Contextual Similarity Distillation for Asymmetric Image Retrieval

Hui Wu[1] Min Wang[2*] Wengang Zhou[1,2*] Houqiang Li[1,2] Qi Tian[3]
[1]CAS Key Laboratory of Technology in GIPAS, EEIS Department, University of Science and Technology of China
[2]Institute of Artificial Intelligence, Hefei Comprehensive National Science Center
[3]Huawei Cloud & AI
wh241300@mail.ustc.edu.cn, wangmin@iai.ustc.edu.cn
{zhwg,lihq}@ustc.edu.cn, tianqi1@huawei.com

Abstract

Asymmetric image retrieval, which typically uses small model for query side and large model for database server, is an effective solution for resource-constrained scenarios. However, existing approaches either fail to achieve feature coherence or make strong assumptions, e.g.*, requiring labeled datasets or classifiers from large model,* etc.*, which limits their practical application. To this end, we propose a flexible contextual similarity distillation framework to enhance the small query model and keep its output feature compatible with that of the large gallery model, which is crucial with asymmetric retrieval. In our approach, we learn the small model with a new contextual similarity consistency constraint without any data label. During the small model learning, it preserves the contextual similarity among each training image and its neighbors with the features extracted by the large model. Note that this simple constraint is consistent with simultaneous first-order feature vector preserving and second-order ranking list preserving. Extensive experiments show that the proposed method outperforms the state-of-the-art methods on the Revisited Oxford and Paris datasets.*

1. Introduction

Most existing image retrieval methods [4, 37, 41, 46, 47, 49] use the same model to map both query and gallery images to feature vectors, which is denoted as *symmetric retrieval* [6, 12]. To achieve high retrieval accuracy, they usually simply select a large model for feature extraction, which suffers inefficiency issue. In some practical scenarios with limited computing and memory resources, such as mobile search, it is hardly affordable to use a large model for feature extraction on the user side, and a lightweight model is more preferable. A naive solution is to directly use lightweight models to extract features for both gallery and queries, which, however, usually degrades the retrieval accuracy due to inferior representation capability of lightweight models. In practice, gallery images can be processed offline with sufficient computing resources while queries undergo feature extraction on the end-user side with limited computing power. In such an *asymmetric retrieval* setting [6, 12], it is feasible to adopt a large model for indexing gallery images and a lightweight one for queries, which

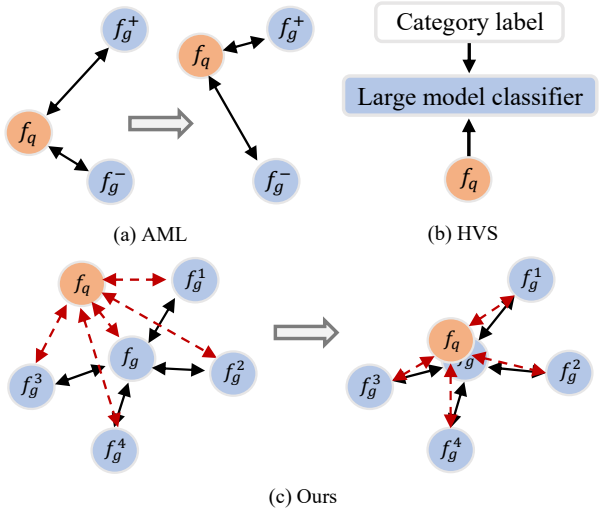


Figure 1. Illustration of existing methods for *asymmetric retrieval* and our contextual similarity distillation framework. $\boldsymbol{f}_q$ (orange) and $\boldsymbol{f}_g$ (blue) denote the embedding vectors from the lightweight query model and the large gallery model, respectively. $\boldsymbol{f}_g^+$: positive sample; $\boldsymbol{f}_g^-$: negative sample; $\boldsymbol{f}_g^i$: The i-th nearest neighbor of $\boldsymbol{f}_g$. Previous methods (a) and (b) require the labels of the dataset for asymmetric distillation, *e.g.*, AML [6] requires triplet labels and HVS [12] requires semantic category labels and the classifer from the large model. Our method (c) is free of supervision from training datasets. During knowledge transfer, it preserves the contextual similarity between training samples and their neighbors.

*Corresponding Author: Min Wang and Wengang Zhou.

makes a trade-off between retrieval accuracy and efficiency.

Lightweight model adaptation is the core problem in *asymmetric retrieval*. Specifically, an optimal lightweight model is supposed to map queries into the same embedding space as the gallery embeddings extracted by a large one. The recent advances [6, 12, 28, 40] generally introduce feature compatibility restrictions into the framework of knowledge distillation and make great progress. In those approaches, they reuse the classifier in the learned large model [12, 28, 40] or use large model to extract features of positive and negative samples for contrastive learning [6], which are shown in Fig. 1 (a) and (b). However, these methods assume the existence of datasets with specific labels when adapting lightweight models or the availability of same training set as the large model, which may be unavailable in real retrieval scenarios. Besides, they only consider the first-order feature preserving restrictions, but ignore the second-order neighbor relationships between images, which have been proven effective for feature learning in [29, 44].

To address above issues, we propose a flexible Contextual Similarity Distillation (CSD) framework to transfer knowledge from large gallery models to lightweight query models while keeping the feature compatibility, as shown in Fig. 1 (c). In our framework, we adopt a new contextual similarity consistency constraint to guide the learning of a lightweight model with a large pretrained fixed model. Specially, for each training image, we first extract its feature using the large fixed model and retrieve its neighbors as anchors in the gallery. The cosine similarities between the training image and its neighboring anchors are used as the contextual similarity to describe the neighbor relationship. Further, we extract the visual feature of the same training image with the lightweight model and compute its contextual similarity vector over the features of its neighboring anchor images, which are extracted by the large model. Finally, we optimize the consistency of the contextual similarity between the large and lightweight models. Remarkably, the whole framework requires no supervision from training datasets during knowledge transfer.

Compared with previous approaches, our framework has two advantages. First, it takes into account contextual consistency constraint for training the lightweight model, which simultaneously optimizes the first-order feature preserving and the second-order neighbor relationship preserving. Second, our framework does not require any supervision from training datasets during knowledge transfer. Therefore, it is possible to train lightweight models using a large amount of unlabeled data, which facilitates the application of our approach in a variety of real-world scenarios.

To evaluate our approach, we conduct comprehensive experiments on the Revisited Oxford and Paris datasets, which are further mixed with one million distractors. Ablation studies demonstrate the effectiveness and generalizability of our framework. Our approach surpasses all state-of-the-art methods by a considerable margin.

