Global Features are All You Need for Image Retrieval and Reranking

Shihao Shao[1]*† Kaifeng Chen[2] Arjun Karpur[2] Qinghua Cui[1] André Araujo[2]
Bingyi Cao[2]*†
[1]Peking University [2]Google Research

Abstract

Image retrieval systems conventionally use a two-stage paradigm, leveraging global features for initial retrieval and local features for reranking. However, the scalability of this method is often limited due to the significant storage and computation cost incurred by local feature matching in the reranking stage. In this paper, we present SuperGlobal, a novel approach that exclusively employs global features for both stages, improving efficiency without sacrificing accuracy. SuperGlobal introduces key enhancements to the retrieval system, specifically focusing on the global feature extraction and reranking processes. For extraction, we identify sub-optimal performance when the widely-used ArcFace loss and Generalized Mean (GeM) pooling methods are combined and propose several new modules to improve GeM pooling. In the reranking stage, we introduce a novel method to update the global features of the query and top-ranked images by only considering feature refinement with a small set of images, thus being very compute and memory efficient. Our experiments demonstrate substantial improvements compared to the state of the art in standard benchmarks. Notably, on the Revisited Oxford+1M Hard dataset, our single-stage results improve by 7.1%*, while our two-stage gain reaches* 3.7% *with a strong* $64,865\times$ *speedup. Our two-stage system surpasses the current single-stage state-of-the-art by* 16.3%*, offering a scalable, accurate alternative for high-performing image retrieval systems with minimal time overhead.*
Code: https://github.com/ShihaoShao-GH/SuperGlobal.

1. Introduction

Image retrieval systems are tasked with searching large databases for visual contents similar to a query image. Generally, the search process consists of two stages. First, an efficient method sorts database images according to estimated high-level similarity to the query. Then, in the reranking stage, the most relevant database images found in the first stage undergo a more comprehensive matching process against the query, to return an improved ranked list of results.

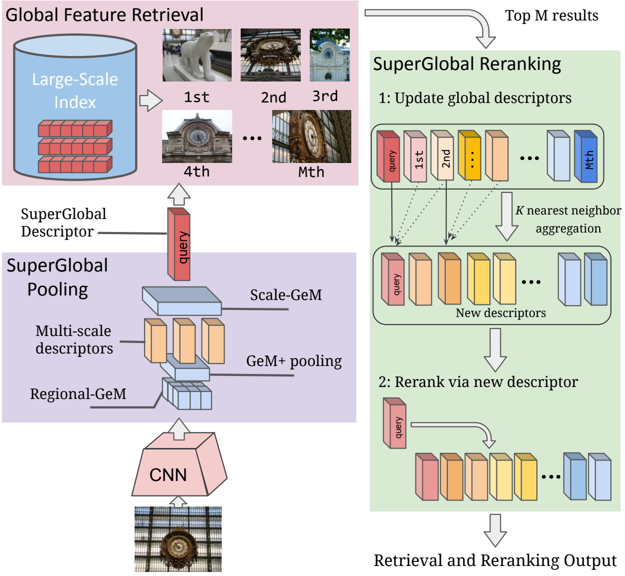

Figure 1: We introduce **SuperGlobal**, a novel method for image retrieval and reranking which relies solely on global image features. SuperGlobal leverages several improvements to the Generalized Mean (GeM) pooling function, across regions and scales, as indicated in the purple box on the left. Our reranking process, illustrated on the right green box, refines the global feature representation based on the query and top retrieved images to produce a more relevant set of results.

In modern implementations, the first stage is instantiated with deep learning-based global features, which has received substantial attention in the past few years [35, 37, 26, 48]. The reranking stage is commonly executed via geometric matching of local image features [30, 32, 4, 8], which provides information on the spatial consistency between the query and a given database image.

A recent trend in this area is on leveraging sophisticated matching processes at the reranking stage, *e.g*. transformers [40] or 4D correlation networks [23], which have led to remarkable improvements in the quality of retrieved results. However, this has come at a significant cost, where reranking latency takes several seconds per query and requires more

*Both authors contributed equally to this paper.
†Co-corresponding authors. Emails:
shaoshihao@pku.edu.cn, bingyi@google.com

than 1MB of memory per database image – making these approaches challenging to scale to large repositories. Our work directly tackles this limitation by proposing the first method fully based on global image features for both stages. In addition, we rethink pooling techniques and propose modules to improve global feature extraction. An overview of our method, SuperGlobal, is presented in Figure 1. More specifically, we introduce the following contributions.

Contributions:

(1) We propose improvements to the core global feature representation, based on enhanced pooling strategies. We point out undesired training behavior when learning global features combining GeM pooling [35] and ArcFace loss [11], and introduce a simple and effective solution to this problem with new pooling modules, including regional and multi-scale techniques.

(2) We introduce a very efficient and effective reranking process, based solely on global features, that is able to adapt the representation of the query and top-ranked database images on the fly in order to better estimate their similarities. This method does away with any need for expensive local features and is inherently very scalable since the features used in both search stages are the same.

(3) Experiments on standard image retrieval benchmarks showcase the effectiveness of our methods, establishing new state-of-the-art results across the board. We boost single-stage results on Revisited Oxford+1M Hard [34] by 7.1%. But even more impressively, our simple reranking mechanism outperforms previous complex methods by 3.7% on the same dataset, while being more than four orders of magnitude faster and requiring 170× less memory.

2. Related Work

Image retrieval methods. Early work in image retrieval leveraged hand-crafted local features [24, 7] as a core building block. While some papers proposed to retrieve directly based on local features [25, 24, 28], others used them to construct global representations, based on Bag-of-Words and similar techniques [39, 19, 20, 42]. Modern systems have revisited these image retrieval techniques with deep learning based components, *e.g.*, deep local feature-based retrieval [27], deep local feature aggregation [41, 43, 46] or deep global feature modeling [6, 13, 37, 26, 48]. A recent survey in this area can be found in [9].

Global feature pooling. In particular, a critical aspect that has been studied for global feature learning is on how to properly pool contributions of image features from different regions into a single high-dimensional vector. SPoC [5] proposed sum pooling of convolutional features, while [36] introduced max pooling, which was later approximated by integral max pooling in R-MAC [44]. Along a similar line, CroW [21] introduced cross-dimensional weighted sum pooling. NetVLAD [2] introduced an aggregation inspired by the VLAD method [19]. Generalized Mean (GeM) pooling [35] is today considered the state-of-the-art method in this area, being used in several recent papers [23, 8, 48]. A key contribution of our paper is to revisit global pooling methods, by pointing out the sub-optimal behavior of GeM when using a popular training loss, and by improving regional and multi-scale pooling. Note that R-MAC [44] had explored regional pooling, with max pooling over regions and sum pooling of these regional descriptors. In contrast, we apply the more modern GeM pooling across regions and scales to achieve enhanced performance.

