

Instance-level Image Retrieval using Reranking Transformers

Fuwen Tan
University of Virginia
fuwen.tan@virginia.edu

Jiangbo Yuan
eBay Computer Vision
jiayuan@ebay.com

Vicente Ordonez
Rice University
vicenteor@rice.edu

Abstract

Instance-level image retrieval is the task of searching in a large database for images that match an object in a query image. To address this task, systems usually rely on a retrieval step that uses global image descriptors, and a subsequent step that performs domain-specific refinements or reranking by leveraging operations such as geometric verification based on local features. In this work, we propose Reranking Transformers (RRTs) as a general model to incorporate both local and global features to rerank the matching images in a supervised fashion and thus replace the relatively expensive process of geometric verification. RRTs are lightweight and can be easily parallelized so that reranking a set of top matching results can be performed in a single forward-pass. We perform extensive experiments on the Revisited Oxford and Paris datasets, and the Google Landmarks v2 dataset, showing that RRTs outperform previous reranking approaches while using much fewer local descriptors. Moreover, we demonstrate that, unlike existing approaches, RRTs can be optimized jointly with the feature extractor, which can lead to feature representations tailored to downstream tasks and further accuracy improvements. The code and trained models are publicly available at <https://github.com/uwavision/RerankingTransformer>.

1. Introduction

Instance recognition is a challenging task that aims to visually recognize an object instance. This is distinct from category-level recognition that identifies only the object class. Instance recognition is important in e-commerce where it is often desired to find a specific product in a large image collection, or in place identification where the objective is to infer the identity of a public landmark. As the number of instances is much larger than the number of object categories, instance recognition is typically cast as image retrieval instead of classification, and usually involves both metric learning and local feature based reranking.

Over the last decade, instance recognition continues to be a major focus of research. Pioneering systems lever-

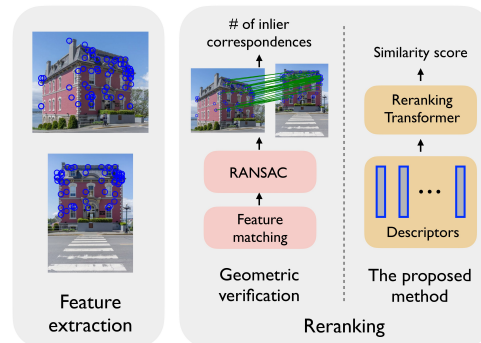


Figure 1. Top performing instance recognition methods often rely on reranking the top results using a score such as the number of inlier correspondences from geometric verification. We propose to replace this step with a Reranking Transformer (RRT) that can be learned with the underlying representations of the images.

aged hand-crafted local descriptors and matching algorithms [52, 42]. More recent approaches incorporate both global and local descriptors extracted from deep learning models [3, 37]. Global descriptors summarize an image into a single vector, leading to a compact representation for large-scale search. Local descriptors encode detailed spatial features for patch-level matching, and are shown to be important for high retrieval precision [56, 9]. The best methods typically use a global descriptor to reduce the solution space to a set of candidate matching images, and local descriptors to *re-rank* the nearest images [51, 9]. While extensive progress has been made to improve image retrieval using global features, fewer efforts have been made to develop similarity metrics based on local features. State-of-the-art approaches still rely on classic matching techniques, such as geometric verification [42] and aggregated selective match kernels (ASMK) [55]. Geometric verification assumes object instances are rigid and local matches between images can be estimated as a homography using RANSAC [20]. It is also an expensive process that requires iterative optimization on a large set of local descriptors. ASMK aggregates the similarities of features without modeling the geometric alignment, but requires offline clustering and encoding procedures. It was mainly used as a global retrieval technique in previous literature. Both geometric verification and

ASMK require large amounts of local descriptors to ensure retrieval performance.

In this work, we propose *Reranking Transformers* (RRTs), which learn to predict the similarity of an image pair directly. Our method is general and can be used as a drop-in replacement for other reranking approaches such as geometric verification. We conduct detailed experiments showing that as either a drop-in replacement or trained together with a global retrieval approach, the proposed method is the top-performing across the standard benchmarks for instance recognition. RRTs leverage the transformer architecture [59] which has led to significant improvements in natural language processing [16, 30] and vision-and-language tasks [28, 12, 33]. Most recently, it has also been used for purely vision tasks, notably for image recognition [18] and object detection [10]. To the best of our knowledge, our work is the first to adapt transformers for a visual task involving the analysis of image pairs in the context of reranking image search results.

Reranking Transformers are lightweight. Compared with typical feature extractors which have over 20 million parameters (e.g. 25 million in ResNet50 [24]), the proposed model only has 2.2 million parameters. It can also be easily parallelized such that re-ranking the top 100 neighbors requires a single pass. As shown in Fig. 1, our method directly predicts a similarity score for the matching images, instead of estimating a homography, which may be challenging under large viewpoint changes or infeasible for deformable objects. Our method requires much fewer descriptors but achieves superior performance, especially for challenging cases. In current state-of-the-art models, the feature extraction and matching are optimized separately, which may lead to suboptimal feature representations. In this work, we first perform experiments using pretrained feature extractors, then demonstrate the benefit of jointly optimizing the feature extractor and our model in a unified framework.

Contributions. (1) We propose *Reranking Transformers* (RRTs), a small and effective model which learns to predict the similarity of an image pair based on global and local descriptors; (2) Compared with existing methods, RRTs require fewer local descriptors and can be parallelized so that reranking the top neighbors only requires a single forward-pass; (3) We perform extensive experiments on Revisited Oxford/Paris [44], Google Landmarks v2 [61], and Stanford Online Products, and show that RRTs outperform prior reranking methods across a variety of settings.

2. Related Work

Feature learning for instance recognition. Hand-crafted local descriptors [34, 32] were widely used in earlier instance retrieval work [52, 36]. Recently, local features extracted from convolution neural networks (CNN) are shown to be more effective [37, 54, 51, 19, 38, 39]. Some

of these works learn feature detection and representation jointly by non-local maximum suppression [19, 56], or attention [37, 54, 9]. The detected local descriptors are usually used for geometric verification [42] or ASMK [55]. Compared to local features, global descriptors provide a compact representation of an image for large-scale search. Current global descriptors are typically extracted from CNN models [3, 58, 43, 21] by spatial pooling [2, 26, 58, 43, 35], which may not be ideal for modeling region-wise relations across images. Recent systems either use global descriptors to reduce the solution space and then local descriptors to *re-rank* the nearest neighbors, or encode local descriptors using a large visual codebook, followed by image matching with an aggregated selective match kernel [55, 54, 56]. This work mainly follows the retrieve-and-rerank paradigm.