2. Related Work

Image Retrieval. Recent years have witnessed a tremendous research progress in content based image retrieval. Prior to deep learning, local feature-based methods [34, 42, 45, 51] have been widely explored. Generally, local features [5, 26] in an image are organized with the bag-of-words model [42] or encoded by aggregation methods, such as ASMK [45], VLAD [20] and Fisher vectors [33], for efficient nearest neighbor search. Further extensions, including spatial verification [34, 51], Hamming embedding [21] and query expansion [9], are also investigated to greatly improve the retrieval accuracy. Nowadays, the most promising retrieval methods are based on fine-tuned convolutional neural networks (CNNs). Many pooling methods, *e.g.*, sum-pooling (SPoC) [3], weighted-sum-pooling (CroW) [24], regional-max-pooling (R-MAC) [47], generalized mean-pooling (GeM) [37] have been explored to aggregate the feature maps of CNNs to form compact global representations. These methods are fine-tuned on a specific dataset with different loss functions [11, 38].

Despite the great progress made by the above methods, the optimal performance usually comes from a large deep model, which is not applicable in some resource-constrained scenarios. We focus on *asymmetric retrieval*, where usually the query (user) side takes a lightweight model while the gallery side applies a large model.

Feature Compatible Learning. BCT [40] first formulates the problem of backward-compatible learning and reuses the classifier of the large model so that feature compatibility is achieved. AML [6] introduces an asymmetric metric learning framework to achieve feature compatibility, which, however, fails to guarantee that the accuracy of *asymmetric retrieval* exceeds that of *symmetric retrieval* when using lightweight models. HVS [12] further considers both model weights and model structures to achieve feature compatibility. *Feature translation* [18] studies interoperability between different retrieval systems. It uses large models for both side, which is unaffordable in a practical scenario.

Differently, we propose a flexible contextual similarity distillation framework, which is free of supervision from training datasets during knowledge transfer. Moreover, optimizing the contextual similarity restrictions allows to focus on both first-order feature compatibility and second-order ranking list preservation, which is directly relevant with the retrieval performance in *asymmetric retrieval*.

Lightweight Network. Large models [14, 25] are superior in performance but usually consumes more resources in computing and memory. Typically, model compression [2, 15] reduces model size by trading accuracy for efficiency. Besides, hand-craft efficient mobile-size ConvNets,

such as SqueezeNets [19], MobileNets [17, 39] and ShuffleNets [27, 50], show great advantages. Recently, neural architecture search becomes increasingly popular for designing efficient mobile-size ConvNets [43]. They achieve higher efficiency than hand-craft mobile ConvNets by extensively tuning the network width, depth convolution kernel type, and size. In this paper, we focus on *asymmetric retrieval* in resource-constrained scenarios, which typically uses lightweight models to extract features for queries. We adopt the mobile ConvNets mentioned above in this work.

Knowledge Transfer. Knowledge transfer is the technique of transferring knowledge from a source model to a target model. A pioneering work by Hinton *et al*. [16] achieves this goal by encouraging the target model to mimic the predicted class logits of the source model. More recently, knowledge transfer has been introduced to metric learning. Some works propose to extract and transfer the rank of similarities between samples [8] and probability distributions of their similarities [1, 13, 32] in the source embedding space. In [31], geometric relationships between samples, such as distances and angles, are used as knowledge to consider the details of the source embedding space. However, none of these approaches achieve feature compatibility between source-domain and target-domain. As a result, they cannot be directly used for *asymmetric retrieval*.

In contrast, our approach preserves the contextual similarity of different samples over their neighboring points, whose features are extracted by the source domain model. This allows feature compatibility between different domains while transferring knowledge.

Contextual Similarity. In image retrieval, contextual similarity has proved to be more effective in distinguishing semantic relevance between images compared to direct feature comparison. CDM [23] iteratively regularizes the average distance of each point to its neighbors to update the similarity matrix. In [30], a lightweight CNN network is trained to explore the contextual information and recalculate the similarities between images. Differently, our approach uses the contextual information captured by the large model to guide the learning of the lightweight model. Notably, the features of neighboring images are always extracted by the large model. Such design allows the lightweight model to consider the second-order contextual information between images while preserving feature compatibility.

3. Method

3.1. Problem Formulation

Let $\phi(\cdot)$ denote a feature extractor, trained on a training set $\mathcal{T}$. $\phi(\cdot)$ is used to map the image x in a gallery $\mathcal{G}$ into an L_2-normalized feature vector $\boldsymbol{f}_g = \phi(x) \in \mathcal{R}^d$, and we denote the model used for gallery indexing as $\phi_g(\cdot)$. During testing, the query model $\phi_q(\cdot)$ maps an image $q \in \mathcal{Q}$ into an L_2-normalized feature vector $\boldsymbol{f}_q = \phi_q(q) \in \mathcal{R}^d$. The cosine similarity between $\boldsymbol{f}_g$ and $\boldsymbol{f}_q$ is used to calculate the similarity between images. The performance of a retrieval system conditioned on $\mathcal{Q}$ and $\mathcal{G}$ is measured by some metrics, such as mean Average Precision (mAP), which we denote as $P(\phi_q(\cdot), \phi_g(\cdot)|\mathcal{Q}, \mathcal{G})$. Specifically, it is calculated by processing query set $\mathcal{Q}$ with $\phi_q(\cdot)$ and indexing gallery $\mathcal{G}$ with $\phi_g(\cdot)$. For convenience, we ignore query and gallery sets and denote it as $P(\phi_q(\cdot), \phi_g(\cdot))$.

Assume $\phi_q(\cdot)$ and $\phi_g(\cdot)$ are different models and $\phi_q(\cdot)$ is significantly smaller than $\phi_g(\cdot)$ in parameter scale. *Symmetric retrieval* adopts either $\phi_q(\cdot)$ or $\phi_g(\cdot)$ to process both query and gallery sets, while *asymmetric retrieval* uses $\phi_q(\cdot)$ to embed query images and $\phi_g(\cdot)$ to process the gallery. A key requirement [12] for *asymmetric retrieval* is that query and gallery models should be compatible, *i.e*., the feature embedding of query model locates in the same or very similar manifold space with that of the gallery model. Generally, it is expected that $P(\phi_q(\cdot), \phi_g(\cdot)) > P(\phi_q(\cdot), \phi_q(\cdot))$ and $P(\phi_q(\cdot), \phi_g(\cdot)) \approx P(\phi_g(\cdot), \phi_g(\cdot))$, which allows *asymmetric retrieval* to strike a balance between performance and efficiency.

3.2. Contextual Similarity Distillation Framework

In this work, we explore a new contextual similarity constraint to learn lightweight query model $\phi_q(\cdot)$ for *asymmetric retrieval*. During the learning of $\phi_q(\cdot)$, it preserves the contextual similarity between each training image and its neighbors with features extracted by gallery model $\phi_g(\cdot)$. An overview of our framework is shown in Fig. 2.