Loss functions for image retrieval. Several types of loss functions have been developed to enhance instance-level discriminability, which is required in image retrieval systems. Earlier work [2, 13, 35] in this area relied on ranking-based losses such as contrastive [10] or triplet [38]. More advanced ranking losses based on differentiable versions of Average Precision (AP) [16] have also demonstrated strong results [37]. A recent trend is to leverage margin-based classification loss functions tuned to this problem, such as ArcFace [11], CosFace [45] or CurricularFace [18] – these have been adopted in image retrieval systems such as [23, 8, 48]. In this work, we point out a critical issue when these margin-based classification losses are coupled with GeM pooling – which we fix with new pooling modules.

Reranking for image retrieval. The reranking of image retrieval results has been traditionally accomplished by local feature matching and Geometric Verification (GV) [30, 32, 4], most often coupled to RANSAC [12]. Modern deep local features [27, 8] have also been used in this manner. A more recent trend is to employ heavier techniques for reranking, based on transformers [40] or dense 4D correlation networks [23]. While achieving high performance, these incur substantial storage and compute costs due to the need to store local features and feed them through complex models. Contrary to this trend, we propose a much simpler reranking technique where only global features are needed – costing orders of magnitude less than the current state-of-the-art solution [23] but still achieving higher accuracy.

3. Improving Global Features

3.1. Background

GeM pooling. Generalized Mean (GeM) pooling [35] is a module that provides a generalized capability for feature aggregation. GeM pooling is widely adopted in ResNet [17] (RN for short) models for image retrieval [23, 8, 48], followed by a fully-connected layer [13], to whiten the aggregated representation. Formally, we denote the feature map from deep convolution layers by $D \in \mathcal{R}^{H_d \times W_d \times C_d}$ and a fully-connected whitening layer with weights and bias as $W \in \mathcal{R}^{C_g \times C_d}$ and $b \in \mathcal{R}^{C_g}$, where C_d and C_g are the channel dimensions of the output from the convolution layer

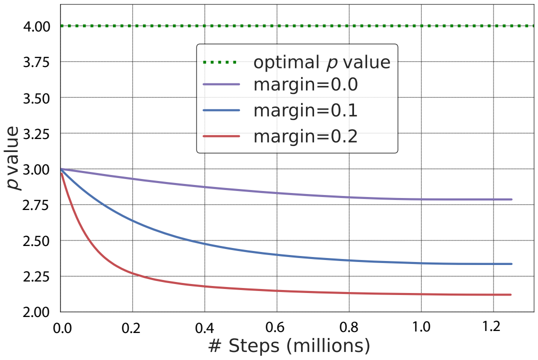

Figure 2: **Trainable GeM pooling p values during DELG training** for different ArcFace margin values, as compared to the optimal p value (4.0). Note that larger margins cause p to deviate further from its optimal value.

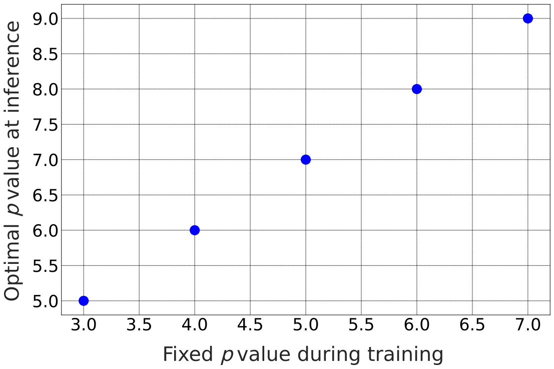

Figure 3: **Optimal p values at inference**, shown on the y-axis, for different fixed GeM pooling p values during DELG training. We note that the optimal p for inference is larger than the fixed p used for training.

and global features, respectively. These two components, GeM pooling and the whitening layer, produce the global feature $g \in \mathcal{R}^{C_g}$ by:

$$g = W \left(\frac{1}{H_d W_d} \sum_{h,w} D_{h,w}^p \right)^{1/p} + b, \qquad (1)$$

where p denotes the generalized mean power.

SoftMax-based loss functions with margin penalties. ArcFace [11] applies a geometric space penalty to expand the angular margin between different classes while gathering the same-class embedding to the center, therefore making it suitable for standard retrieval tasks [48, 8]. CurricularFace [18] proposes to further improve angular margin losses by embedding curriculum training into the loss function and has the ability to adaptively adjust the relative importance of easy and hard samples during the course of training, which has been used in recent image retrieval work [23].

Multi-scale inference. Multi-scale inference is one of the commonly used methods to aggregate features from different scales to further improve the performance of image retrieval, previous papers [8, 48, 23] commonly average the embeddings from different scales.

3.2. Suboptimal GeM Pooling with Margin-based Losses

We observe that combining CurricularFace or ArcFace loss with GeM pooling systematically causes the trainable p value in GeM pooling to converge to a lower value w.r.t. its optimal value for image retrieval. In Figure 2, we show such phenomena when training DELG [8] with learnable p values initialized to 3.0, on GLDv2-clean [47]. The optimal p value in the test split of GLDv2-retrieval, found by simple grid search to inform the best possible value, is marked with a dotted green line (in this case, the optimal p value for the converged model was found to be the same for these three runs). During training, p values keep decreasing from its original value, and are further away from the optimal value of 4.0. Moreover, higher angular margin causes p to deviate further from its optimal value. In the cases of using fixed p for GeM pooling during training, the optimal p during inference could also be different from that in training. We examine the optimal p values for inference by grid search in the test split of GLDv2-retrieval for different fixed p values in training DELG models [8] on GLDv2-clean [47] and find that optimal p at inference is always higher, as shown in Figure 3. SOLAR [26] pointed out a similar phenomenon for models trained with the contrastive loss, while the underlying reason was not explained.

In this section, we provide an intuition on the change of p over the course of training based on our empirical study as follows. A high p value generally forces a small portion of features to dominate the aggregation process. On the contrary, a lower p leads to the opposite behavior: *e.g.*, for $p = 1$, equivalent to average pooling, all features contribute equally. At the beginning of training, when the features are not refined, a lower p value is preferred in order to gather more information; this is supported by the observation that p goes down rapidly at the start of the training in Figure 2. In the later stage of training, when the features are further refined, focusing on critical features rather than all features may further improve the model performance. However, due to the decaying learning rate, p value is not allowed to go up although higher p may be preferred in this case. This aligns with Figure 2, where p slowly converges later at training.