Reranking for instance recognition/retrieval. Geometric verification is the dominant image reranking approach and widely used in both traditional [42] and more recent works [51, 37, 9]. Inspired by text retrieval, query expansion techniques have also been introduced for image retrieval [14, 13, 57, 22]. These methods differ from geometric verification and our work as they rely on analysing the local nearest neighbor graph for each query during testing. Diffusion based approaches [17, 63, 25, 5, 4] aim to learn the structure of the data manifold by similarity propagation over the global affinity graph built on a query and all the gallery images, which is nontrivial to scale. Overall, the motivation of image reranking is to make better use of test-time knowledge to refine retrieval results. Our work shares the same vision with this line of research but focuses more on learning the similarity of an image pair directly.

Transformers for visual tasks. Transformers have become the dominant architecture for representing text [16, 30]. Recently, it has also been introduced to vision-and-language [28, 33] and pure vision tasks [40, 10, 27]. As the key ingredient of the transformer architecture, the self-attention mechanism has also been studied for visual recognition [6, 45, 64]. These works apply transformers for single image predictions while we leverage transformers to learn the visual relation of an image pair. Our work is also closely related to SuperGlue [50], which while not a Transformer, also relies on self-attention. SuperGlue aims to learn local correspondences between images with pixel-level supervision. Our work differs from SuperGlue in that it learns the similarity of an image pair with image-level supervision. We provide a best-effort comparison with this approach.

3. Methodology

3.1. Attention Modules in Transformers

First, we briefly review the key ingredients in the Transformer architecture: Single-Head Attention (SHA) and Multi-Head Attention (MHA).

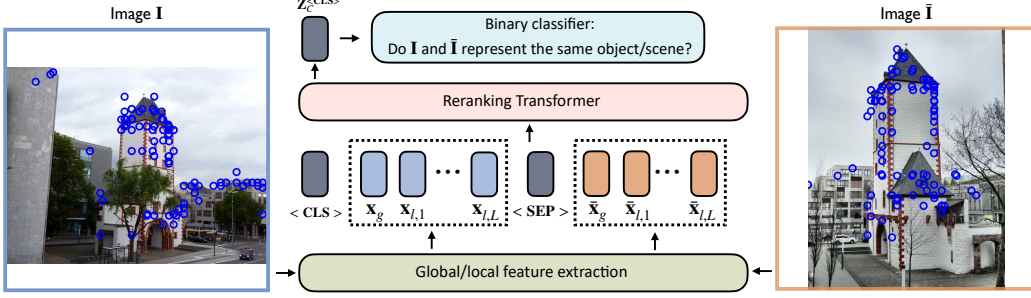


Figure 2. Illustration of the proposed *Reranking Transformer* (RRT) model. The input of RRT is a sequence of global and local descriptors (circled in blue) extracted from an image pair $(\mathbf{I}, \bar{\mathbf{I}})$. This sequence, together with two special tokens, are fed into a multi-layer transformer model which produces a similarity score of $(\mathbf{I}, \bar{\mathbf{I}})$. The model is trained to optimize a binary cross entropy loss.

Single-Head Attention (SHA): The input of a SHA layer comprises three sets of variables: the queries $\mathbf{Q} := \{\mathbf{q}_i \in \mathbb{R}^{d_q}\}_{i=1}^N$, the keys $\mathbf{K} := \{\mathbf{k}_j \in \mathbb{R}^{d_k}\}_{j=1}^M$, and the values $\mathbf{V} := \{\mathbf{v}_j \in \mathbb{R}^{d_v}\}_{j=1}^M$. Here, d_q, d_k, d_v are the dimensions of the corresponding feature vectors, N and M are the sequence lengths. SHA produces a new feature sequence where each vector is a linear combination of $\{\mathbf{v}_j\}$. In doing this, $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ are first linearly projected as $\bar{\mathbf{Q}} = \mathbf{Q}\mathbf{W}^Q$, $\bar{\mathbf{K}} = \mathbf{K}\mathbf{W}^K$, $\bar{\mathbf{V}} = \mathbf{V}\mathbf{W}^V$, using parameter tensors: $\mathbf{W}^Q \in \mathbb{R}^{d_q \times d_h}$, $\mathbf{W}^K \in \mathbb{R}^{d_k \times d_h}$, $\mathbf{W}^V \in \mathbb{R}^{d_v \times d_h}$, where d_h is the new feature dimension. The output of a SHA layer is computed as: $\text{SHA}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) := \text{SOFTMAX}\left(\frac{\bar{\mathbf{Q}}\bar{\mathbf{K}}^T}{\sqrt{d_h}}\right)\bar{\mathbf{V}}$.

Multi-Head Attention (MHA): Like SHA, MHA takes $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ as input and comprises multiple SHA modules: $\text{MHA}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) := [\text{HEAD}_1; \dots; \text{HEAD}_h]\mathbf{W}^O$, $\text{HEAD}_i := \text{SHA}_i(\mathbf{Q}, \mathbf{K}, \mathbf{V})$. Here $[\cdot]$ denotes the concatenation operator, h is the number of the SHA heads. $\mathbf{W}^O \in \mathbb{R}^{(hd_h) \times d_e}$ is a linear projection with an output dimension of d_e .

3.2. Model

With the fundamental building blocks defined above, we introduce the detailed formulation of our model:

Image representations: An image \mathbf{I} is represented by a global descriptor of a dimension d_g : $\mathbf{x}_g \in \mathbb{R}^{d_g}$ and a set of L local descriptors: $\mathbf{x}_l = \{\mathbf{x}_{l,i} \in \mathbb{R}^{d_l}\}_{i=1}^L$, each of a dimension d_l . Both \mathbf{x}_g and \mathbf{x}_l are extracted from a CNN backbone (to be discussed in Sec. 4.2). Optionally, each $\mathbf{x}_{l,i}$ is associated with a coordinate tuple $\mathbf{p}_{l,i} = (u, v) \in \mathbb{R}^2$ and a scale factor $s_{l,i} \in \mathbb{R}$, indicating the pixel location and image scale where $\mathbf{x}_{l,i}$ is extracted from. In this work, $s_{l,i}$ is an integer, indexing a set of pre-defined image scales.

Input: As a sequence transduction model [16, 30], Transformers take as input a list of “tokens” (e.g. $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ in Sec. 3.1). In image retrieval, these “tokens” can be derived from the features of an image pair $(\mathbf{I}, \bar{\mathbf{I}})$. Following the BERT transformer encoder [16], we define the input as:

$$\mathbf{X}(\mathbf{I}, \bar{\mathbf{I}}) := [\langle \text{CLS} \rangle; f_g(\mathbf{x}_g); f_l(\mathbf{x}_{l,1}); \dots; f_l(\mathbf{x}_{l,L}); \langle \text{SEP} \rangle; \bar{f}_g(\bar{\mathbf{x}}_g); \bar{f}_l(\bar{\mathbf{x}}_{l,1}); \dots; \bar{f}_l(\bar{\mathbf{x}}_{l,L})], \quad (1)$$

where:

$$\begin{aligned} f_g(\mathbf{x}_g) &:= \mathbf{x}_g + \alpha; \\ f_l(\mathbf{x}_{l,i}) &:= \mathbf{x}_{l,i} + \varphi(\mathbf{p}_{l,i}) + \psi(s_{l,i}) + \beta \\ \bar{f}_g(\bar{\mathbf{x}}_g) &:= \bar{\mathbf{x}}_g + \bar{\alpha}; \\ \bar{f}_l(\bar{\mathbf{x}}_{l,i}) &:= \bar{\mathbf{x}}_{l,i} + \varphi(\bar{\mathbf{p}}_{l,i}) + \psi(\bar{s}_{l,i}) + \bar{\beta}. \end{aligned} \quad (2)$$

Here, $\langle \text{CLS} \rangle$ is a special token used for summarizing the signals from both images. $\langle \text{SEP} \rangle$ is an extra separator token. $\alpha, \bar{\alpha}, \beta, \bar{\beta}$ are one dimensional segment embeddings, being used to distinguish the global and local descriptors of \mathbf{I} and $\bar{\mathbf{I}}$. φ is a position embedding function, as used in [10]. ψ is a linear embedding taking the scale index $s_{l,i}$ as input.