During the training of the lightweight query model, the gallery model $\phi_g(\cdot)$ pretrained on the training set $\mathcal{T}_g$ is frozen. With a separate gallery $\mathcal{G}_t$ to mine neighbor images, we first extract the features $\boldsymbol{F} = \left[\boldsymbol{f}_g^1, \boldsymbol{f}_g^2, \cdots, \boldsymbol{f}_g^N\right] \in \mathcal{R}^{d\times N}$ of images in $\mathcal{G}_t$:

$$\boldsymbol{f}_g^i = \phi_g(g_i) \in \mathcal{R}^d, \text{ for } i = 1, 2, \cdots, N, \tag{1}$$

where g_i is the i-th image in the gallery. Then, for each training sample $x \in \mathcal{T}_q$, we embed it with both gallery model $\phi_g(\cdot)$ and query model $\phi_q(\cdot)$ to get $\boldsymbol{g}$ and $\boldsymbol{q}$:

$$\boldsymbol{g} = \phi_g(x) \in \mathcal{R}^d, \ \boldsymbol{q} = \phi_q(x) \in \mathcal{R}^d. \tag{2}$$

$\boldsymbol{g}$ is treated as a query and we obtain a ranking list of top-K images $R = [r_1, r_2, \cdots, r_K]$ as anchors from the gallery by a retrieval algorithm, where r_i denotes the ID of the i-th image. We assume the query image is not contained in the gallery, and insert it directly to the front of the ranking list. Thus, the features of the anchor images in the ranking list are described as $\boldsymbol{F}_K = \left[\boldsymbol{g}, \boldsymbol{f}_g^{r_1}, \cdots, \boldsymbol{f}_g^{r_K}\right] \in \mathcal{R}^{d\times(K+1)}$.

Since the gallery model has been well trained, the retrieval results adequately reflect the neighbor structure of x in the gallery embedding space. We further represent this

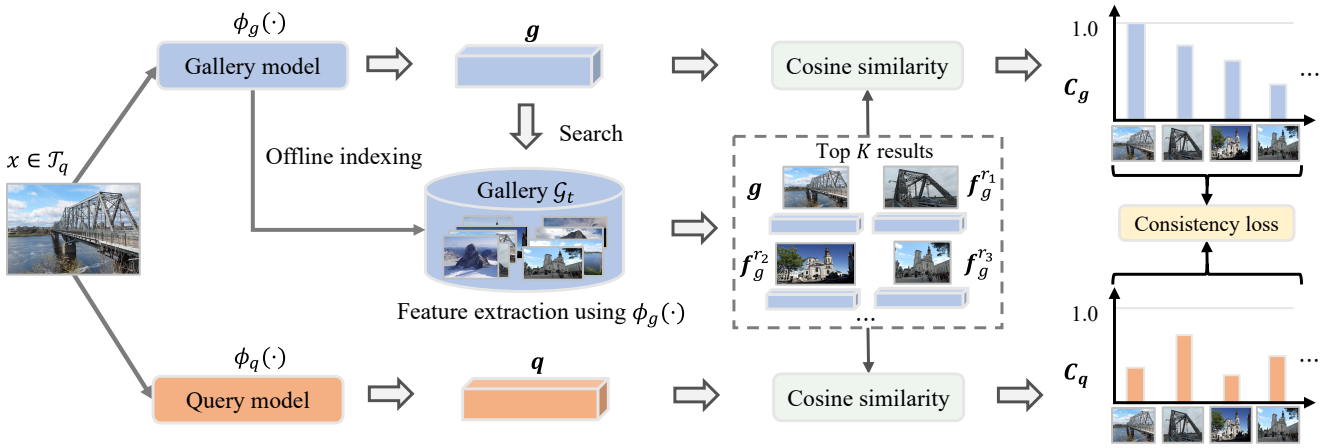


Figure 2. An overview of our framework. Given an image $\boldsymbol{x} \in \mathcal{T}_q$, the gallery model and the query model map it into feature vectors $\boldsymbol{g}$ and $\boldsymbol{q}$, respectively. Then, $\boldsymbol{g}$ is used to retrieve its top K neighbors in the gallery $\mathcal{G}_t$, which is indexed by $\phi_g(\cdot)$. Further, $\boldsymbol{C}_g$ and $\boldsymbol{C}_q$ are calculated to measure the contextual similarity between image $\boldsymbol{x}$ and its neighbors. Finally, by constraining the consistency between $\boldsymbol{C}_g$ and $\boldsymbol{C}_q$, the query model preserves the contextual similarity between images and their neighbors under the *asymmetric retrieval* setting.

structure as contextual similarity. Specifically, we compute the cosine similarity between the query $\boldsymbol{g}$ and features $\boldsymbol{F}_K$ of the ranking list as the contextual similarity. Formally,

$$\boldsymbol{C}_g = \left[\boldsymbol{g}^T\boldsymbol{g}, \boldsymbol{g}^T\boldsymbol{f}_g^{r_1}, \cdots, \boldsymbol{g}^T\boldsymbol{f}_g^{r_K}\right] \in \mathcal{R}^{K+1}. \tag{3}$$

For the feature $\boldsymbol{q}$ extracted by the query model $\phi_q(\cdot)$, we obtain the corresponding contextual similarity:

$$\boldsymbol{C}_q = \left[\boldsymbol{q}^T\boldsymbol{g}, \boldsymbol{q}^T\boldsymbol{f}_g^{r_1}, \cdots, \boldsymbol{q}^T\boldsymbol{f}_g^{r_K}\right] \in \mathcal{R}^{K+1}. \tag{4}$$

After that, we impose consistency constraints $\mathcal{L}_c$ on the contextual similarities $\boldsymbol{C}_g$ and $\boldsymbol{C}_q$ to optimize the $\phi_q(\cdot)$ so that embedding $\boldsymbol{q}$ has the same neighboring context as $\boldsymbol{g}$ with the neighbor images in the embedding space of gallery. Notably, by computing both contextual similarities of $\boldsymbol{g}$ and $\boldsymbol{q}$ in gallery embedding space, we transfer the neighbor structure of the gallery embedding space to the query embedding space and keep them compatible with each other.

Caching the gallery $\mathcal{G}_t$. Since we use a very deep model (*e.g.*, ResNet101) as gallery model $\phi_g(\cdot)$, it will require very large computational and storage resources if we use it online to compute the feature embeddings of gallery images $\mathcal{G}_t$ and training images $\mathcal{T}_g$. Fortunately, our framework does not need to optimize the gallery model. Hence, we extract the features of all the images in both gallery and training dataset before training. During training, the features are cached in the memory. For each training sample, we find its neighbors and load the corresponding features.