To conclude, we consistently observe that margin-based losses lead to a sub-optimal p value, resulting in the degradation of retrieval performance. This finding provides evidence of improvements and inspires future work for several state-of-the-art models, *e.g.* CVNet [23] which uses CurricularFace loss function and DELG [8] with ArcFace loss. We introduce a set of modules specifically designed to optimize pooling for image retrieval in the following section.

3.3. SuperGlobal Pooling

In this section, we revisit global feature pooling and propose three new modules to enhance retrieval capabilities: GeM+, Scale-GeM and Regional-GeM – which are illustrated in the purple box in Figure 1.

GeM+. As discussed in the previous subsection, GeM's p value becomes sub-optimal with margin-based softmax losses. Thus, we propose a method that starts by training with margin-based loss, then introduces a parameter tuning stage that will adjust p in an efficient manner. We find that in practice this tuning stage leads to the optimal value in a consistent manner for many datasets. This approach is named GeM+ and seeks to find the optimal p value of GeM pooling for image retrieval.

Regional-GeM. When adopting GeM for global pooling in image retrieval, we expect it to amplify discriminative information when aggregating the features to the final embedding. However, in addition to discriminative information at the global level, regional information such as object shape and arrangement can be important for distinguishing between different instances. Such fine-grained details may not be captured robustly when simply pooling at the global level. Therefore, besides using GeM pooling, we further adjust aggregation in order to incorporate regional information. We refer to this method as Regional-GeM.

We perform regional aggregation by adapting the L_p pooling approach [15] to our network, with parameter p_r. This can be viewed as a version of GeM pooling which acts on a regional level. In this setup, activations from the feature map D are aggregated in a convolutional manner, resulting in a new feature map, $M \in \mathcal{R}^{H_d \times W_d \times C_d}$. We then combine M and D to produce a more robust feature map, obtaining an improved global feature as:

$$g_r = W \left(\frac{1}{H_d W_d} \sum_{h,w} (\frac{M_{h,w} + D_{h,w}}{2})^p \right)^{1/p} + b. \quad (2)$$

With this formulation, we incorporate both regional information ($M_{h,w}$ produced by L_p pooling with parameter p_r) and global information ($D_{h,w}$ produced by the original convolutional layer) in GeM pooling. This module is integrated into our model without the need for additional training, leveraging the parameter p obtained by the GeM+ process.

Scale-GeM. Though averaging multi-scale features can be effective [8, 48, 23], a more generalized multi-scale aggregation may unlock larger retrieval gains. With this motivation, we explore the application of GeM for enhanced multi-scale feature inference, and we refer to this as Scale-GeM.

GeM pooling can be applied before or after a fully-connected whitening layer. Our preliminary experiments applying it prior to projection yield sub-optimal performance, so we proceed by first extracting each scale's global feature according to Equation 2. Naively applying GeM pooling to such global features could fail due to the possible negative values in the features to be pooled. To address this issue, we consider a modified version of GeM designed for multi-scale inference as follows:

$$g_{ms} = \left(\frac{1}{N} \sum_{s=1}^{N} (g_s + \zeta_s)^{p_{ms}} \right)^{1/p_{ms}} - \zeta_s, \quad (3)$$

where $\zeta_s = -min(g_s)$ denotes a shift of each scale's global feature g_s, N denotes the number of scales and p_{ms} is the multi-scale power parameter used in aggregation.

4. Reranking with Global Features

4.1. Refining Global Feature for Reranking

Robust image representations are critical for the accuracy of image retrieval. Combining the representations of similar images with that of the original image into an expanded representation that is then reissued as the query is a technique widely used to refine global features, generally leading to increased recall [14, 3]. Query expansion (QE) [14] is an example, as it replaces the original representation of the query image by its expanded version, which is then used to search better images in the database. On the other hand, database-side augmentation (DBA) [13] is a method to apply QE to each image in the database. The key idea is that visually similar images are highly likely to contain the same object from different viewpoints and illumination conditions. Feature refinement with these images improves the robustness of the image representation. It also emphasizes the key features of the object of interest, which further improves the representations. QE and DBA methods are very powerful but suffer from high cost: QE has to issue a new query against the entire database; DBA requires comparing all database images against each other, which can be infeasible in large scale. Furthermore, adding a new image to the database with DBA requires querying it against the entire database.

Reranking is usually conducted on the top-M retrieved database images, where M is much smaller than the database size – making it feasible to apply feature refinement for each of these images on the fly, to then issue the updated query against the M retrieved images with the updated representations. Inspired by QE and DBA, our SuperGlobal reranking proposes a simple but effective method to aggregate information between the top-ranked images and the query, to update their image representations. Unlike previous QE/DBA work [14, 3] that generally focuses on improving the features for a better recall, our work aims to refine the features for

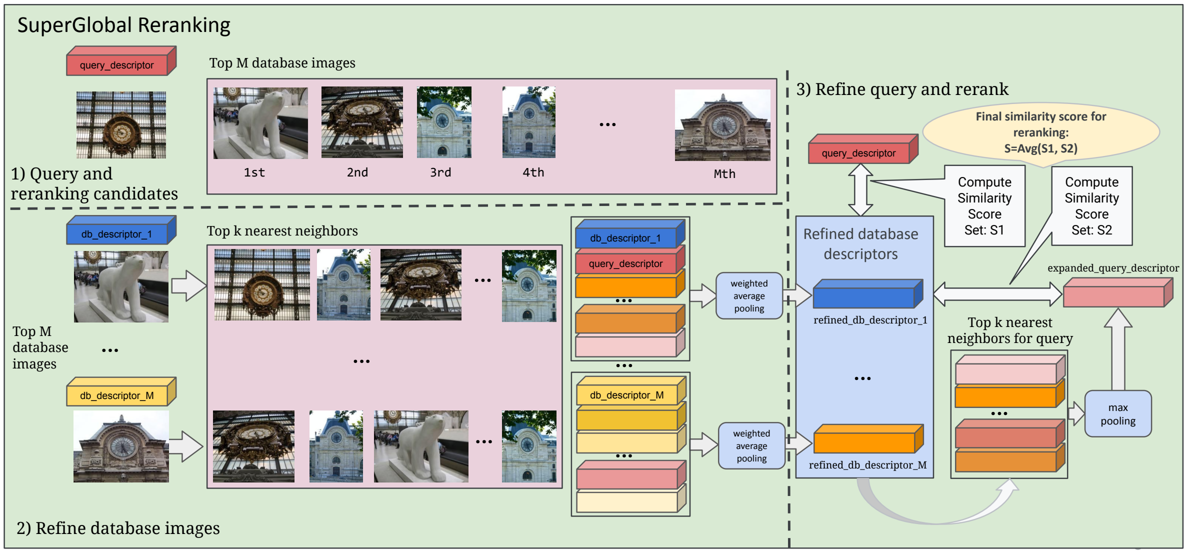

Figure 4: When reranking with global features, our system first performs feature aggregation for both query and top-M retrieved images and then runs reranking via updated global features. More details can be found in Section 4.2.

higher precision, since the reranking is performed on the top-M results only.