Model architecture: With input $\mathbf{X}(\mathbf{I}, \bar{\mathbf{I}})$, we define a multi-layer transformer where each layer is formulated as:

$$\begin{aligned} \bar{\mathbf{Z}}_{i+1} &= \text{LAYERNORM}(\mathbf{Z}_i + \text{MHA}(\mathbf{Z}_i)), \\ \mathbf{Z}_{i+1} &= \text{LAYERNORM}(\text{MLP}(\bar{\mathbf{Z}}_{i+1})), \\ \text{MLP}(\bar{\mathbf{Z}}_{i+1}) &= \text{RELU}(\bar{\mathbf{Z}}_{i+1}\mathbf{W}_1^T)\mathbf{W}_2^T, \\ i &= 0, \dots, C-1. \end{aligned} \quad (3)$$

In this setting, the $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ features for MHA are the same set of vectors \mathbf{Z}_i , with $\mathbf{Z}_0 = \mathbf{X}(\mathbf{I}, \bar{\mathbf{I}})$. MLP is a two-layer perceptron with parameter matrices $\mathbf{W}_1 \in \mathbb{R}^{d_e \times d_c}$ and $\mathbf{W}_2 \in \mathbb{R}^{d_c \times d_e}$, and an intermediate dimension d_c . LAYERNORM is a layer normalization function proposed in [1]. The model includes C transformer layers in total.

Training objective: Our model is trained to optimize a binary cross entropy loss:

$$\mathcal{E}(\mathbf{I}, \bar{\mathbf{I}}) = \text{BCE}(\text{SIGMOID}(\mathbf{Z}_C^{(\text{CLS})}\mathbf{W}_z^T), \mathbb{1}(\mathbf{I}, \bar{\mathbf{I}})), \quad (4)$$

$\mathbf{Z}_C^{(\text{CLS})} \in \mathbb{R}^{d_e}$ is a feature vector, corresponding to the $\langle \text{CLS} \rangle$ token. It is extracted from the last transformer layer. $\mathbf{W}_z^T \in \mathbb{R}^{d_e \times 1}$ is a linear function mapping $\mathbf{Z}_C^{(\text{CLS})}$ to a logit scalar. $\mathbb{1}(\mathbf{I}, \bar{\mathbf{I}})$ is an indicator function which equals to one when \mathbf{I} and $\bar{\mathbf{I}}$ represent the same object, or zero otherwise. Fig. 2 provides an illustration of the proposed model.

4. Experiments

Next, we describe the datasets we use to evaluate our approach, and details about our implementation.

4.1. Datasets

We perform experiments on three datasets, Google Landmarks v2 (GLDv2) [61], Revisited Oxford/Paris [44], and Stanford Online Products (SOP) [53].

GLDv2: Google Landmarks v2 (GLDv2) [61] is a new benchmark for instance recognition that includes over five million images from 200k natural landmarks. As the Reranking Transformer has limited parameters (e.g. 2.2 million), we sample a small subset of the images from the “v2-clean” split of GLDv2 for training. We *randomly* sample 12,000 landmarks where each landmark has *at least* 10 images. For each landmark, we *randomly* sample *at most* 500 images. This results in 322,008 images, which is 20% of the “v2-clean” split and 8% of the original training set. The names of the sampled images are included in the supplementary material. For testing, we evaluate on the standard test set for the retrieval task, which contains 1,129 query images and 761,757 gallery images.

ROxf and **RPar:** Revisited Oxford (ROxf) and Paris (RPar) [44] are standard benchmarks for instance recognition, which have 4,993 and 6,322 gallery images respectively. They both have 70 query images, each with a bounding box depicting the location and span of the prominent landmark. An extra distractor set (R1M) with 1,001,001 images is included for large-scale experiments. We follow the standard evaluation protocol [44, 9] and crop the query image using the provided bounding box. We report mean Average Precision (mAP) on the Medium and Hard setups.

SOP: To investigate the benefit of jointly optimizing the feature representation and our Reranking Transformer, we perform experiments on a dataset of product images: Stanford Online Products (SOP) [53]. SOP is a commonly used benchmark for metric learning [62, 49, 47, 8, 48, 60, 7], which includes 120,053 images, 59,551 for training, 60,502 for testing. We follow the evaluation protocol for metric learning and report the R@K scores.

4.2. Implementation

Experiments on the pretrained descriptors: We first perform experiments with descriptors obtained from a pretrained feature extractor, DELG [9]. Our main experiments leverage ResNet50 [24] as the CNN backbone, but we also include experiments with ResNet101 in the supplementary material. DELG provides a unified framework for global/local feature extraction. The local descriptors, each with a dimension of 128, are extracted at 7 image scales ranging from 0.25 to 2.0. The global descriptor with a dimension of 2048 is extracted at 3 scales: $\{\frac{1}{\sqrt{2}}, 1, \sqrt{2}\}$. We use an extra linear projection to reduce the global descriptor

to a dimension of 128. In the original DELG model, the top 1000 local descriptors with the highest attention scores are selected for image reranking. We observe that RRT does not require this amount of descriptors, and the retrieval performance saturates at 500 local descriptors. Thus, in our experiments we choose the top 500 local descriptors and set $L = 500$, $d_g = d_l = 128$. For images with fewer descriptors, we pad the feature sequence with empty vectors and use a binary attention mask, as in BERT [16], to indicate the padding locations. Both the global and local features are L2 normalized to unit norm. During training, the positive image is *randomly* sampled from the images sharing the same label as the query. The negative image is *randomly* sampled from the top 100 neighbors returned by the global retrieval, which have a different label from the query. DELG is pretrained on both Google Landmarks (GLD) v1 [37] and v2-clean [61]. Thus, we perform experiments on two sets of descriptors extracted from these two models. For the architecture, we use 4 SHA heads ($h = 4$) and 6 transformer layers ($C = 6$). d_q, d_k, d_v and d_e in SHA are set to 128, d_h is set to 32, d_c in MLP (Eq. 3) is set to 1024. The number of learnable parameters is 2,243,201, which is 9% of the amount in ResNet50. The model is trained with AdamW [31] for 15 epochs, using a learning rate of 0.0001 and a weight decay of 0.0004.

