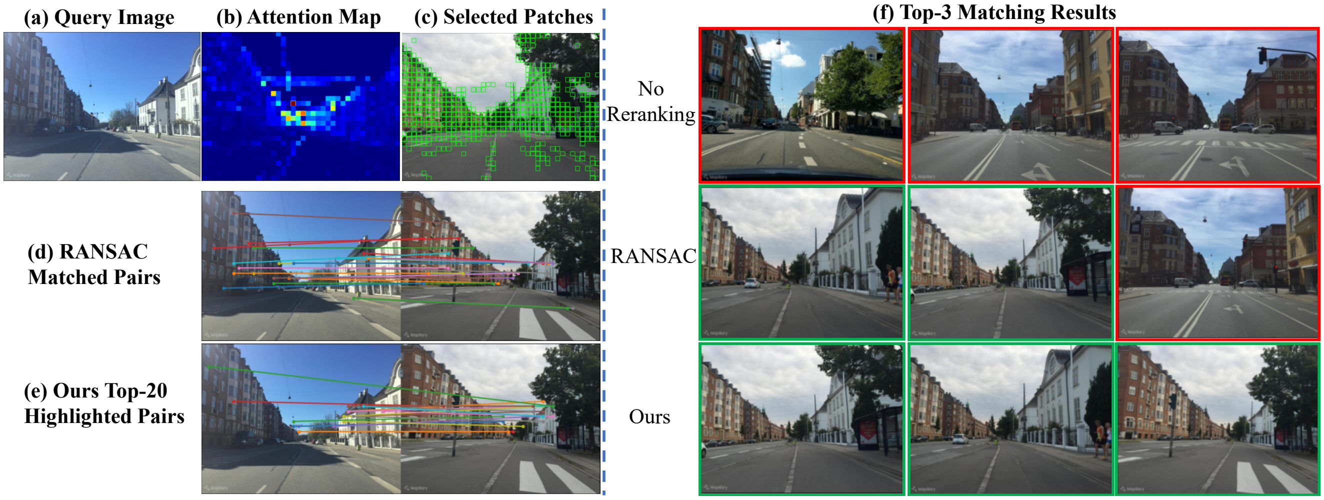


Figure 4. Visualization of attention map and matched pairs. Green and red boxes in (f) indicate correct and wrong predictions.

works [26, 53], indicating that our reranking module is a generic component for both CNN and transformer backbone architectures. With similar global retrieval performance and the same reranking method (RANSAC or Ours), the ViT-Small model achieves slightly better performance than ResNet50+GeM, which means the transformer token is a good alternative for local matching as compared with CNN local feature. Furthermore, we adopt a larger backbone ViT-Base, which achieves significant performance improvement on global retrieval and RANSAC-based reranking over ViT-Small. However, the overall performance of our ViT-Base model has very small performance improvement (mainly on R@10), which indicates a possible bottleneck of the proposed framework.

Components Ablation. In Table 7, we use the "Reranking" configuration in Table 5 as a baseline and ablate different components to check the importance of each component of our method. For simplicity, we freeze the global retrieval module and all the models are trained with partial negative mining. We have the following observations. (1) The most important components are the Transformer Block-2 and Correlation Value, and removing them causes a significant performance drop. This indicates that the feature correlation contributes most to the final decision. (2) When we replace the attention-based feature selection with random selection, the model becomes unstable and the performance drops moderately. (3) Removing other components, *i.e.* Transformer Block-1, xy Coordinates Value, Attention Value (defined in Sec. 3.2), causes a slight performance drop, indicating the effectiveness of these blocks for performance purposes. (4) We observe negligible performance drop when removing positional embedding or replacing top-5 pairs selection with top-1, which means the Sinusoidal positional embedding does not help much and the top-1 pair contains most of the information that is needed for reranking.

4.5. Visualization

In Fig. 4, we show a detailed case where the global retrieval fails on the top-3 results. Fig. 4 (b) and (c) show the attention map and the selected patches, which cover most of the informative regions for reranking. Both RANSAC and our method work well on reranking the top-100 candidates and our method finds more correct reference images in the top-3 results (Fig. 4 (f)). Although our reranking module is not as explainable as RANSAC, the transformer still focuses on specific patch pairs and we plot the top-20 pairs with the highest attention in the reranking module. As shown in Fig. 4 (d) and (e), our reranking also focuses on meaningful pairs and most of them are correct matches. As compared with RANSAC, our matched pairs may not have tight geometric consistency, but they may provide different information for the reranking as adjusted according to the task-relevant training data.