Selecting the candidate images for feature refinement may be challenging, since we don't have guarantees that they are actually relevant to the query. False positives may harm the expanded representation. Besides, the method to aggregate the features to reinforce the information shared between them and inject new information not available in the original representation is unclear. Our work addresses these challenges and proposes a reranking method only via refined global features, as described in the following.

4.2. SuperGlobal Reranking

SuperGlobal leverages GeM pooling to refine the global features for a given query and its top-M retrieved images on the fly, and then reranks them via the updated descriptors, as illustrated in Figure 4. Different from previous studies and given that feature refinement runs at query time, the query image is also included as a candidate when refining the features for the database images. This helps to narrow the focus on query-specific feature refinement. The design details are discussed as follows.

Top-K nearest neighbors as refinement candidates. For a given query image, we retrieve the top-M images based on the global descriptors, where M is a constant and typically below 1000. Then for the query and the M images, we fetch the top-K nearest neighbors via global feature similarity, which are the candidates for the feature refinement, where K is a constant and usually $K \leq 10$.

Feature refinement via GeM pooling. SuperGlobal reranking leverages GeM pooling for feature refinement. As previously mentioned, if there are false positives in the nearest neighbors, they may not be helpful but instead harmful to the expanded representation. Without strong Geometric Verification of local features to select true positives, the top-K nearest neighbors could potentially contain false positives. SuperGlobal proposes effective strategies for database and query side separately as illustrated in Figure 4.

For the database side, we propose a weighted pooling approach, with the global similarity score as the weight with additional multiplier factor β. After weighting the features, we demonstrate that applying average pooling ($p = 1$) on top is the most effective for the database images. That is, $g_{dr} = (g_d + \sum_{i=1}^{K}(g_d \cdot g_i)\beta g_i)/(1 + \sum_{i=1}^{K}(g_d \cdot g_i)\beta)$, where g_d is the original global feature of the database image, g_{dr} is the refined global feature we get and g_i is the i-th most similar global feature to g_d.

For the query side, we apply GeM pooling to the refined features of the top K retrieved database images to produce an expanded global descriptor g_{qe}, and we find the optimal parameter p is greater than 10, thus max pooling is applied (since, when $p \to \infty$, GeM pooling becomes max pooling). Both the original and the expanded descriptors of the query image are then used to compute the similarity scores for the final reranking, as follows.

Reranking with refined representations. Each query image possesses its original representation and the expanded representation. We compute the similarity scores $S1$ between each original descriptor g_q and refined global descriptors g_{dr} for each database image. We also compute another set of similarity scores $S2$ between the expanded query descriptor g_{qe} and each g_d. In the end, we average $S1$ and $S2$

Method	Medium				Hard			
	$\mathcal{R}$Oxf	$\mathcal{R}$Oxf+1M	$\mathcal{R}$Par	$\mathcal{R}$Par+1M	$\mathcal{R}$Oxf	$\mathcal{R}$Oxf+1M	$\mathcal{R}$Par	$\mathcal{R}$Par+1M
Global feature retrieval								
RN50-DELG [8]	73.6	60.6	85.7	68.6	51.0	32.7	71.5	44.4
RN101-DELG [8]	76.3	63.7	86.6	70.6	55.6	37.5	72.4	46.9
RN50-DOLG [48]	80.5	**76.6**	89.8	80.8	58.8	52.2	77.7	62.8
RN101-DOLG [48]	81.5	77.4	91.0	83.3	61.1	54.8	80.3	66.7
RN50-CVNet [23]	81.0	72.6	88.8	79.0	62.1	50.2	76.5	60.2
RN101-CVNet [23]	80.2	74.0	90.3	80.6	63.1	53.7	79.1	62.2
RN50-SuperGlobal (No reranking) **[ours]**	**83.9**	74.7	**90.5**	**81.3**	**67.7**	**53.6**	**80.3**	**65.2**
RN101-SuperGlobal (No reranking) **[ours]**	**85.3**	**78.8**	**92.1**	**83.9**	**72.1**	**61.9**	**83.5**	**69.1**
Global feature retrieval + Local feature reranking								
RN50-DELG (GV rerank top 100) [8]	78.3	67.2	85.7	69.6	57.9	43.6	71.0	45.7
RN101-DELG (GV rerank top 100) [8]	81.2	69.1	87.2	71.5	64.0	47.5	72.8	48.7
RN50-CVNet (Rerank top 400) [23]	87.9	**80.7**	90.5	82.4	75.6	**65.1**	80.2	67.3
RN101-CVNet (Rerank top 400) [23]	87.2	81.9	91.2	83.8	75.9	67.4	81.1	69.3
SuperGlobal retrieval and reranking								
RN50-SuperGlobal (Rerank top 400) **[ours]**	**88.8**	80.0	**92.0**	**83.4**	**77.1**	64.2	**84.4**	**68.7**
RN101-SuperGlobal (Rerank top 400) **[ours]**	**90.9**	**84.4**	**93.3**	**84.9**	**80.2**	**71.1**	**86.7**	**71.4**

Table 1: Results (% mAP) on the $\mathcal{R}$Oxford and $\mathcal{R}$Paris datasets (and their large-scale versions $\mathcal{R}$Oxf+1M and $\mathcal{R}$Par+1M), with both Medium and Hard evaluation protocols. The best scores for RN50 and RN101, with and without reranking, are highlighted in **bold black** and **bold blue**, respectively.

similarity scores for the final reranking. Given the fact that today's large-scale databases may contain billions of images, previous QE/DBA methods are much more costly compared to our approach, which has time complexity of $\mathcal{O}(M^2)$ and is extremely efficient at reranking.

5. Experiments

5.1. Experimental Setup

Common setting. Our proposed methods can be applied to any model in a plug-in style. Here, we adopt our methods to the well-known structure CVNet [23] with pre-trained weights downloaded from their GitHub repository. The modules we proposed in this paper are all implemented using TensorFlow [1] and Pytorch [29]. The training and inference are both conducted on four A100 GPUs with Intel® Xeon ® Gold 6330 CPU @ 2.00GHz.