Experiments on SOP: We perform experiments on SOP [53] using a single image scale, following the protocol for metric learning [60]. During training, each image is randomly cropped to 224×224 , followed by a random flip. During testing, each image is first resized to of 256×256 then cropped at the center to 224×224 . We use ResNet50 and extract features from the last convolutional layer, which leads to 49 (7×7) local descriptors for each image. The global descriptor is obtained by spatially averaging the local responses. Both the global and local descriptors are linearly projected a dimension of 128. The RRT architecture and most of the training details remain the same as in the DELG experiments. Here we only describe the main differences. The global model is trained with a contrastive loss, as in [60]. Different from [60], we do not rely on a cross batch memory but simply use a batch size of 800. As all the local features are used, we do not incorporate the global descriptor term ($f_g(\mathbf{x}_g), \bar{f}_g(\bar{\mathbf{x}}_g)$) in Eq. 1. We also drop the scale embedding (ψ) as only one image scale is used. The global model is trained using SGD with Nesterov momentum for 100 epochs, using a learning rate of 0.001, a weight decay of 0.0004 and a momentum of 0.9. The learning rate drops by a factor of 10 after 60 and 80 epochs. We train an RRT model on top of the pretrained global model, either freezing or finetuning the CNN backbone. Both models are trained with AdamW [31] for 100 epochs, using a learning rate of 0.0001. The learning rate drops by a factor of 10 after 60 and 80 epochs. We implement RRTs in PyTorch [41].

Method	# local desc.	# Reranked images	Desc. version	Medium				Hard			
				\mathcal{ROxf}	$+\mathcal{R1M}$	\mathcal{RPar}	$+\mathcal{R1M}$	\mathcal{ROxf}	$+\mathcal{R1M}$	\mathcal{RPar}	$+\mathcal{R1M}$
DELG global	0	0	R50-v1	69.7	55.0	81.6	59.7	45.1	27.8	63.4	34.1
GV	1000	100	R50-v1	75.4	61.1	82.3	60.5	54.2	36.8	64.9	34.8
RRT (ours)	500	100	R50-v1	75.5	61.2	82.7	60.7	56.4	37.0	68.6	37.5
GV*	1000	200	R50-v1	77.2	63.1	82.5	60.9	55.4	37.9	63.2	34.7
RRT (ours)	500	200	R50-v1	77.9	63.5	84.4	62.1	58.8	39.5	71.6	39.5
DELG global	0	0	R50-v2-clean	73.6	60.6	85.7	68.6	51.0	32.7	71.5	44.4
GV	1000	100	R50-v2-clean	78.3	67.2	85.7	69.6	57.9	43.6	71.0	45.7
RRT (ours)	500	100	R50-v2-clean	78.1	67.0	86.7	69.8	60.2	44.1	75.1	49.4
GV*	1000	200	R50-v2-clean	79.2	68.2	85.5	69.6	57.5	42.9	67.2	44.5
RRT (ours)	500	200	R50-v2-clean	79.5	68.6	87.8	71.5	62.5	46.3	77.1	52.3

Table 1. Comparison to geometric verification on Revisited Oxford/Paris [44]. The mAP scores on the Medium ($+\mathcal{R1M}$) and Hard ($+\mathcal{R1M}$) setups are reported. Results marked by * are evaluated by us using the public models provided by [9].

Method	# local desc.	Desc. version	Retrieval	
			Public	Private
DELG global	0	R50-v1	18.3	20.4
GV	1000	R50-v1	20.4	22.3
RRT (ours)	500	R50-v1	21.5	23.1
DELG global	0	R50-v2-clean	22.2	24.2
GV	1000	R50-v2-clean	–	24.3
RRT (ours)	500	R50-v2-clean	24.6	27.0

Table 2. Comparison to geometric verification on the GLDv2 retrieval task [44]. The mAP@100 scores on the public and private test sets are reported.

Position embedding: For the experiments on DELG, where the keypoints are sparsely sampled, we observe no benefit in applying the position embedding and do not use the φ term in Eq. 2. For the experiments on SOP, we find the position embedding is helpful, we posit it is because all the positions are used in this experiment.

Latency and memory For each query, when using an NVIDIA P100 GPU, RRT reranks the top-100 retrieved images in a single forward-pass, which takes 0.36/0.013 seconds on average in the DELG [9]/SOP [53] experiments. In the DELG experiments, we use the same global descriptor but only half (500 out of 1000) of the local descriptors for each image. In other words, the memory footprint is approximately half of that in DELG [9]. We agree that this is still a high cost for large-scale systems. In the future, we’d like to explore techniques that can potentially reduce the memory footprint, e.g. quantization.

5. Results

Here, we demonstrate the effectiveness of the Reranking Transformers (RRTs) across different settings, benchmarks and use cases.

5.1. Comparison with Geometric Verification

We consider geometry verification (GV) as the main baseline. We compare GV and RRT using the same pre-trained DELG [9] descriptors. Following the protocol in [9], given a query, we use its global descriptor to retrieve a set of top-ranked images. The top-100 neighbors are reranked by GV and RRT. We present results on two sets of descriptors: DELG pretrained on GLD v1 [37] and v2-clean [61].

On \mathcal{ROxf} and \mathcal{RPar} , both GV and RRT outperform global-only retrieval, as shown in Table 1. RRT shows further advantages over GV, with much fewer local descriptors. On \mathcal{ROxf} ($+\mathcal{R1M}$), RRT performs on par with GV on the Medium setup and consistently better on the Hard setup. On \mathcal{RPar} ($+\mathcal{R1M}$), RRT consistently outperforms GV. The largest performance gap is achieved on the Hard setup. RRT obtains 2.2 (3.7) absolute improvements over GV on \mathcal{ROxf} (\mathcal{RPar}), when using the “v1” descriptors. We posit that, while GV is effective for sufficiently similar images, it has difficulty handling challenging cases, e.g. large variations in viewpoint. To verify this, we reranked more images (e.g. top-200), resulting in a larger performance gap. RRT obtains 3.4 (8.4) absolute improvement over GV on \mathcal{ROxf} (\mathcal{RPar}), when using the “v1” descriptors.

We present results on the GLDv2 retrieval task [61] in Table 2. Following [9], we report the mAP@100 scores on the public and private test sets. Compared to \mathcal{ROxf} and \mathcal{RPar} , the improvement of applying reranking on GLDv2 becomes smaller. On the other hand, RRT performs consistently better than global-only and GV. When using the “v2-clean” descriptors, the absolute improvements of RRT over global-only (GV) on the private set are 2.8 (2.7).