3.3. Contextual Similarity Consistency Constraints

For *asymmetric retrieval*, the query model $\phi_q(\cdot)$ requires the capabilities of feature compatibility and neighbor structure preserving. To this end, the optimal $\phi_q^*(\cdot)$ is learned by minimizing the contextual similarity consistency constraint over training set $\mathcal{T}_q$. In this work, we consider two types of consistency loss, *i.e.*, regression loss and KL divergence loss, which are discussed in the following.

$\mathcal{L}_1$ and $\mathcal{L}_2$ distances. A naive option is to encourage both models to have close contextual similarity for the same input example. To measure the closeness between vectors, $\mathcal{L}_1$ and $\mathcal{L}_2$ distance metrics are two most popular, with which we define the regress loss as follows,

$$\mathcal{L}_D = \left(\sum_{i=1}^{K+1} \left|\boldsymbol{C}_q^i - \boldsymbol{C}_g^i\right|^\alpha\right)^{\frac{1}{\alpha}}, \ \alpha = 1, 2. \tag{5}$$

Essentially, Eq. (5) is equivalent to the following equation:

$$\mathcal{L}_D^\alpha = \underbrace{\left|\boldsymbol{q}^T\boldsymbol{g} - 1\right|^\alpha}_{\text{first-order}} + \underbrace{\sum_{i=1}^{K} \left|\boldsymbol{q}^T\boldsymbol{f}_g^{r_i} - \boldsymbol{g}^T\boldsymbol{f}_g^{r_i}\right|^\alpha}_{\text{second-order}}. \tag{6}$$

Optimizing the above constraint is consistent with optimizing both the first-order feature compatibility and the second-order ranking list preserving losses.

KL Divergence. Another alternative loss to optimize the above contextual similarity consistency is based on KL divergence. To this end, we first convert the contextual similarity into the form of probability distribution over neighboring anchors:

$$p_j^i = \frac{\exp\left(\boldsymbol{C}_j^i/\tau_j\right)}{\sum_{l=1}^{K+1}\exp\left(\boldsymbol{C}_j^l/\tau_j\right)}, \text{ for } i = 1, 2, \cdots, K+1, \tag{7}$$

where τ_j is the temperature coefficient and $j \in \{q, g\}$. Since the ranking list may contain images far away from a training image in the embedded feature space, temperature

coefficient τ_g is set less than 1 to keep $\phi_q(\cdot)$ focus mainly on the near-neighbor structure of the training images, rather than on distant points. Then, the consistency constraint can be defined as the KL divergence between two probabilities over the same set of neighbors:

$$\mathcal{L}_{KL} = D_{KL}(p_g||p_q) = \sum_{l=1}^{K+1} p_g^l \log \frac{p_g^l}{p_q^l}. \tag{8}$$

Similar to Eq. (6), we also split the KL divergence loss into the first-order and the second-order terms. Let $D_q = \sum_{l=1}^{K+1} \exp(C_q^l/\tau_q)$ and Eq. (8) is rewritten as:

$$\begin{aligned}\mathcal{L}_{KL} &= \underbrace{\sum_{l=1}^{K+1} p_g^l \log p_g^l}_{\text{constant } C} - \sum_{l=1}^{K+1} p_g^l \log p_q^l \\ &= C - \underbrace{\frac{p_g^1}{\tau_q} \boldsymbol{q}^T \boldsymbol{g}}_{\text{first-order}} + \underbrace{p_g^1 \log D_q - \sum_{l=2}^{K+1} p_g^l \log p_q^l}_{\text{second-order}}.\end{aligned} \tag{9}$$

Thus, the effect of optimizing both first-order and second-order losses is also achieved.

4. Experiments

4.1. Experimental Setup

Training Dataset. (1) *SfM120k* [37]. It is obtained by 3D reconstruction of large-scale unlabeled image collections. We follow the setting of AML [6] to use 551 3D models for training and the remaining 162 3D models for validation. (2) GLDv2 [48]. It is first collected by Google for the landmark retrieval competition, which consists of 1,580,470 images from 81,313 categories. We randomly divide it into two subsets, *i.e.,* 'train' and 'val', with 80% and 20% of the images, respectively. The 'train' split is used for learning while the 'val' split is used for validation. We use training set as gallery $\mathcal{G}_t$ in all cases unless otherwise stated.

Networks. R101-DELG [7] and R101-GeM [37] are taken as gallery models with feature dimensionality d of 2048. Lightweight networks including ShuffleNets [27, 50], MobileNets [17,39] and EfficientNets [43] are adopted as query models. GeM [37] pooling is used for all models. Tab. 1 gives the number of parameters and the computational complexity (in FLOPS) of different networks used in this work. Both models are adapted to image retrieval by removing all the fully connected layers. Query models are further equipped with an additional fully connected layer to match the output dimensionality of the gallery model.

Evaluation Datasets and Metrics. Revisited versions of Oxford5k [34] and Paris6k [35] are used for evaluation, which are denoted as $\mathcal{R}$Oxf and $\mathcal{R}$Par [36]. Both datasets contain 70 queries, with 4,993 and 6,322 gallery images, respectively. We report Mean Average Precision (mAP) on the Medium and Hard setups. Large-scale results are further reported with the $\mathcal{R}$1M [36] distractor images.

Query Model	Gallery Model	GFLOPS		Param(M)	
		ABS	%	ABS	%
ResNet101	ResNet101	42.85	100.0	42.50	100.0
MobileNetV2	ResNet101	2.50	5.83	4.85	11.4
EfficientNetB3		6.26	14.61	13.84	32.56

Table 1. FLOPS and parameters for models used in this work, absolute (ABS) and relative (%) to gallery model. All the models are adapted for image retrieval with fully connected layers removed. Please refer to supplementary materials for more results.

Implementation Details. When using *SfM120k* for training, we follow the setting of AML [6] for fair comparison. We train query model for 10 epochs on one NVIDIA RTX 3090 GPU with a batch size of 64. When using GLDv2, we extract a 512×512-pixel crop from the randomly resized image and perform random color jittering. Batch size is set as 256 and we train our model on 4 NVIDIA RTX 3090 GPUs for 5 epochs. all models are optimized using Adam with an initial learning rate of 10^{-3} and a weight decay of 10^{-6}. A linearly decaying scheduler is adopted to gradually decay the learning rate to 0 when the desired number of steps is reached. Length K of ranking list is set to 4096. τ_g and τ_q are set to 0.01 and 1.0. For GeM [37] pooling, we fix parameter p as 3.0. We train each model five times and report the mean and standard deviation.

During testing, images are resized with the larger dimension equal to 1024 pixels, preserving the aspect ratio. We extract image features at three scales, *i.e.*, $\{1/\sqrt{2}, 1, \sqrt{2}\}$, and perform L_2 normalization for each scale independently. Then, the features are averaged across different scales, followed by another L_2 normalization. Under the *asymmetric retrieval* setting, we extract the features of queries using a lightweight query model $\phi_q(\cdot)$, and those of the gallery images with a large model $\phi_g(\cdot)$.