Estimating p, p_r and p_{ms}. We use $\mathcal{R}$Oxford 5k [34] as the tuning dataset to estimate the pooling parameters p, p_r and p_{ms} for GeM+, Regional-GeM and Scale-GeM, respectively, and show that the obtained values are sufficiently precise. Firstly, we run inference on the model and store the last feature map for every image. Then, we apply the different types of pooling, varying the pooling parameters, on the feature map for each image. To search for the optimal parameter, we begin by performing a grid search with a step size of 1 and monitor the mAP metric. We terminate the grid search if the mAP in the current iteration is smaller than the previous one. Then, we decrease the grid search step size to 0.1 and redo the previously mentioned steps. Once this procedure is completed, we obtain the values of $p = 4.6$ and $p_r = 2.5$. For Scale-GeM, similar experimentation finds that $p_{ms} \rightarrow \infty$, *i.e.*, max pooling over the multi-scale global features, leads to the best performance. These final obtained parameters are used for experimentation on all evaluation datasets.

SuperGlobal reranking. For reranking evaluations, we follow the same setting as CVNet and rerank the top 400 candidates in most experiments, *i.e.* $M = 400$. Given that our method is drastically more efficient than CVNet, we also study the performance with larger M in specific cases. We pick $K = 9$ for the reranking method and set $\beta = 0.15$ for feature refinement described in Section 4.2.

ReLU adjustment. During our reranking experimentation, following the same way as we explore the impact of p in GeM pooling, we also revisit the ReLU activation [22] by considering a generalized version where the threshold is treated as a parameter denoted by α, which is reduced to vanilla ReLU when $\alpha = 0$. In the best setup, we set threshold α to 0.014 for the first block and the joints between blocks.

5.2. Evaluation Benchmarks

We conduct our experiments on several well-established benchmarks. First, we use Oxford [31] and Paris [33] with revisited annotations [34], referred to as $\mathcal{R}$Oxf and $\mathcal{R}$Par, respectively. There are 4993 (6322) database images in the $\mathcal{R}$Oxf ($\mathcal{R}$Par) dataset, and a different query set for each, both with 70 images. Large-scale results are further reported with the $\mathcal{R}$1M distractor set [34], which contains 1M images. In addition, we also report results on the Google Landmarks dataset v2 (GLDv2) [47], using the latest ground-truth version (2.1). GLDv2-retrieval has 1129 queries (379 validation and 750 testing) and 762k database images.

Method	Multi-scale		Extraction time	Reranking time	Memory (GB)	
	global	local	(ms per image)	(ms on reranking top-400)	$\mathcal{R}$Oxf	$\mathcal{R}$Par
Global features						
RN101-DELG [8]	3	7	65	3.6×10^6 on 100	4.25	5.35
RN101-CVNet [23]	3	1	65	2.4×10^4 on 400	27.02	33.55
RN101-CVNet^Q [23]	3	1	65	2.4×10^4 on 400	6.88	8.52
RN101-SuperGlobal (Ours)	3	-	65	0.37 on 400	0.04	0.05

Table 2: Latency and Memory on the $\mathcal{R}$Oxford and $\mathcal{R}$Paris datasets . Extraction time measures the time needed for the model to produce global features. Reranking time measures the latency of the reranking stage after the global/local features are already computed. Memory usage measures the hardware memory required to store the features.

Method	mAP@100
RN50-DELG retrieval	24.1
+ GV (Rerank top-100)	24.3
RN101-DELG retrieval	26.0
+ GV (Rerank top-100)	26.8
RN50-CVNet retrieval	30.2
+ CVNet reranking (Rerank top-100)	32.4
RN101-CVNet Retrieval	32.5
+ CVNet-reranking (Rerank top-100)	34.9
RN50-SuperGlobal retrieval **[ours]**	31.1
+ SuperGlobal reranking (Rerank top-100)	32.5
+ SuperGlobal reranking (Rerank top-800)	**32.7**
+ SuperGlobal reranking (Rerank top-1600)	32.6
RN101-SuperGlobal retrieval **[ours]**	33.4
+ SuperGlobal reranking (Rerank top-100)	34.6
+ SuperGlobal reranking (Rerank top-800)	34.9
+ SuperGlobal reranking (Rerank top-1600)	**35.0**

Table 3: **GLDv2-retrieval evaluation.** Experimental results (% mAP@100) on GLDv2-retrieval [47]. The best scores are presented in **bold black** and **bold blue** colors for each ResNet backbone.

5.3. Results

We compare different components of SuperGlobal against state-of-the-art models in Table 1. We split the settings into three categories: (1) Global feature retrieval. (2) Global feature retrieval + Local feature reranking. (3) SuperGlobal retrieval and reranking. In addition to the comparisons of performance, to illustrate the efficiency of our method, we compare SuperGlobal against CVNet and DELG in the number of scales, reranking time and the peak memory consumption, and summarize the results in Table 2.

Firstly, as seen from Table. 1, SuperGlobal retrieval significantly outperforms existing models in single-stage retrieval. For instance, in setting (1), our methods (RN101-SuperGlobal without reranking) outperform the second best RN101-DOLG by a significant margin of +7.1% in Revisited Oxford+1M Hard. Under the retrieval then reranking paradigm in setting (2), SuperGlobal retrieval and reranking in setting (3) achieves +3.7% against the second best RN101-CVNet when reranking top 400 in Revisited Oxford+1M Hard. Moreover, SuperGlobal reranking is $64,865\times$ faster and requires $170\times$ less memory, as indicated by Table. 2. Remarkably, our method, even including the reranking time, is almost as efficient as RN101-CVNet-Global with only almost zero overhead.

To evaluate our proposed method when reranking more candidates, we further conduct experiments on GLDv2-retrieval and show the results in Table. 3. First, by increasing the number of images in reranking, SuperGlobal achieves further performance improvements. Considering the significantly reduced latency and memory requirements of our method, SuperGlobal is capable of reranking many more images with the same compute budget. When increasing the reranking budget to top 800 or 1600 candidates, SuperGlobal shows superior performance compared with CVNet reranking (rerank top 100), while still being $16,216\times$ faster and $85\times$ more memory efficient.

5.4. Ablation Study

To evaluate the contribution from each module, we conduct a detailed ablation on $\mathcal{R}$Oxf and $\mathcal{R}$Par, based on the RN101-CVNet pre-trained backbone. We sequentially add the modules one by one to examine whether they lead to a higher performance. Results are presented in Table 4. In summary, GeM+ contributes the most to the performance, while Regional-GeM and Scale-GeM make further improvements. Our finding of modifying ReLU also brings an additional +1% improvement.