5.2. Comparison with Query Expansion

Query expansion (QE) [14, 13, 57] is another popular reranking technique for image retrieval. Different from

Method	# Reranked images	Desc. version	Medium				Hard			
			$\mathcal{R}Oxf$	$+\mathcal{R}1M$	$\mathcal{R}Par$	$+\mathcal{R}1M$	$\mathcal{R}Oxf$	$+\mathcal{R}1M$	$\mathcal{R}Par$	$+\mathcal{R}1M$
DELG global		R50-v1	69.7	55.0	81.6	59.7	45.1	27.8	63.4	34.1
α QE		R50-v1	72.9	60.7	<u>83.4</u>	<u>63.7</u>	49.4	33.6	66.1	<u>38.1</u>
RRT (ours)	100	R50-v1	75.5	61.2	82.7	60.7	56.4	37.0	68.6	37.5
RRT (ours)	200	R50-v1	77.9	63.5	84.4	62.1	58.8	39.5	71.6	39.5
RRT (ours)	400	R50-v1	79.2	66.2	86.3	64.0	60.5	42.6	74.1	41.6
α QE		R50-v1	72.9	60.7	83.4	63.7	49.4	33.6	66.1	38.1
α QE + RRT (ours)	100	R50-v1	78.7	66.2	85.6	65.4	59.8	42.1	72.8	43.1
DELG global		R50-v2-clean	73.6	60.6	85.7	68.6	51.0	32.7	71.5	44.4
α QE		R50-v2-clean	76.6	66.4	86.7	<u>72.8</u>	54.6	39.5	73.2	<u>51.2</u>
RRT (ours)	100	R50-v2-clean	78.1	67.0	86.7	69.8	60.2	44.1	75.1	49.4
RRT (ours)	200	R50-v2-clean	79.5	68.6	87.8	71.5	62.5	46.3	77.1	52.3
RRT (ours)	400	R50-v2-clean	80.5	70.6	89.1	73.8	64.2	49.5	78.1	55.6
α QE		R50-v2-clean	76.6	66.4	86.7	72.8	54.6	39.5	73.2	51.2
α QE + RRT (ours)	100	R50-v2-clean	80.4	71.7	88.5	74.8	64.0	50.9	77.7	57.1

Table 3. Comparison to α QE [43] on Revisited Oxford/Paris [44]. The mAP scores on the Medium ($+\mathcal{R}1M$) and Hard ($+\mathcal{R}1M$) setups are reported. We underline the scores of α QE that RRT cannot match by just reranking the top-100 neighbors. RRT consistently outperforms α QE when reranking the top-400 neighbors for each query. Moreover, combining α QE with RRT significantly outperforms using α QE only, showing that RRT and α QE are complementary to each other.

Method	# local desc.	Desc. version	Medium		Hard	
			$\mathcal{R}Oxf$	$\mathcal{R}Par$	$\mathcal{R}Oxf$	$\mathcal{R}Par$
DELG global	0	R50-v1	69.7	81.6	45.1	63.4
ASMK global	1000	R50-v1	71.2	80.8	47.1	61.6
ASMK rerank	1000	R50-v1	71.3	82.6	47.5	66.2
RRT (ours)	500	R50-v1	75.5	82.7	56.4	68.6
DELG global	0	R50-v2-clean	73.6	85.7	51.0	71.5
ASMK global	1000	R50-v2-clean	70.4	80.9	45.8	62.0
ASMK rerank	1000	R50-v2-clean	73.1	86.3	49.3	71.9
RRT (ours)	500	R50-v2-clean	78.1	86.7	60.2	75.1

Table 4. Comparison to Aggregated Selective Match Kernel (ASMK) on Revisited Oxford/Paris [44]. The mAP scores on the Medium and Hard setups are reported.

GV and RRT, QE aggregates the query image and a number of top-ranked neighbors into a new query. This new query is used to rerank all the gallery images rather than the nearest ones as in GV and RRT. We compare RRT with one of the most widely used query expansion methods: α -weighted query expansion (α QE) proposed in [43]. We use the public implementation of α QE released by [46]. α QE has two hyper-parameters: (1) nQE , the number of top-ranked neighbors to aggregate; (2) α , the exponential weight. In [46], they are set as $(nQE, \alpha) = (10, 2.0)$. Our experiment shows that these values do not work out of the box for the DELG descriptors. We tune these parameters on $\mathcal{R}Oxf$ over the ranges: $nQE \in [2, 15]$, $\alpha \in [0.1, 3.0]$, and eventually set them as $(nQE, \alpha) = (2, 0.3)$.

Table 3 shows the results on $\mathcal{R}Oxf$ and $\mathcal{R}Par$. When reranking the top 100 neighbors, the performance of RRT is superior to α QE on five of the eight settings, except for $\mathcal{R}Par$ +Medium, $\mathcal{R}Par$ + $\mathcal{R}1M$ +Medium, $\mathcal{R}Par$ + $\mathcal{R}1M$ +Hard

(underlined numbers). We believe it is because α QE reranks all the gallery images while RRT reranks only 100 neighbors and keeps the ranks of all the other images unchanged. By reranking more neighbors, e.g. 200, 400, we observe that the performance of RRT progressively improves and eventually surpasses α QE by significant margins across all settings. On the Hard setup with the “v1” descriptors, the absolute gains of RRT over α QE on ($\mathcal{R}Oxf$, $\mathcal{R}Oxf$ + $\mathcal{R}1M$, $\mathcal{R}Par$, $\mathcal{R}Par$ + $\mathcal{R}1M$) are (11.1, 9.0, 8.0, 3.5).

We also perform experiments on combining α QE and RRT by reranking the top neighbors produced by α QE. As shown in Table 3, combining α QE and RRT considerably improves over using α QE only, with improvements of (10.4, 8.5, 6.7, 5.0) on the Hard setup of ($\mathcal{R}Oxf$, $\mathcal{R}Oxf$ + $\mathcal{R}1M$, $\mathcal{R}Par$, $\mathcal{R}Par$ + $\mathcal{R}1M$) for the “v1” descriptors. We consider query expansion and RRT are thus complementary.