4.2. Ablation Study

Length of Ranking List. Fig. 3 shows the mAP of our method with different lengths of ranking list R. As the length increases, the performance increases under all settings but saturates when the list length $K \geq 1024$. When K is small, these images are not enough to cover the neighbors of queries. On the contrary, when K is large, it contains a large number of samples far from queries. These images do not provide useful information for describing the neighbor structure of queries, and thus the performance saturates.

Anchor Image Selection. Our approach uses a gallery model to mine the neighbors of each training image, which are further utilized as anchor images for computing the contextual similarity. In this experiment, we test two other vari-

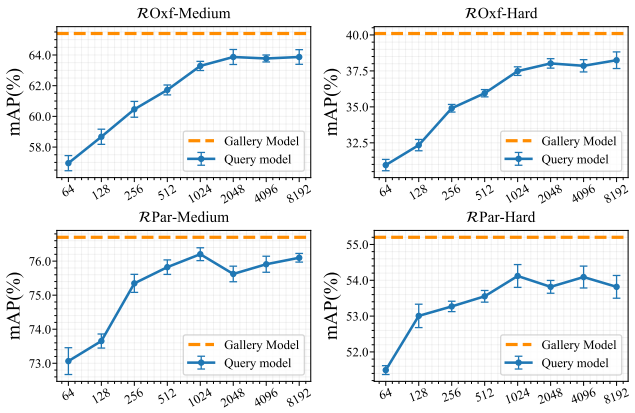


Figure 3. mAP (*asymmetric retrieval*) comparison of different K. The gallery model R101-GeM [37] is used to index the database. Queries are processed by the gallery model or query model MobileNetV2 [39], respectively.

ants to choose those anchor images. (1) *Random*: For each training image, a certain number of images are randomly selected from gallery as anchor images. (2) *Fixed*: Gallery images are clustered into several clusters and centroid vectors are used as anchor features for each training image.

As shown in Fig. 4, both *Random* and *Fixed* variants cause severe performance degradation, which indicates that preserving contextual similarity on near neighbors is beneficial for *asymmetric retrieval*. Randomly selected images may contain many samples far away from a specific training sample, which makes contextual similarity uninformative. Similarly, with a small number of clusters, the granularity for space partitioning is too coarse to capture the neighbor structure of any training sample. The performance of both variants gradually increases as the number of anchor images increases. Thus, a large number of samples are needed to cover the neighbors of any training image.

Impact of Loss Type. As shown in Tab. 2, $\mathcal{L}_2$ loss and $\mathcal{L}_{KL}$ loss both lead to good performance, while the $\mathcal{L}_1$ loss performs the worst. This is due to the fact that the $\mathcal{L}_1$ loss uses absolute values as distances, which leads to difficulty in optimization. We take KL divergence as our default consistency constraint. In Tab. 3, we further verify that optimizing our contextual similarity consistency constraint is consistent with optimizing both first-order feature preserving and second-order ranking list preserving losses. $\mathcal{L}_c^-$ denotes that we omit the first-order term in Eq. (9) when calculating the consistency constraint. The worst result is achieved using only the first-order feature regression loss $\mathcal{L}_r$. Better performance is achieved when $\mathcal{L}_c^-$ is used, which shows that second-order ranking list preserving is more important for *asymmetry retrieval* relative to feature regression. From the 2nd and 3rd row, the performance is further enhanced when we adopt both $\mathcal{L}_r$ and $\mathcal{L}_c^-$. Directly using $\mathcal{L}_c$ achieves the best performance, which demonstrates that our contextual similarity consistency includes both first-order feature preserving and second-order ranking list preserving losses.

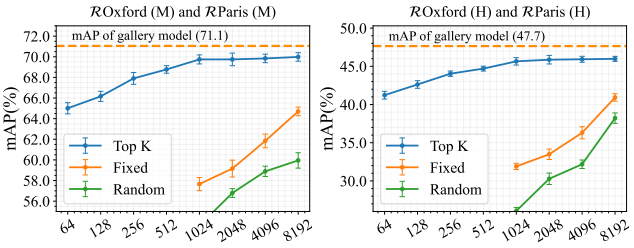


Figure 4. mAP (*asymmetric retrieval*) comparison of different methods to select anchor images with different K. mAP are average over two difficulty setups, Medium (left) and Hard (right). R101-GeM [37] and MobileNetV2 [39] are used as gallery and query model, respectively.

Loss Type	Medium		Hard	
	$\mathcal{R}$Oxf	$\mathcal{R}$Par	$\mathcal{R}$Oxf	$\mathcal{R}$Par
$\mathcal{L}_1$	60.7	70.8	35.6	49.2
$\mathcal{L}_2$	64.0	75.7	**38.6**	53.6
$\mathcal{L}_{KL}$	**64.1**	**76.1**	37.5	**54.2**

Table 2. mAP (*asymmetric retrieval*) comparison of loss type, with $\tau_q = 1.0$ and $\tau_g = 0.01$. R101-GeM [37] and MobileNetV2 [39] are used as gallery and query model, respectively.

$\mathcal{L}_r$	$\mathcal{L}_{KL}^-$	$\mathcal{L}_{KL}$	Medium		Hard	
			$\mathcal{R}$Oxf	$\mathcal{R}$Par	$\mathcal{R}$Oxf	$\mathcal{R}$Par
✓			50.3	63.7	28.9	39.8
	✓		60.4	73.7	35.2	50.8
✓	✓		62.5	75.1	**37.6**	52.1
		✓	**64.1**	**76.1**	37.5	**54.2**

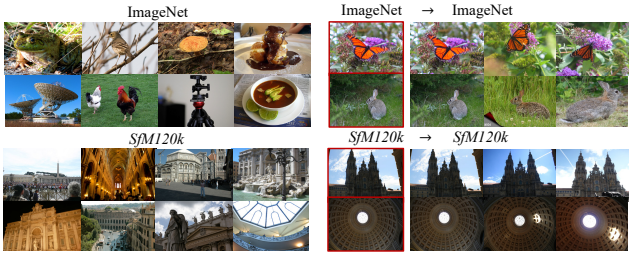
Table 3. Ablation experiments about the first-order and second-order terms in our consistency loss. $\mathcal{L}_r$: $\mathcal{L}_2$ distance between visual features output by the gallery and query models, which is found best by AML [6] for the asymmetric setting; $\mathcal{L}_{KL}$: default constraint Eq. (8); $\mathcal{L}_{KL}^-$: First-order term in Eq. (9) is ignored when calculating $\mathcal{L}_{KL}$. R101-GeM [37] and MobileNetV2 [39] are used as gallery and query model, respectively.