5.5. Qualitative Results

Retrieval only. In Figure 5, we show images with different ranks retrieved from SuperGlobal and CVNet, in the absence of reranking. The ranking positions are selected such that SuperGlobal retrieves matching images (highlighted in green boxes) while CVNet doesn't (highlighted in red boxes). We observe that SuperGlobal pays more attention to the fine-grained details of the query image because of the updated pooling techniques proposed in this work.

Reranking. Figure 6 shows top results after SuperGlobal retrieval and reranking. The ranking positions are selected such that the reranked images (highlighted in green boxes) match the query whereas the retrieved images (highlighted in red boxes) do not. These examples show the additional improvement over single-stage SuperGlobal retrieval by ap-

Method	GeM+	Regional-GeM	Scale-GeM	ReLU	Medium		Hard	
					$\mathcal{R}$Oxf	$\mathcal{R}$Par	$\mathcal{R}$Oxf	$\mathcal{R}$Par
Global features								
RN101-CVNet-Global [23]	✘	✘	✘	✘	80.2	90.3	63.1	79.1
RN101-CVNet-Global	✓	✘	✘	✘	84.7	90.8	69.6	81.1
RN101-CVNet-Global	✓	✓	✘	✘	84.8	91.3	70.6	81.9
RN101-CVNet-Global	✓	✓	✓	✘	84.7	91.5	71.1	82.5
RN101-CVNet-Global (SuperGlobal retrieval)	✓	✓	✓	✓	85.3	92.1	72.1	83.5

Table 4: Results (% mAP) on the $\mathcal{R}$Oxford 5k and $\mathcal{R}$Paris 6k datasets, with both Medium and Hard evaluation protocols. Note reranking is not applied in the evaluation.

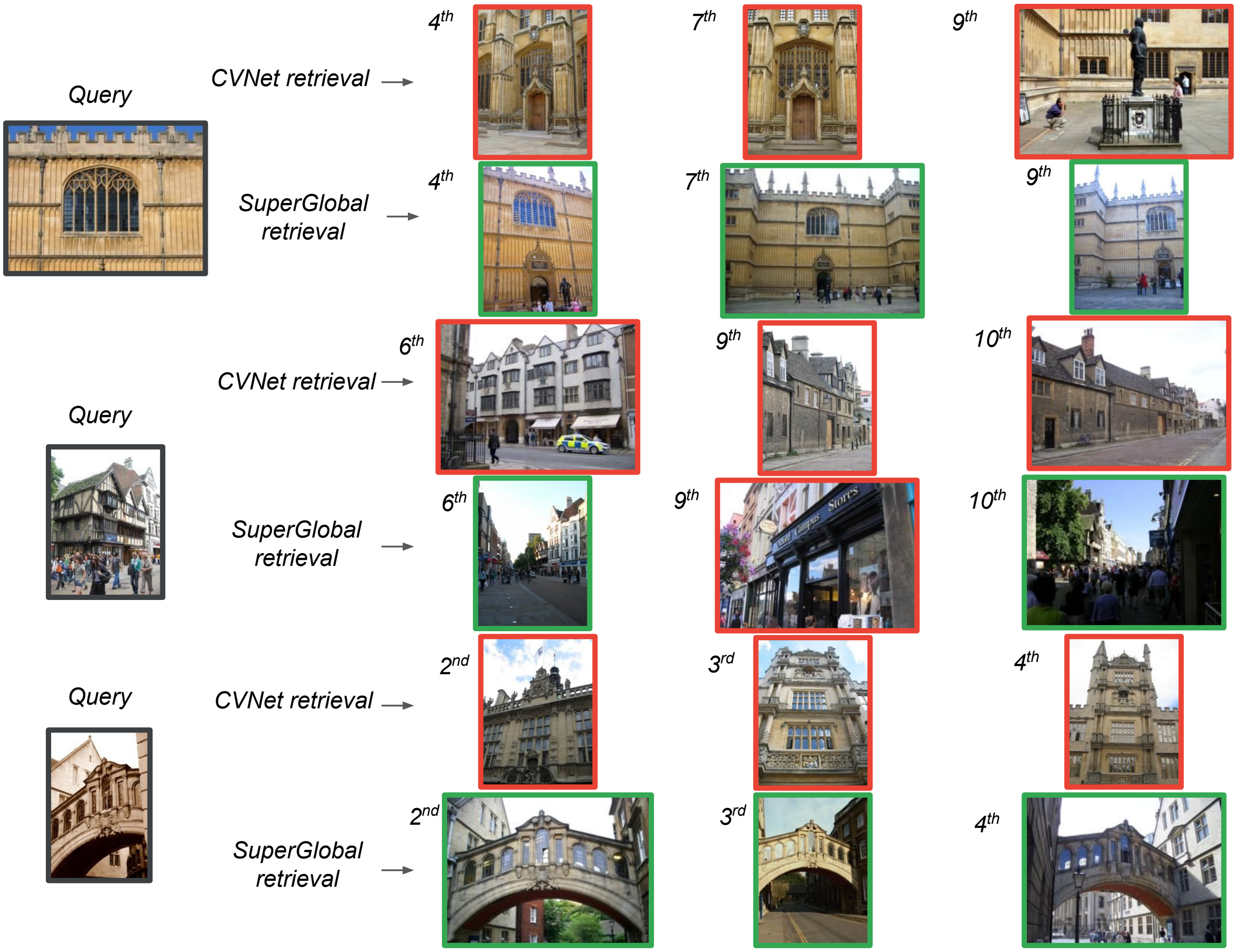

Figure 5: Examples of SuperGlobal retrieval and CVNet retrieval results on $\mathcal{R}$Oxf and $\mathcal{R}$Par dataset.

plying SuperGlobal reranking, demonstrating the techniques in Section 4.2 further refine the order of the top candidates.