5.3. Comparison with Aggregated Selective Match Kernel (ASMK)

Aggregated Selective Match Kernel (ASMK) [55] also leverages local descriptors for image retrieval. The key idea is to create a large visual codebook (i.e. filter banks) by clustering the local descriptors. This visual codebook is used to encode the query and gallery images into global descriptors. The clustering and encoding procedures are typically performed offline as they’re relatively time-consuming. Previously, ASMK was mainly considered as a global retrieval technique. In this paper, we treat ASMK as both a global retrieval baseline and a reranking baseline. We use the public implementation of ASMK released by [54]. Following the common practice proposed in [54], we train a codebook of 65,536 visual words on $\mathcal{R}Oxf$ for

Method	Training set	Net	# local desc.	Medium				Hard			
				$\mathcal{R}Oxf$	$+\mathcal{R}1M$	$\mathcal{R}Par$	$+\mathcal{R}1M$	$\mathcal{R}Oxf$	$+\mathcal{R}1M$	$\mathcal{R}Par$	$+\mathcal{R}1M$
<i>(A) Global features</i>											
R-MAC [21]	Landmarks	R101	0	60.9	39.3	78.9	54.8	32.4	12.5	59.4	28.0
GeM [43]	SfM-120k	R101	0	64.7	45.2	77.2	52.3	38.5	19.9	56.3	24.7
GeM-AP [46]	SfM-120k	R101	0	67.5	47.5	80.1	52.5	42.8	23.2	60.5	25.1
DELG [9]	GLDv1	R50	0	69.7	55.0	81.6	59.7	45.1	27.8	63.4	34.1
<i>(B) Local feature aggregation</i>											
DELG-ASMK[54]	Landmarks	R50	1000	67.8	53.8	76.9	57.3	43.1	31.2	55.4	26.4
HOW-ASMK[56]	SfM-120k	R50	1000	78.3	63.6	80.1	58.4	55.8	36.8	60.1	30.7
HOW-ASMK[56]	SfM-120k	R50	2000	79.4	65.8	81.6	61.8	56.9	38.9	62.4	33.7
<i>(C) Global features + Re-ranking</i>											
GeM \uparrow +DSM [51]	SfM-120k	R101	1000	65.3	47.6	77.4	52.8	39.2	23.2	56.2	25.0
DELG [9] + GV	GLDv1	R50	1000	75.1	61.1	82.3	60.5	54.2	36.8	64.9	34.8
DELG [9] + RRT (ours)	GLDv1&v2-clean	R50	500	75.5	61.2	82.7	60.7	56.4	37.0	68.6	37.5
DELG [9] + GV	GLDv2-clean	R50	1000	78.3	67.2	85.7	69.6	57.9	43.6	71.0	45.7
DELG [9] + RRT (ours)	GLDv2-clean	R50	500	78.1	67.0	86.7	69.8	60.2	44.1	75.1	49.4

Table 5. Comparison to the state-of-the-art on Revisited Oxford/Paris [44]. The mAP scores on the Medium and Hard setups are reported.

Method	Desc. dim.	SOP			
		$R@1$	$R@10$	$R@100$	$R@1k$
<i>Global-only retrieval</i>					
Margin [62, 48]	128	76.1	88.4	95.1	98.3
FastAP [8]	128	73.8	88.0	94.9	98.3
XBM [60]	128	80.6	91.6	96.2	98.7
CE [7]	2048	81.1	91.7	96.3	98.8
CO	128	80.7	91.9	96.6	99.0
CO + RRT (frozen)	128	81.8	92.4	96.6	99.0
CO + RRT (finetuned)	128	84.5	93.2	96.6	99.0

Table 6. Results on jointly optimizing the feature extractor and RRT. The R@K (K =1, 10, 100, 1000) scores on the SOP test set [53] are reported.

retrieval experiments on $\mathcal{R}Par$, and vice-versa. We conduct two experiments: a) ASMK global: using ASMK for global retrieval, as in all the previous literature [55, 54, 56]; b) ASMK rerank: using ASMK for image reranking, e.g. reranking the top-100 images from DELG global retrieval.

We present the results on $\mathcal{R}Oxf$ and $\mathcal{R}Par$ in Table 4. ASMK, when used as a global retrieval approach, shows comparable or inferior performance to DELG global. When used as a reranking approach, ASMK further improves over DELG global, showing that they are complementary. The proposed method consistently outperforms ASMK global/rerank in all settings. We posit that compared with the hand-crafted kernel matching paradigm, RRTs learn a more holistic region-wise similarity between the images.

5.4. Feature Learning & RRT: Joint Optimization

To further demonstrate the possibility of jointly optimizing the feature representations and RRTs, we per-

form experiments on the Stanford Online Products (SOP) dataset [53]. We study three models: (1) *CO*: A global retrieval model trained with a contrastive loss [60], using the metric learning protocol, i.e. the global descriptor has a dimension of 128 [48, 8, 49, 47, 60]; (2) *CO + RRT (frozen)*: an RRT model trained on top of *CO*. The pretrained *CO* remains frozen and an extra linear layer is used to reduce the dimension of the local descriptors to 128; (3) *CO + RRT (finetune)*: a model with the same architecture as *CO + RRT (frozen)* but the backbone is also finetuned. It is also initialized by *CO + RRT (frozen)*. During testing, we perform global retrieval using the global descriptor from *CO*. The top-100 neighbors for each query are reranked by either *CO + RRT (frozen)* or *CO + RRT (finetune)*. We present our results along with the results of the most recent metric learning approaches [48, 8, 60, 7] to provide an overview in the context of the state-of-the-art on SOP.

As shown in Table 6, the global *CO* model, which is trained with a contrastive loss using a relatively large batch size, performs surprisingly well. It achieves the same level of accuracy as well-established works on metric learning. This aligns with the recent research on self-supervised learning [11, 23] showing that contrastive loss is very effective for feature learning. *CO + RRT (frozen)* further improves the performance, demonstrating the effectiveness of reranking. Note that, as only the top-100 images are reranked, the $R@100$ and $R@1k$ scores remain unchanged. *CO + RRT (finetuned)* achieves the best reranking performance, with an absolute improvement of 3.8 over the global-only retrieval on $R@1$. We believe it is because jointly optimizing the backbone and our model leads to better local features that are tailored to the reranking tasks.

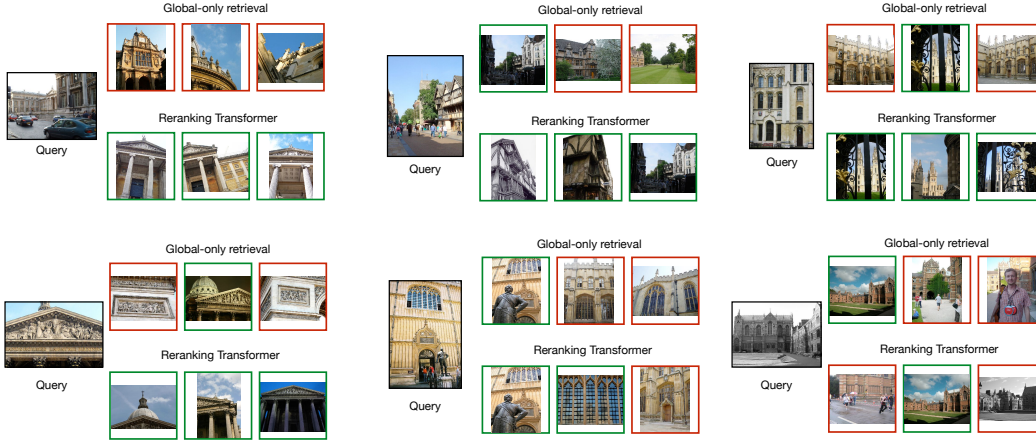


Figure 3. Qualitative examples from Revisited Oxford/Paris [44]. For each query, the top-3 neighbors ranked by the global retrieval and reranked by RRT are presented. Correct/incorrect neighbors are marked with green/red borders.