5. Conclusion

We propose a unified retrieval and reranking framework for place recognition with only transformers. We for the first time show that vision transformer tokens are comparable to or even better than CNN local feature in terms of reranking. Our reranking module is generic and can be adopted on other CNN or transformer backbones. Remarkably, it significantly outperforms previous methods on major datasets with much less inference time and memory consumption. Our method can also provide possible matched local pairs like RANSAC and could be improved with geometric modeling in the future. We discuss the limitations and societal impact in the supplementary material.

References

[1] Amar Ali-bey, Brahim Chaib-draa, and Philippe Giguère. Mixvpr: Feature mixing for visual place recognition. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2998–3007, 2023. 6

[2] Adrien Angeli, David Filliat, Stéphane Doncieux, and Jean-Arcady Meyer. Fast and incremental method for loop-closure detection using bags of visual words. *IEEE transactions on robotics*, 24(5):1027–1037, 2008. 2

[3] 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. 1, 2, 3, 4, 5, 6

[4] 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. 2

[5] Gabriele Berton, Carlo Masone, Valerio Paolicelli, and Barbara Caputo. Viewpoint invariant dense matching for visual geolocalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12169–12178, 2021. 2

[6] Gabriele Berton, Riccardo Mereu, Gabriele Trivigno, Carlo Masone, Gabriela Csurka, Torsten Sattler, and Barbara Caputo. Deep visual geo-localization benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5396–5407, 2022. 1, 2, 3, 4, 5, 6, 7

[7] Gabriele Moreno Berton, Valerio Paolicelli, Carlo Masone, and Barbara Caputo. Adaptive-attentive geolocalization from few queries: A hybrid approach. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2918–2927, 2021. 2

[8] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000. 6

[9] Guillaume Bresson, Zayed Alsayed, Li Yu, and Sébastien Glaser. Simultaneous localization and mapping: A survey of current trends in autonomous driving. *IEEE Transactions on Intelligent Vehicles*, 2(3):194–220, 2017. 1

[10] 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. 2

[11] David M Chen, Georges Baatz, Kevin Köser, Sam S Tsai, Ramakrishna Vedantham, Timo Pylvänäinen, Kimmo Roimela, Xin Chen, Jeff Bach, Marc Pollefeys, et al. City-scale landmark identification on mobile devices. In *CVPR 2011*, pages 737–744. IEEE, 2011. 5

[12] Gabriella Csurka, Christopher Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV*, volume 1, pages 1–2. Prague, 2004. 2

[13] Mark Joseph Cummins and Paul Newman. Highly scalable appearance-only slam - fab-map 2.0. In *Robotics: Science and Systems*, 2009. 6

[14] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 3, 5

[15] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 224–236, 2018. 2, 4, 5, 6

[16] Carl Doersch, Ankush Gupta, and Andrew Zisserman. Crosstransformers: spatially-aware few-shot transfer. *Advances in Neural Information Processing Systems*, 33:21981–21993, 2020. 6

[17] 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. 1, 2, 3, 4, 5, 7

[18] Zhaoxin Fan, Zhenbo Song, Hongyan Liu, Zhiwu Lu, Jun He, and Xiaoyong Du. Svt-net: Super light-weight sparse voxel transformer for large scale place recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 551–560, 2022. 6

[19] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 1, 2, 5, 7

[20] Matthew Gadd, Daniele De Martini, and Paul Newman. Look around you: Sequence-based radar place recognition with learned rotational invariance. In *2020 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, pages 270–276. IEEE, 2020. 2

[21] Sourav Garg, Tobias Fischer, and Michael Milford. Where is your place, visual place recognition? In Zhi-Hua Zhou, editor, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 4416–4425. International Joint Conferences on Artificial Intelligence Organization, 8 2021. Survey Track. 2

[22] Sourav Garg and Michael Milford. Seqnet: Learning descriptors for sequence-based hierarchical place recognition. *IEEE Robotics and Automation Letters*, 6(3):4305–4312, 2021. 2

[23] 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*, pages 369–386. Springer, 2020. 2, 5, 6

[24] Albert Gordo, Jon Almazan, Jerome Revaud, and Diane Larlus. End-to-end learning of deep visual representations for image retrieval. *International Journal of Computer Vision*, 124(2):237–254, 2017. 2

[25] Ali Hassani, Steven Walton, Nikhil Shah, Abulikemu Abuduweili, Jiachen Li, and Humphrey Shi. Escaping the big data paradigm with compact transformers. *arXiv preprint arXiv:2104.05704*, 2021. 6

[26] 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. 1, 2, 4, 5, 6, 7, 8

[27] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2, 3, 7

[28] Hervé Jégou, Florent Perronnin, Matthijs Douze, Jorge Sánchez, Patrick Pérez, and Cordelia Schmid. Aggregating local image descriptors into compact codes. *IEEE transactions on pattern analysis and machine intelligence*, 34(9):1704–1716, 2011. 2