Flexibility and Scalability. In Tab. 4, we further show the scalability of our framework. We first take *SfM120k* as the training set and randomly sample 10% data from GLDv2 dataset to join it. This brings us 0.4%, 2.1% mAP boosts on $\mathcal{R}$Oxf and 0.1%, 0.3% boosts on $\mathcal{R}$Par datasets, when using R101-GeM as the gallery model. For R101-DELG as the gallery model, the performance improvement is even more remarkable. Next, we use R101-DELG as the gallery model and sample different numbers of images from GLDv2 for training. The performance gradually increases as the number of training data increases. Without labels, our framework improves the performance of the query model with the large amount of data present, which shows its flexibility and scalability.

Sensitivity to Gallery $\mathcal{G}_t$. During training, we take training images as queries and retrieve their neighbors in the gallery

Gallery Net $\phi_g(\cdot)$	Training Set $\mathcal{T}_q$	Image Numbers	Medium		Hard	
			$\mathcal{R}$Oxf	$\mathcal{R}$Par	$\mathcal{R}$Oxf	$\mathcal{R}$Par
R101-GeM	*SfM120k*	91,642	64.1	76.1	37.5	54.2
R101-GeM	*SfM120k* + GLDv2 (×0.1)	219,720	**64.5**	**76.2**	**39.6**	**54.5**
R101-DELG	*SfM120k*	91,642	72.7	83.2	53.8	67.9
R101-DELG	*SfM120k* + GLDv2 (×0.1)	219,720	**73.6**	**83.4**	**55.9**	**70.8**
R101-DELG	GLDv2 (×0.1)	128,078	70.6	81.6	52.4	64.5
R101-DELG	GLDv2 (×0.2)	256,156	73.2	83.7	54.6	70.3
R101-DELG	GLDv2 (×0.3)	384,234	74.4	84.9	56.5	71.6
R101-DELG	GLDv2 (×0.4)	512,312	**75.2**	**86.6**	**57.5**	**72.5**

Table 4. mAP (*asymmetric retrieval*) comparison of different dataset size. ($\times x$) denotes the small dataset formed by randomly selecting x proportion of images from the full GLDv2 dataset. MobileNetV2 [39] is used as query model.



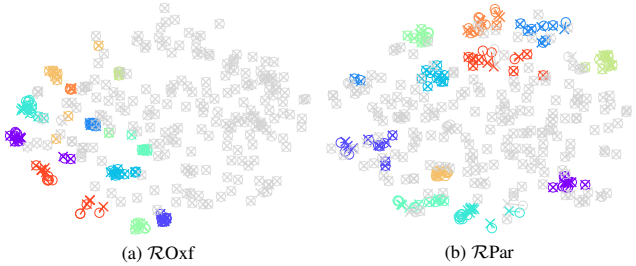
(a) Sample images from different datasets (b) Search results on different galleries

Figure 5. Example images and retrieval results for different datasets. (a) The data distributions of ImageNet and *SfM120k* are dramatically different. (b) Using training set as gallery, R101-GeM [37] achieves promising retrieval results. Query images are on the left(red outline).

Training Set $\mathcal{T}_q$	Gallery Set $\mathcal{G}_t$	Medium		Hard	
		$\mathcal{R}$Oxf	$\mathcal{R}$Par	$\mathcal{R}$Oxf	$\mathcal{R}$Par
ImageNet (×0.1)	*SfM120k*	52.4	58.9	27.8	34.7
SfM120k	ImageNet (×0.1)	57.3	69.8	31.3	46.7
ImageNet (×0.1)	ImageNet (×0.1)	61.3	75.2	35.9	51.7
SfM120k	*SfM120k*	**64.1**	**76.1**	**37.5**	**54.2**

Table 5. mAP (*asymmetric retrieval*) comparison of different training datasets and galleries during training. R101-GeM [37] and MobileNetV2 [39] are used as gallery and query model.

$\mathcal{G}_t$. Intuitively, if the distribution of training images and gallery are disparately different, the anchors in the ranking list may not reflect the neighbors of training images, which will degrade the learning of query model. We use *SfM120k* and 10% random samples from ImageNet [10] to verify our intuition. Fig. 5 (a) shows some examples from these datasets. Tab. 5 shows that the performance declines when the distributions of training data and gallery vary dramatically. Interestingly, it still works well when ImageNet is adopted as both training set and gallery. As shown in Fig. 5 (b), when the training set and the gallery share the same data distribution, the gallery model return true neighbors of training images, which allows query model to focus on the near-neighbor structure of each training image.



(a) $\mathcal{R}$Oxf (b) $\mathcal{R}$Par

Figure 6. T-SNE embeddings of $\mathcal{R}$Oxf and $\mathcal{R}$Par datasets. Different colors represent different buildings with gray representing distractors. We randomly select 10 for each building category and 100 samples from all distractors. Gallery model: circles; Query model: $\times$. A line connects two representations of each example.

Qualitative Results. Fig. 6 shows the embeddings of some $\mathcal{R}$Oxf and $\mathcal{R}$Par images, each processed by a gallery and a query model. For *asymmetric retrieval*, it is critical to keep feature compatibility. During training, the query model is constrained to preserve the contextual similarity between each training image and its neighbors in the embedding space of the gallery model. This keeps the output space of the query and gallery models compatible.

4.3. Comparison with State-of-the-art Methods

mAP Comparison. We conduct extensive comparisons of our method with state-of-the-art methods on the full benchmark. As shown in Tab. 6, our framework achieves the best performance under the *asymmetric* setting. When using R101-DELG as gallery model and GLDv2 as training set, the EfficientNetB3 trained with our framework outperforms best previous method in mAP by 1.03%, 2.84% on $\mathcal{R}$Oxf and 0.87%, 1.15% on $\mathcal{R}$Par datasets with Medium and Hard protocols, respectively. For $\mathcal{R}$1M, we also achieve the best performance, outperforming HVS [12] in mAP by 0.49% on $\mathcal{R}$Oxf-Medium, 1.83% on $\mathcal{R}$Par-Medium, 1.34% on $\mathcal{R}$Oxf-Hard and 1.00% on $\mathcal{R}$Par-Hard. These results well demonstrate the superiority of our framework.