5.5. Comparison with the State-of-the-Art

In Table 5, we compare the proposed method with the state-of-the-art on the \mathcal{ROxf} (+ $\mathcal{R1M}$) and \mathcal{RPar} (+ $\mathcal{R1M}$) benchmarks. We include the most recent instance recognition/retrieval models in three different groups: (A) Retrieval by global features only; (B) Retrieval by local feature aggregation; (C) Retrieval by combining global features with reranking. While our method performs favorably on most of the settings (except for \mathcal{ROxf} , $\mathcal{ROxf}+\mathcal{R1M}$), these results include comparisons to methods that differ on the training data, the CNN backbones, and the number of local features, etc. For context we provide as much information about each method regarding these differences.

In Fig. 3, we present qualitative examples on image retrieval when using only global features and when using our full reranking approach. While global-only retrieval can return highly similar images in general, reranking by global/local descriptors captures a more fine-grained matching between images, leading to better recognition accuracy.

Finally, Table 7 shows a comparison with SuperGlue [50]. Similar as in geometry verification, the number of inlier correspondences predicted by SuperGlue is used as the similarity score. As SuperGlue is not designed for global retrieval, we use the pretrained DELG v1/v2-clean descriptors for global retrieval, so that SuperGlue and our method are evaluated on the same initial ranking lists. We use the SuperGlue pretrained on MegaDepth [29], which contains 130K images with dense annotations. Note that both SuperGlue and our work are trained on datasets (MegaDepth vs a subset of GLDv2-clean) that have different data distributions than the test sets (Revisited Oxford/Paris). SuperGlue leverages SuperPoint [15] as the backbone, and uses different feature dimensions and keypoint numbers as our work. As shown in Table 7, reranking by both SuperGlue and our work significantly improve the performance over the global-only retrieval. RRT gains

Method	Desc. version	# local desc.	Desc. dim.	Medium \mathcal{ROxf}	Medium \mathcal{RPar}	Hard \mathcal{ROxf}	Hard \mathcal{RPar}
DELG global	R50-v1	0	-	69.7	81.6	45.1	63.4
SuperGlue [50]	SuperPoint [15]	500	256	72.6	81.9	49.7	62.4
SuperGlue [50]	SuperPoint [15]	1024	256	74.4	82.3	55.2	64.6
RRT (ours)	R50-v1	500	128	75.5	82.7	56.4	68.6
DELG global	R50-v2-clean	0	-	73.6	85.7	51.0	71.5
SuperGlue [50]	SuperPoint [15]	500	256	76.2	85.9	54.6	68.2
SuperGlue [50]	SuperPoint [15]	1024	256	78.3	86.2	60.0	70.4
RRT (ours)	R50-v2-clean	500	128	78.1	86.7	60.2	75.1

Table 7. Comparison to the pretrained SuperGlue model [50] on Revisited Oxford/Paris [44]. The SuperGlue model is pretrained on MegaDepth [29] with SuperPoint [15] as the backbone. The mAP scores on the Medium and Hard setups are reported.

larger improvements on most of the settings, especially for challenge cases. We provide more experiments on SuperGlue in the supplemental material.

6. Conclusion

We introduce *Reranking Transformers* (RRTs) for instance image retrieval. We show that RRTs outperform prior reranking approaches across a variety of settings. Compared to geometric verification [42] and other local feature based methods [55], RRTs use fewer descriptors and can be parallelized such that reranking requires a single forward pass. We also demonstrate that, unlike previous reranking approaches, RRTs can be optimized jointly with the feature extractor, leading to further gains.

Acknowledgements: This project was funded by a gift from eBay Inc to the University of Virginia. We also thank Robinson Piramuthu for his encouragement in starting this collaboration, Dmytro Mishkin for suggesting comparisons with SuperGlue and anonymous reviewers for their feedback. V.O. was also supported by a Facebook Research Award, a gift from Leidos Inc, and NSF Award IIS-2045773.

References

- [1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 3
- [2] Artem Babenko and Victor S. Lempitsky. Aggregating deep convolutional features for image retrieval. In *Int. Conf. Comput. Vis.*, 2015. 2
- [3] Artem Babenko, Anton Slesarev, Alexander Chigorin, and Victor S. Lempitsky. Neural codes for image retrieval. In *Eur. Conf. Comput. Vis.*, 2014. 1, 2
- [4] S. Bai, X. Bai, Q. Tian, and L. J. Latecki. Regularized diffusion process on bidirectional context for object retrieval. *IEEE Trans. Pattern Anal. Mach. Intell.*, 41(5):1213–1226, 2019. 2
- [5] Song Bai, Peng Tang, Philip H.S. Torr, and Longin Jan Latecki. Re-ranking via metric fusion for object retrieval and person re-identification. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019. 2
- [6] Irwan Bello, Barret Zoph, Ashish Vaswani, Jonathon Shlens, and Quoc V. Le. Attention augmented convolutional networks. In *Int. Conf. Comput. Vis.*, 2019. 2
- [7] Malik Boudiaf, J'erome Rony, Imtiaz Masud Ziko, Éric Granger, Marco Pedersoli, Pablo Piantanida, and Ismail Ben Ayed. Metric learning: cross-entropy vs. pairwise losses. In *Eur. Conf. Comput. Vis.*, 2020. 4, 7
- [8] Fatih Cakir, Kun He, Xide Xia, Brian Kulis, and Stan Sclaroff. Deep metric learning to rank. In *IEEE Conf. Comput. Vis. Pattern Recog.*, June 2019. 4, 7
- [9] Bingyi Cao, Andre Araujo, and Jack Sim. Unifying deep local and global features for image search. In *Eur. Conf. Comput. Vis.*, 2020. 1, 2, 4, 5, 7
- [10] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Eur. Conf. Comput. Vis.*, 2020. 2, 3
- [11] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Everest Hinton. A simple framework for contrastive learning of visual representations. In *Int. Conf. Mach. Learn.*, 2020. 7
- [12] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Universal image-text representation learning. In *Eur. Conf. Comput. Vis.*, pages 104–120. Springer, 2020. 2
- [13] O. Chum, A. Mikulík, M. Perdoch, and J. Matas. Total recall ii: Query expansion revisited. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 889–896, 2011. 2, 5
- [14] O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman. Total recall: Automatic query expansion with a generative feature model for object retrieval. In *Int. Conf. Comput. Vis.*, 2007. 2, 5
- [15] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *IEEE Conf. Comput. Vis. Pattern Recog. Worksh.*, 2018. 8
- [16] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. 2, 3, 4
- [17] M. Donoser and H. Bischof. Diffusion processes for retrieval revisited. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2013. 2
- [18] 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 *Int. Conf. Learn. Represent.*, 2021. 2
- [19] Mihai Dusmanu, Ignacio Rocco, Tomáš Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. D2-net: A trainable CNN for joint detection and description of local features. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019. 2
- [20] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981. 1
- [21] Albert Gordo, Jon Almazán, Jérôme Revaud, and Diane Larlus. End-to-end learning of deep visual representations for image retrieval. *Int. J. Comput. Vis.*, 2017. 2, 7
- [22] Albert Gordo, Filip Radenovic, and Tamara Berg. In *Eur. Conf. Comput. Vis.*, 2020. 2
- [23] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020. 7
- [24] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 770–778, 2016. 2, 4
- [25] Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, Teddy Furon, and Ondrej Chum. Efficient diffusion on region manifolds: Recovering small objects with compact CNN representations. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017. 2
- [26] Yannis Kalantidis, Clayton Mellina, and Simon Osindero. Cross-dimensional weighting for aggregated deep convolutional features. In *Eur. Conf. Comput. Vis. Worksh.*, 2015. 2
- [27] Jack Lanchantin, Tianlu Wang, Vicente Ordonez, and Yanjun Qi. General multi-label image classification with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16478–16488, 2021. 2
- [28] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*, 2019. 2
- [29] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018. 8
- [30] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *1907.11692*, 2019. 2, 3