[29] Wei Jiang, Eduard Trulls, Jan Hosang, Andrea Tagliasacchi, and Kwang Moo Yi. Cotr: Correspondence transformer for matching across images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6207–6217, 2021. 2

[30] Hyo Jin Kim, Enrique Dunn, and Jan-Michael Frahm. Learned contextual feature reweighting for image geo-localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2136–2145, 2017. 2

[31] Seongwon Lee, Hongje Seong, Suhyeon Lee, and Euntai Kim. Correlation verification for image retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5374–5384, June 2022. 2, 7

[32] Seongwon Lee, Hongje Seong, Suhyeon Lee, and Euntai Kim. Correlation verification for image retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5374–5384, 2022. 7

[33] Yunpeng Li, Noah Snavely, Dan Huttenlocher, and Pascal Fua. Worldwide pose estimation using 3d point clouds. In *European conference on computer vision*, pages 15–29. Springer, 2012. 5

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

[35] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 5

[36] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 5

[37] David G Lowe. Object recognition from local scale-invariant features. In *Proceedings of the seventh IEEE international conference on computer vision*, volume 2, pages 1150–1157. Ieee, 1999. 2

[38] 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, 2016. 2

[39] Carlo Masone and Barbara Caputo. A survey on deep visual place recognition. *IEEE Access*, 9:19516–19547, 2021. 2

[40] Michael J Milford and Gordon F Wyeth. Mapping a suburb with a single camera using a biologically inspired slam system. *IEEE Transactions on Robotics*, 24(5):1038–1053, 2008. 5

[41] Piotr Mirowski, Matt Grimes, Mateusz Malinowski, Karl Moritz Hermann, Keith Anderson, Denis Teplyashin, Karen Simonyan, Andrew Zisserman, Raia Hadsell, et al. Learning to navigate in cities without a map. In *Advances in Neural Information Processing Systems*, pages 2419–2430, 2018. 1

[42] 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*, 32:8026–8037, 2019. 5

[43] 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. 1, 2, 4, 6, 7

[44] 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. 2, 5, 6

[45] Torsten Sattler, Akihiko Torii, Josef Sivic, Marc Pollefeys, Hajime Taira, Masatoshi Okutomi, and Tomas Pajdla. Are large-scale 3d models really necessary for accurate visual localization? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1637–1646, 2017. 5

[46] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 2

[47] Elena Stumm, Christopher Mei, and Simon Lacroix. Probabilistic place recognition with covisibility maps. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4158–4163. IEEE, 2013. 1

[48] Fuwen Tan, Jiangbo Yuan, and Vicente Ordonez. Instance-level image retrieval using reranking transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12105–12115, 2021. 2, 7

[49] Akihiko Torii, Relja Arandjelovic, Josef Sivic, Masatoshi Okutomi, and Tomas Pajdla. 24/7 place recognition by view synthesis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1808–1817, 2015. 5, 6

[50] 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. 5, 6

[51] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021. 3, 5

[52] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 2, 4, 5

[53] 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. 1, 2, 3, 4, 5, 6, 7, 8

[54] Ziqi Wang, Jiahui Li, Seyran Khademi, and Jan van Gemert. Attention-aware age-agnostic visual place recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019. 2

[55] 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. 2, 5, 6

[56] Tobias Weyand, Andre Araujo, Bingyi Cao, and Jack Sim. Google landmarks dataset v2-a large-scale benchmark for instance-level recognition and retrieval. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2575–2584, 2020. 7

[57] Yifan Xu, Pourya Shamsolmoali, Eric Granger, Claire Nicodeme, Laurent Gardes, and Jie Yang. Transvlad: Multi-scale attention-based global descriptors for visual geo-localization. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2840–2849, 2023. 6

[58] Amir Roshan Zamir and Mubarak Shah. Image geo-localization based on multiplenearest neighbor feature matching usinggeneralized graphs. *IEEE transactions on pattern analysis and machine intelligence*, 36(8):1546–1558, 2014. 2

[59] Hao Zhang, Xin Chen, Heming Jing, Yingbin Zheng, Yuan Wu, and Cheng Jin. Etr: An efficient transformer for re-ranking in visual place recognition. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5665–5674, 2023. 6

[60] Xiwu Zhang, Lei Wang, and Yan Su. Visual place recognition: A survey from deep learning perspective. *Pattern Recognition*, 113:107760, 2021. 2

[61] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Computer Vision and Pattern Recognition*, 2016. 7

[62] Sijie Zhu, Mubarak Shah, and Chen Chen. Transgeo: Transformer is all you need for cross-view image geo-localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1162–1171, 2022. 6