Training Efficiency Comparison. Since our framework requires retrieval during training, it may affect the training storage overhead and time efficiency when the size of training gallery $\mathcal{G}_t$ is large. In our implementation, the feature dimension is 2048. We compress the memory requirements with PQ [22]. Specifically, we divide the features into 256 segments and quantize each segment into 8 bits. For GLDv2, it takes 0.3 GB space to store the gallery index in the memory and the online retrieval latency is 0.105 s, which is negligible compared to the network training time. As for Contr* [6], it performs hard sample mining before each epoch, which requires a complex and time-consuming re-extraction of image features. It takes Contr* [6] about 7

METHOD	QUERY NET $\phi_q(\cdot)$	GALLERY NET $\phi_g(\cdot)$	MEDIUM				HARD			
			$\mathcal{R}$Oxf	$\mathcal{R}$Oxf+$\mathcal{R}$1M	$\mathcal{R}$Par	$\mathcal{R}$Par+$\mathcal{R}$1M	$\mathcal{R}$Oxf	$\mathcal{R}$Oxf+$\mathcal{R}$1M	$\mathcal{R}$Par	$\mathcal{R}$Par+$\mathcal{R}$1M
(A) Training without gallery model										
GeM† [37] (*SfM120k*)	MobileNetV2	MobileNetV2	58.81	40.02	67.87	42.25	33.41	17.71	40.97	16.59
GeM† [37] (*SfM120k*)	EfficientNetB3	EfficientNetB3	54.22	37.10	71.21	44.67	27.53	17.49	48.00	18.45
GeM† [37] (*SfM120k*)	R101	R101	**65.43**	**45.23**	**76.75**	**52.34**	**40.13**	**19.92**	**55.24**	**24.77**
DELG† [7] (GLDv2)	MobileNetV2	MobileNetV2	62.42	42.21	77.91	55.09	36.56	18.64	57.96	28.81
DELG† [7] (GLDv2)	EfficientNetB3	EfficientNetB3	66.64	49.67	81.78	61.10	43.82	24.89	63.90	32.34
DELG† [7] (GLDv2)	R101	R101	**78.55**	**66.02**	**88.58**	**73.65**	**60.89**	**41.75**	**76.05**	**51.46**
(B) Training with R101-GeM as gallery model and SfM120k as training dataset										
RKD† [31]			1.36	0.01	4.06	0.03	0.70	0.01	2.51	0.01
DR† [8]			1.64	0.01	3.89	0.02	0.83	0.01	2.45	0.01
Contr*† [6]			48.73	24.89	61.13	33.01	26.02	8.67	38.22	12.32
Contr* [6]	MobileNetV2	R101	47.10	18.00	61.50	28.80	21.80	6.30	37.70	8.80
Reg† [6]			50.27	30.40	63.66	34.01	28.85	11.22	39.77	12.33
Reg [6]			49.20	26.50	65.00	34.60	23.30	7.80	40.70	12.70
Ours			**64.12**	**39.38**	**76.16**	**44.40**	**37.53**	**17.73**	**54.29**	**18.08**
RKD† [31]			1.71	0.01	4.33	0.04	0.72	0.01	2.59	0.01
DR† [8]			1.85	0.01	4.01	0.03	0.67	0.01	2.36	0.01
Contr*† [6]			47.70	26.25	62.57	32.96	22.18	4.18	39.37	13.01
Contr* [6]	EfficientNetB3	R101	45.20	24.70	63.70	32.80	19.60	12.20	40.90	12.50
Reg† [6]			56.25	36.45	66.20	39.90	34.78	15.63	42.85	16.67
Reg [6]			52.90	29.70	65.20	39.00	27.80	10.40	42.40	16.00
Ours			**65.16**	**43.05**	**75.94**	**46.76**	**38.62**	**18.81**	**53.05**	**19.43**
(C) Training with R101-DELG as gallery model and GLDv2 as training dataset										
RKD† [31]			1.64	0.01	4.10	0.02	0.83	0.01	2.57	0.01
DR† [8]			1.52	0.01	3.76	0.01	0.81	0.01	2.32	0.01
Contr*† [6]			66.42	45.76	83.13	53.10	45.99	23.34	66.79	30.24
Reg† [6]	MobileNetV2	R101	72.75	56.03	85.81	65.23	53.07	32.21	69.96	39.29
HVS† [12]			74.39	58.24	86.86	67.44	54.68	34.77	72.42	43.39
LCE† [28]			75.45	58.03	87.24	67.30	54.95	33.88	73.03	43.01
Ours			**76.01**	**58.42**	**87.55**	**69.24**	**57.61**	**36.58**	**74.82**	**45.67**
RKD† [31]			1.60	0.01	3.83	0.03	0.73	0.01	2.41	0.01
DR† [8]			2.09	0.01	3.59	0.02	0.78	0.01	2.26	0.01
Contr*† [6]			69.45	49.70	83.81	59.36	46.19	26.49	68.15	35.24
Reg† [6]	EfficientNetB3	R101	74.60	59.88	86.09	67.69	53.41	33.31	72.21	42.63
HVS† [12]			76.41	62.72	87.07	71.54	56.13	36.86	74.53	49.09
LCE† [28]			75.89	61.90	86.63	70.98	55.21	36.53	73.62	48.94
Ours			**77.44**	**63.21**	**87.94**	**73.37**	**58.97**	**38.20**	**75.68**	**50.09**

Table 6. mAP (*asymmetric retrieval*) comparison against existing methods on the full benchmark. Black bold: best results. †: our re-implementation; R101: ResNet101 [14]. Training datasets for gallery models are shown in brackets.

hours to train a model, while the training of our model only needs 2 hours. Compared with HVS [12] and LCE [28], our method converges much faster. We only need 5 epochs (1 day) of training on GLDv2 to reach the optimal performance, while they require 20 epochs (about 3 days).

5. Conclusion

In this paper, we propose a flexible contextual similarity distillation framework for *asymmetric retrieval*. During the query model training, a new contextual similarity consistency constraint is adopted to preserve the contextual similarity between each training sample and its neighboring anchors. Optimizing this constraint is consistent with optimizing both first-order feature preserving and second-order ranking list preserving losses. The proposed framework can be trained using unlabeled datasets even from a different domain, which shows the generalizability of our approach. Extensive experiments demonstrate superior performance of our approach over existing state-of-the-art methods under the *asymmetric retrieval* setting.

Limitation. In our framework, the gallery model is kept frozen without being optimized simultaneously when adapting the lightweight model. As a result, the performance of the lightweight model is heavily dependent on that of the gallery model. In the future, we will explore how to optimize both gallery and query models to achieve better retrieval performance and efficiency.

Acknowledgements. This work was supported in part by the National Key R&D Program of China under contract 2018YFB1402605, in part by the National Natural Science Foundation of China under Contract 62102128 and 62021001, and in part by the Youth Innovation Promotion Association CAS under Grant 2018497. It was also supported by the GPU cluster built by MCC Lab of Information Science and Technology Institution, USTC.