5.6. Local vs Global Feature Reranking

SuperGlobal is proved to be significantly more efficient than CVNet reranking. For completeness, we perform experiments to examine whether conducting CVNet reranking on top of the SuperGlobal reranking results can further improve the performance. Table 5 shows that the results are not improved via CVNet reranking, except for a marginal improvement in $\mathcal{R}$Oxford Hard. This indicates that local and global feature reranking somehow overlap in the cases which they are able to improve, and our hypothesis for this is as follows. Global feature reranking combines features from visually similar images with diverse viewpoints and lighting conditions, leading to enhanced representation ca-

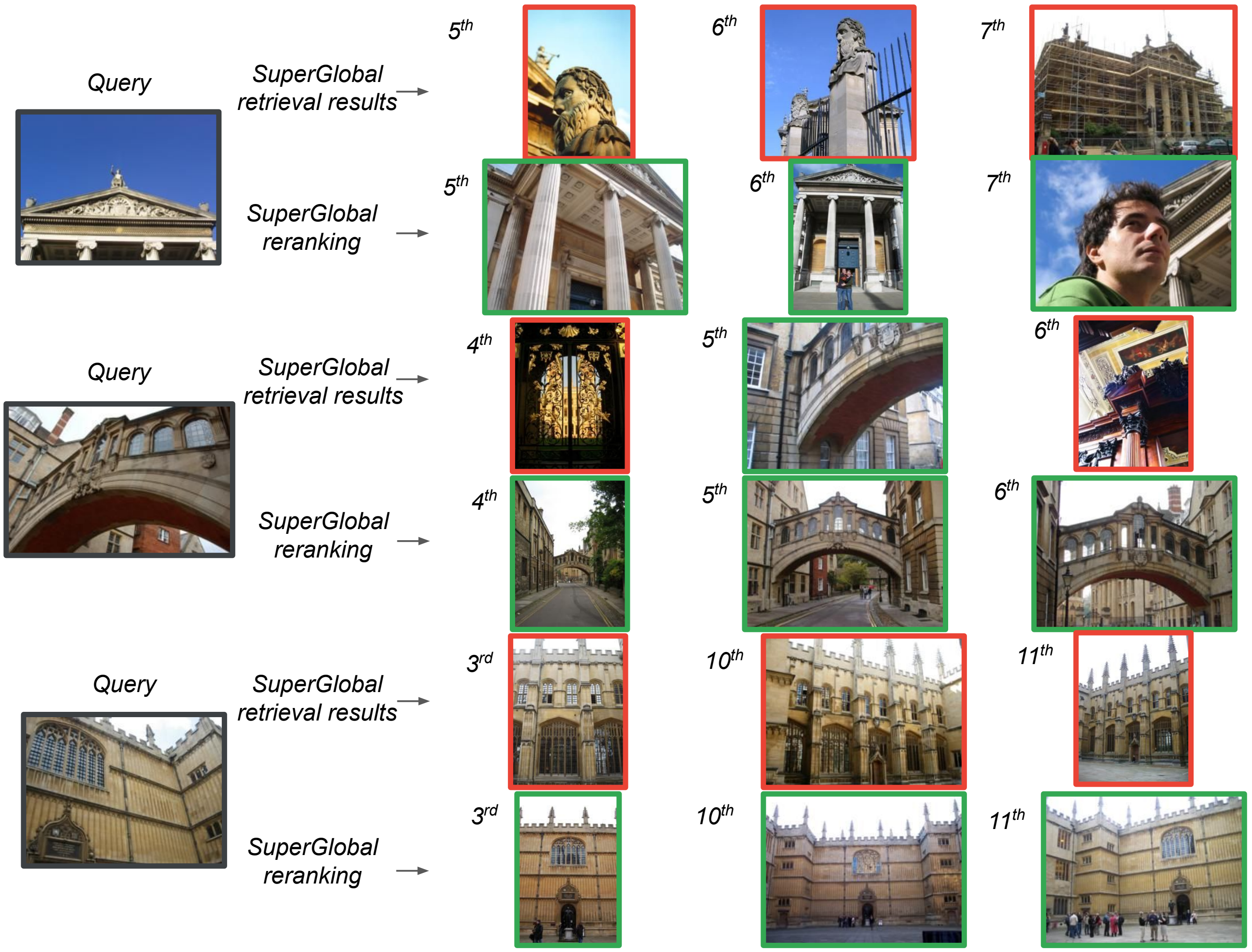

Figure 6: Examples of SuperGlobal retrieval and reranking results on $\mathcal{R}$Oxf and $\mathcal{R}$Par dataset.

Method	CVNet reranking	Medium		Hard	
		$\mathcal{R}$Oxf	$\mathcal{R}$Par	$\mathcal{R}$Oxf	$\mathcal{R}$Par
SuperGlobal	✘	90.9	93.3	80.2	86.7
	✓	90.9	91.9	81.0	79.6

Table 5: Results (% mAP) of conducting CVNet reranking on top of our SuperGlobal reranking results on the $\mathcal{R}$Oxford and $\mathcal{R}$Paris datasets, with both Medium and Hard evaluation protocols.

pability and robustness of the updated features. Therefore, global feature reranking could play a similar role as local feature reranking in retrieval systems and this might result in negligible gains when applying CVNet reranking on top of SuperGlobal.

6. Conclusions

In this paper, we propose a novel image retrieval system, SuperGlobal, which consists of various modules to refine global features for image retrieval and reranking. All of our proposed methods can be plugged into other existing models, and are easy to implement. For global feature refinement, we proposed improved pooling techniques by better training, besides leveraging regional and multi-scale components. In contrast to conventional expensive reranking systems, we devise a strategy that requires only global features, delivering much improved performance while being four orders of magnitude more efficient. This paper marks a first solution to the retrieval and reranking problems relying on a single global image feature. We hope this will spur further research around this direction, to enable continued improvements to the scalabity of these systems.

References

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R.

Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems, 2015. 6

[2] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. NetVLAD: CNN Architecture for Weakly Supervised Place Recognition. In *Proc. CVPR*, 2016. 2

[3] R. Arandjelović and A. Zisserman. Three Things Everyone Should Know to Improve Object Retrieval. In *Proc. CVPR*, 2012. 4

[4] Y. Avrithis and G. Tolias. Hough Pyramid Matching: Speeded-up Geometry Re-ranking for Large Scale Image Retrieval. *IJCV*, 2014. 1, 2

[5] A. Babenko and V. Lempitsky. Aggregating Local Deep Features for Image Retrieval. In *Proc. ICCV*, 2015. 2

[6] A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky. Neural Codes for Image Retrieval. In *Proc. ECCV*, 2014. 2

[7] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-Up Robust Features (SURF). *CVIU*, 2008. 2

[8] B. Cao, A Araujo, and J. Sim. Unifying Deep Local and Global Features for Image Search. In *Proc. ECCV*, 2020. 1, 2, 3, 4, 6, 7

[9] W. Chen, Liu Y, W. Wang, E. M. Bakker, T. Georgiou, P. Fieguth, L. Liu, and M. S. Lew. Deep Learning for Instance Retrieval: A Survey. *TPAMI*, 2022. 2