- [31] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *Int. Conf. Learn. Represent.*, 2019. 4
- [32] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.*, 2004. 2
- [33] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Adv. Neural Inform. Process. Syst.*, pages 13–23. Curran Associates, Inc., 2019. 2
- [34] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(10):1615–1630, 2005. 2
- [35] Tony Ng, Vassileios Balntas, Yurun Tian, and Krystian Mikolajczyk. SOLAR: second-order loss and attention for image retrieval. In *Eur. Conf. Comput. Vis.*, 2020. 2
- [36] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *IEEE Conf. Comput. Vis. Pattern Recog.*, volume 2, pages 2161–2168, 2006. 2
- [37] Hyeonwoo Noh, Andre Araujo, Jack Sim, and Bohyung Han. Image retrieval with deep local features and attention-based keypoints. In *Int. Conf. Comput. Vis.*, 2017. 1, 2, 4, 5
- [38] Shanmin Pang, Jin Ma, Jianru Xue, Jihua Zhu, and Vicente Ordonez. Deep feature aggregation and image re-ranking with heat diffusion for image retrieval. *IEEE Trans. Multimedia*, 21(6):1513–1523, 2019. 2
- [39] Shanmin Pang, Jihua Zhu, Jiaying Wang, Vicente Ordonez, and Jianru Xue. Building discriminative cnn image representations for object retrieval using the replicator equation. *Pattern Recognition*, 83:150–160, 2018. 2
- [40] Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, and Alexander Ku. Image transformer. In *Int. Conf. Mach. Learn.*, 2018. 2
- [41] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. pages 8024–8035. 2019. 4
- [42] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1–8, 2007. 1, 2, 8
- [43] Filip Radenovic, Giorgos Tolias, and Ondrej Chum. Fine-tuning CNN image retrieval with no human annotation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2018. 2, 6, 7
- [44] Filip Radenović, Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondřej Chum. Revisiting oxford and paris: Large-scale image retrieval benchmarking. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018. 2, 4, 5, 6, 7, 8
- [45] Prajit Ramachandran, Niki Parmar, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jon Shlens. Stand-alone self-attention in vision models. In *Adv. Neural Inform. Process. Syst.*, volume 32, pages 68–80, 2019. 2
- [46] Jerome Revaud, Jon Almazan, Rafael S. Rezende, and Cesar Roberto de Souza. Learning with average precision: Training image retrieval with a listwise loss. In *Int. Conf. Comput. Vis.*, 2019. 6, 7
- [47] Karsten Roth, Biagio Brattoli, and Bjorn Ommer. Mic: Mining interclass characteristics for improved metric learning. In *Int. Conf. Comput. Vis.*, October 2019. 4, 7
- [48] Karsten Roth, Timo Milbich, Samarth Sinha, Prateek Gupta, Björn Ommer, and Joseph Paul Cohen. Revisiting training strategies and generalization performance in deep metric learning. In *Int. Conf. Mach. Learn.*, July 2020. 4, 7
- [49] Artsiom Sanakoyeu, Vadim Tschernezki, Uta Buchler, and Bjorn Ommer. Divide and conquer the embedding space for metric learning. In *IEEE Conf. Comput. Vis. Pattern Recog.*, June 2019. 4, 7
- [50] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020. 2, 8
- [51] Oriane Siméoni, Yannis Avrithis, and Ondrej Chum. Local features and visual words emerge in activations. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019. 1, 2, 7
- [52] Josef Sivic and Andrew Zisserman. Video google: A text retrieval approach to object matching in videos. In *Int. Conf. Comput. Vis.*, pages 1470–1477 vol.2, 2003. 1, 2
- [53] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016. 4, 5, 7
- [54] Marvin Teichmann, Andre Araujo, Menglong Zhu, and Jack Sim. Detect-to-retrieve: Efficient regional aggregation for image search. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019. 2, 6, 7
- [55] Giorgos Tolias, Yannis Avrithis, and Hervé Jégou. Image search with selective match kernels: Aggregation across single and multiple images. *Int. J. Comput. Vis.*, 116(3):247–261, 2016. 1, 2, 6, 7, 8
- [56] Giorgos Tolias, Tomas Jenicek, and Ondřej Chum. Learning and aggregating deep local descriptors for instance-level recognition. In *Eur. Conf. Comput. Vis.*, 2020. 1, 2, 7
- [57] Giorgos Tolias and Hervé Jégou. Visual query expansion with or without geometry: Refining local descriptors by feature aggregation. *Pattern Recognition*, 47(10):3466 – 3476, 2014. 2, 5
- [58] Giorgos Tolias, R. Sivic, and H. Jégou. Particular object retrieval with integral max-pooling of cnn activations. In *Int. Conf. Learn. Represent.*, 2016. 2
- [59] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Adv. Neural Inform. Process. Syst.*, pages 5998–6008. 2017. 2
- [60] Xun Wang, Haozhi Zhang, Weilin Huang, and Matthew R. Scott. Cross-batch memory for embedding learning. In *IEEE Conf. Comput. Vis. Pattern Recog.*, June 2020. 4, 7
- [61] Tobias Weyand, Andre Araujo, Bingyi Cao, and Jack Sim. Google landmarks dataset v2 – a large-scale benchmark for instance-level recognition and retrieval. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020. 2, 4, 5

- [62] Chao-Yuan Wu, R. Manmatha, Alexander J. Smola, and Philipp Krahenbuhl. Sampling matters in deep embedding learning. In *Int. Conf. Comput. Vis.*, Oct 2017. [4](#), [7](#)
- [63] S. Zhang, M. Yang, T. Cour, K. Yu, and D. N. Metaxas. Query specific rank fusion for image retrieval. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(4):803–815, 2015. [2](#)
- [64] Hengshuang Zhao, Jiaya Jia, and Vladlen Koltun. Exploring self-attention for image recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020. [2](#)