References

[1] Soroush Abbasi Koohpayegani, Ajinkya Tejankar, and Hamed Pirsiavash. Compress: Self-supervised learning by compressing representations. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 12980–12992, 2020. 3

[2] Polino Antonio, Razvan Pascanu, and Alistarh Dan. Model compression via distillation and quantization. In *International Conference on Learning Representations (ICLR)*, pages 1–12, 2016. 2

[3] Artem Babenko and Victor Lempitsky. Aggregating deep convolutional features for image retrieval. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1269–1277, 2015. 2

[4] Artem Babenko, Anton Slesarev, Alexandr Chigorin, and Victor Lempitsky. Neural codes for image retrieval. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 584–599, 2014. 1

[5] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. SURF: Speeded up robust features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 404–417, 2006. 2

[6] Mateusz Budnik and Yannis Avrithis. Asymmetric metric learning for knowledge transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8228–8238, 2021. 1, 2, 5, 6, 7, 8

[7] Bingyi Cao, André Araujo, and Jack Sim. Unifying deep local and global features for image search. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 726–743, 2020. 5, 8

[8] Yuntao Chen, Naiyan Wang, and Zhaoxiang Zhang. DarkRank: Accelerating deep metric learning via cross sample similarities transfer. *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2018. 3, 8

[9] Ondrej Chum, James Philbin, Josef Sivic, Michael Isard, and Andrew Zisserman. Total Recall: Automatic query expansion with a generative feature model for object retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2007. 2

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, 2009. 7

[11] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. ArcFace: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[12] Rahul Duggal, Hao Zhou, Shuo Yang, Yuanjun Xiong, Wei Xia, Zhuowen Tu, and Stefano Soatto. Compatibility-aware heterogeneous visual search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10723–10732, 2021. 1, 2, 3, 7, 8

[13] Zhiyuan Fang, Jianfeng Wang, Lijuan Wang, Lei Zhang, Yezhou Yang, and Zicheng Liu. Seed: Self-supervised distillation for visual representation. In *International Conference on Learning Representations (ICLR)*, 2021. 3

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2, 8

[15] Yihui He, Ji Lin, Zhijian Liu, Hanrui Wang, Li-Jia Li, and Song Han. AMC : Automl for model compression and acceleration on mobile devices. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 784–800, 2018. 2

[16] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 3

[17] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. MobileNets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 3, 5

[18] Jie Hu, Rongrong Ji, Hong Liu, Shengchuan Zhang, Cheng Deng, and Qi Tian. Towards visual feature translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[19] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. SqueezeNet: AlexNet-level accuracy with 50× fewer parameters and <0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016. 3

[20] Hervé Jégou, Matthijs Douze, Jorge Sánchez, Patrick Pérez, and Cordelia Schmid. Aggregating local image descriptors into compact codes. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, pages 1704–1716, 2011. 2

[21] Herve Jegou, Matthijs Douze, and Cordelia Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 304–317, 2008. 2

[22] Hervé Jégou, Matthijs Douze, and Cordelia Schmid. Product quantization for nearest neighbor search. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, pages 117–128, 2011. 7

[23] Herve Jegou, Hedi Harzallah, and Cordelia Schmid. A contextual dissimilarity measure for accurate and efficient image search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2007. 3

[24] Yannis Kalantidis, Clayton Mellina, and Simon Osindero. Cross-dimensional weighting for aggregated deep convolutional features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 685–701, 2016. 2

[25] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2012. 2

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

[27] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. ShuffleNet v2: Practical guidelines for efficient cnn architec-

ture design. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 116–131, 2018. 3, 5

[28] Qiang Meng, Chixiang Zhang, Xiaoqiang Xu, and Feng Zhou. Learning compatible embeddings. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9939–9948, 2021. 2, 8

[29] Tony Ng, Vassileios Balntas, Yurun Tian, and Krystian Mikolajczyk. SOLAR: second-order loss and attention for image retrieval. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 253–270, 2020. 2

[30] Jianbo Ouyang, Wengang Zhou, Min Wang, Qi Tian, and Houqiang Li. Collaborative image relevance learning for visual re-ranking. *IEEE Transactions on Multimedia (TMM)*, 2020. 3

[31] Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 3, 8

[32] Nikolaos Passalis and Anastasios Tefas. Learning deep representations with probabilistic knowledge transfer. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 268–284, 2018. 3

[33] Florent Perronnin, Yan Liu, Jorge Sánchez, and Hervé Poirier. Large-scale image retrieval with compressed fisher vectors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3384–3391, 2010. 2

[34] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2007. 2, 5

[35] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008. 5

[36] Filip Radenović, Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondřej Chum. Revisiting oxford and paris: Large-scale image retrieval benchmarking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5706–5715, 2018. 5

[37] Filip Radenović, Giorgos Tolias, and Ondřej Chum. Fine-tuning cnn image retrieval with no human annotation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, pages 1655–1668, 2018. 1, 2, 5, 6, 7, 8

[38] Jerome Revaud, Jon Almazán, Rafael S Rezende, and Cesar Roberto de Souza. Learning with average precision: Training image retrieval with a listwise loss. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5107–5116, 2019. 2

[39] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. MobileNetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3, 5, 6, 7

[40] Yantao Shen, Yuanjun Xiong, Wei Xia, and Stefano Soatto. Towards backward-compatible representation learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2

[41] Oriane Simeoni, Yannis Avrithis, and Ondrej Chum. Local features and visual words emerge in activations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1

[42] Josef Sivic and Andrew Zisserman. Video google: A text retrieval approach to object matching in videos. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1470–1470, 2003. 2

[43] Mingxing Tan and Quoc Le. EfficientNet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning (ICML)*, pages 6105–6114, 2019. 3, 5

[44] Yurun Tian, Xin Yu, Bin Fan, Fuchao Wu, Huub Heijnen, and Vassileios Balntas. SOSNet: Second order similarity regularization for local descriptor learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[45] Giorgos Tolias, Yannis Avrithis, and Hervé Jégou. To aggregate or not to aggregate: Selective match kernels for image search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1401–1408, 2013. 2

[46] Giorgos Tolias, Tomas Jenicek, and Ondřej Chum. Learning and aggregating deep local descriptors for instance-level recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 460–477, 2020. 1

[47] Giorgos Tolias, Ronan Sicre, and Hervé Jégou. Particular Object Retrieval With Integral Max-Pooling of CNN Activations. In *International Conference on Learning Representations (ICLR)*, pages 1–12, 2016. 1, 2

[48] Tobias Weyand, Andre Araujo, Bingyi Cao, and Jack Sim. Google landmarks dataset v2 a large-scale benchmark for instance-level recognition and retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2575–2584, 2020. 5

[49] Hui Wu, Min Wang, Wengang Zhou, and Houqiang Li. Learning deep local features with multiple dynamic attentions for large-scale image retrieval. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 11416–11425, 2021. 1

[50] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. ShuffleNet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6848–6856, 2018. 3, 5

[51] Wengang Zhou, Yijuan Lu, Houqiang Li, Yibing Song, and Qi Tian. Spatial coding for large scale partial-duplicate web image search. In *Proceedings of the ACM international conference on Multimedia (MM)*, pages 511–520, 2010. 2