[10] S. Chopra, R. Hadsell, and Y. LeCun. Learning a Dimilarity Metric Discriminatively, with Application to Face Verification. In *Proc. CVPR*, 2005. 2

[11] J. Deng, J. Guo, and S. Zafeiriou. ArcFace: Additive Angular Margin Loss for Deep Face Recognition. In *Proc. CVPR*, 2019. 2, 3

[12] M. Fischler and R. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM*, 1981. 2

[13] A. Gordo, J. Almazan, J. Revaud, and D. Larlus. End-to-End Learning of Deep Visual Representations for Image Retrieval. *IJCV*, 2017. 2, 4

[14] A. Gordo, F. Radenovic, and T. Berg. Attention-Based Query Expansion Learning. In *Proc. ECCV*, 2020. 4

[15] C. Gulcehre, K. Cho, R. Pascanu, and Y. Bengio. Learned-Norm Pooling for Deep Feedforward and Recurrent Neural Networks. In *Proc. ECML/PKDD*, 2014. 4

[16] K. He, Y. Lu, and S. Sclaroff. Local Descriptors Optimized for Average Precision. In *Proc. CVPR*, 2018. 2

[17] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. In *Proc. CVPR*, 2016. 2

[18] Y. Huang, Y. Wang, Y. Tai, X. Liu, P. Shen, S. Li, J. Li, and F. Huang. Curricularface: Adaptive Curriculum Learning Loss for Deep Face Recognition. In *Proc. CVPR*, 2020. 2, 3

[19] H. Jégou, M. Douze, C. Schmid, and P. Perez. Aggregating Local Descriptors into a Compact Image Representation. In *Proc. CVPR*, 2010. 2

[20] H. Jégou, F. Perronnin, M. Douze, J. Sanchez, P. Perez, and C. Schmid. Aggregating Local Image Descriptors into Compact Codes. *TPAMI*, 2012. 2

[21] Y. Kalantidis, C. Mellina, and S. Osindero. Cross-Dimensional Weighting for Aggregated Deep Convolutional Features. In *Proc. ECCV Workshops*, 2015. 2

[22] A. Krizhevsky, I. Sutskever, and G. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *Proc. NeurIPS*, 2012. 6

[23] S. Lee, H. Seong, S. Lee, and E. Kim. Correlation Verification for Image Retrieval. In *Proc. CVPR*, 2022. 1, 2, 3, 4, 6, 7, 8

[24] D. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *IJCV*, 2004. 2

[25] K. Mikolajczyk and C. Schmid. An Affine Invariant Interest Point Detector. In *Proc. ECCV*, 2002. 2

[26] T. Ng, V. Balntas, Y. Tian, and K. Mikolajczyk. SOLAR: Second-Order Loss and Attention for Image Retrieval. In *Proc. ECCV*, 2020. 1, 2, 3

[27] H. Noh, A. Araujo, J. Sim, T. Weyand, and B. Han. Large-Scale Image Retrieval with Attentive Deep Local Features. In *Proc. ICCV*, 2017. 2

[28] S. Obdrzalek and J. Matas. Sub-linear indexing for large scale object recognition. In *Proc. BMVC*, 2005. 2

[29] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Proc. NeurIPS*, 2019. 6

[30] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object Retrieval with Large Vocabularies and Fast Spatial Matching. In *Proc. CVPR*, 2007. 1, 2

[31] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object Retrieval with Large Vocabularies and Fast Spatial Matching. In *Proc. CVPR*, 2007. 6

[32] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Lost in Quantization: Improving Particular Object Retrieval in Large Scale Image Databases. In *Proc. CVPR*, 2008. 1, 2

[33] James Philbin, Ondřej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008. 6

[34] F. Radenović, A. Iscen, G. Tolias, Y. Avrithis, and O. Chum. Revisiting Oxford and Paris: Large-Scale Image Retrieval Benchmarking. In *Proc. CVPR*, 2018. 2, 6

[35] F. Radenović, G. Tolias, and O. Chum. Fine-Tuning CNN Image Retrieval with No Human Annotation. *TPAMI*, 2018. 1, 2

[36] A. S. Razavian, J. Sullivan, S. Carlsson, and A. Maki. Visual Instance Retrieval with Deep Convolutional Networks. *ITE Transactions on Media Technology and Applications*, 2016. 2

[37] J. Revaud, J. Almazan, R. S. Rezende, and C. R. Souza. Learning With Average Precision: Training Image Retrieval With a Listwise Loss. In *Proc. ICCV*, October 2019. 1, 2

[38] F. Schroff, D. Kalenichenko, and J. Philbin. FaceNet: A Unified Embedding for Face Recognition and Clustering. In *Proc. CVPR*, 2015. 2

[39] J. Sivic and A. Zisserman. Video Google: A Text Retrieval Approach to Object Matching in Videos. In *Proc. ICCV*, 2003. 2

[40] F. Tan, J. Yuan, and V. Ordonez. Instance-level Image Retrieval using Reranking Transformers. In *Proc. ICCV*, 2021. 1, 2

[41] M. Teichmann, A. Araujo, M. Zhu, and J. Sim. Detect-To-Retrieve: Efficient Regional Aggregation for Image Search. In *Proc. CVPR*, 2019. 2

[42] G. Tolias, Y. Avrithis, and H. Jegou. Image Search with Selective Match Kernels: Aggregation Across Single and Multiple Images. *IJCV*, 2015. 2

[43] G. Tolias, T. Jenícek, and O. Chum. Learning and Aggregating Deep Local Descriptors for Instance-Level Recognition. In *Proc. ECCV*, 2020. 2

[44] G. Tolias, R. Sicre, and H. Jégou. Particular Object Retrieval with Integral Max-Pooling of CNN Activations. In *Proc. ICLR*, 2015. 2

[45] H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, and W. Liu. Large Margin Cosine Loss for Deep Face Recognition. In *Proc. CVPR*, 2018. 2

[46] P. Weinzaepfel, T. Lucas, D. Larlus, and Y. Kalantidis. Learning Super-Features for Image Retrieval. In *Proc. ICLR*, 2022. 2

[47] T. Weyand, A. Araujo, B. Cao, and J. Sim. Google Landmarks Dataset v2 – A Large-Scale Benchmark for Instance-Level Recognition and Retrieval. In *Proc. CVPR*, 2020. 3, 6, 7

[48] M. Yang, D. He, M. Fan, B. Shi, X. Xue, F. Li, E. Ding, and J. Huang. DOLG: Single-Stage Image Retrieval with Deep Orthogonal Fusion of Local and Global Features. In *Proc. CVPR*, 2021. 1, 2, 3, 4, 6