Data-Efficient Ranking Distillation for Image Retrieval

Zakaria Laskar[1] and Juho Kannala[1]

Aalto University, Finland
firstname.lastname@aalto.fi

Abstract. Recent advances in deep learning has lead to rapid developments in the field of image retrieval. However, the best performing architectures incur significant computational cost. The paper addresses this issue using knowledge distillation for metric learning problems. Unlike previous approaches, our proposed method jointly addresses the following constraints: i) limited queries to teacher model, ii) black box teacher model with access to the final output representation, and iii) small fraction of original training data without any ground-truth labels. In addition, the distillation method does not require the student and teacher to have same dimensionality. The key idea is to augment the original training set with additional samples by performing linear interpolation in the final output representation space. In low training sample settings, our approach outperforms the fully supervised baseline approach on ROxford5k and RParis6k with the least possible teacher supervision.

1 Introduction

Instance level image retrieval have dramatically improved with the advent of Convolutional Neural Networks (CNN) [1–3]. The improvement in performance is particularly driven by deeper networks such as VGG [4], ResNet [5] family of networks. However, with the increased accuracy also comes higher inference time and computational burden at test time. There are two main ideas that have been proposed to address this challenge. One is to quantize(and/or prune) the trained bigger network to a lighter version with reduced precision and weights but with the same depth [6]. The other direction is to transfer knowledge from the bigger network (teacher model) to a different but much smaller and lighter network (student model). In this paper, we focus on the second direction, popularly known as Knowledge Distillation (KD)[7, 8], although it can be applied to the former case as well.

The idea of using information from a teacher model(s) to train a student model was first proposed by Caruana [7], and, was later improved upon by Hinton [8]. Instead of providing a one hot vector as target or ground-truth class label, KD aims to distill additional information from the teacher to the student model. Such additional knowledge is usually constituted by the output at various layers of the teacher model, e.g. logits from the layer before the softmax in the teacher constitute softer targets for the student. Traditionally proposed for classification problem, KD was later extended to the metric learning scenario [9, 10]. However, the knowledge being distilled was addressed differently as traditional KD methods did not perform well in this setting [9, 10]. While [9] proposed to distill the teacher ranking, [10] proposed to distill the teacher distance for a given query-database sample(s). In both cases, the student model tries to learn the relation (rank/distance) between query-database samples instead of learning the exact input-output mapping. This allows the student network to maintain its own output dimensionality. We refer to these methods as Metric Knowledge Distillation (MKD) methods.

In this paper we address MKD from the perspective of data-efficiency. While existing distillation approaches [9, 10] have addressed test time efficiency by compressing the knowledge from cumbersome models onto compact ones, they have failed to address training time efficiency. We address this issue by defining the training time efficiency as i) the number of queries to the teacher model to obtain teacher knowledge (pseudo ground-truth) in the form of final output representation or logits, and ii) the number of training samples required to distill the teacher knowledge onto the student model. In this paper we propose an MKD method under the above mentioned budget constraints while operating under the setting of black-box teacher models, preserving student-teacher dimensionality and achieving comparable performance to the no-budget scenario. Large scale datasets are costly in terms of memory (storage), computation (training) and economic (data accumulation/labelling). In addition private data such as trained teacher models or full training dataset can have limited or partial access due to privacy concerns. Our proposed method reduces the dependency on large scale datasets for learning new models while being efficient also during training.

The key ingredient in our proposed method is the idea of mixup [11, 12]. Using mixup, one can augment a small original training set with large number of additional samples by convex combination of the training samples. Such idea has been recently used in [13] to address data-efficient knowledge distillation using mixup based data augmentations. While the existing mixup based methods have addressed classification problems, we extend the idea to the problem of metric learning. In contrast to [11–13], we perform mixup at the global image representation level. That is, each image is represented by a global representation vector obtained by spatial encoding of 2D representation maps from a CNN. Thereafter, augmented representations are obtained by linearly interpolating between representations from original samples. We then perform distillation between teacher and student models in the joint space of original and augmented global representations. In particular we train the student model to mimic the teacher ranking for each sample in the joint representation space of respective models using the recently proposed ranking loss [14]. Representation level mixup requires orders of magnitude less queries to the teacher model compared to mixup at the input image level [13]. In the process our proposed method still achieves comparable performance to fully supervised models trained on the full training dataset.

2 Related Work

We describe the image retrieval and knowledge distillation based related work in this section

Image Retrieval Before the advent of CNNs, earlier works addressed the image retrieval problem using SIFT [15] based local descriptors [16], and additionally encoded into global representations [17, 18] followed by geometric verification [19]. Since then CNN based methods [1, 2, 20–22, 3] have improved the performance on challenging benchmarks such as Oxford5k and Paris6k [23]. The success can be attributed to large datasets [24] , improved global encoding methods [25–27, 22], and robust ranking loss functions [28, 29, 2, 14]. In particular, the Average Precision (AP) loss [14] has the least training compute overhead by avoiding hard negative mining with joint ranking supervision over large batches of images.

Knowledge Distillation Knowledge distillation can be traced back to the work of Breiman *et al.* [30] where the knowledge of multiple tree models were distilled onto a single model. The term knowledge distillation (KD) was itself coined by Hinton *et al.* in his work [8]. In addition to the standard supervisory signals, the student model was additionally trained to match the softmax distribution of a heavier teacher model. Since then, several works have been proposed that provide additional information apart from the softmax logits. [31] and [32] propose to transfer attention maps. Self-distillation where the student and the teacher share the same network architecture have also been proposed [33, 34].

For image retrieval and metric learning in general, Chen *et al.* [9] propose to transfer rank information. Similarly, Park *et al.* [10] proposed to distill distance information between teacher and student models. Both these methods show improvements over standard KD approaches.

Mixup. Mixup based regularizers were first proposed by [11, 35]. Later, mixup based interpolation was extended to hidden representations of a CNN by several works [12, 36]. Recently, [13] proposed mixup based augmentation for data-efficient knowledge distillation. For each mixed sample, the above approaches require a new feed-forward pass through the CNN. Instead our approach performs mixing of the global vector representations requiring just a single feed-forward pass. Mixed samples can be obtained by simply interpolating between original global representations.

3 Proposed Method

In this section we propose an algorithm to train a compact student model, S by distilling the knowledge from a cumbersome teacher model, T. The key idea of knowledge distillation in classification problem is to use soft labels from T as targets in addition to ground-truth hard labels. Soft labels encode semantic similarity by providing inter class similarity information. However, we consider a general scenario where ground-truth labels are not known apriori. In addition metric learning involves optimizing the representation space directly without explicit label prediction as in classification problems. Thus it is not clear how to generate and incorporate teacher soft labels for unsupervised knowledge distillation in the domain of metric learning problems.

First we present preliminaries followed by the data augmentation algorithm to address low training sample complexity. This is followed by teacher label generation and computing the ranking loss to train the student model. Finally, we present the algorithm combining the above steps in a single framework.

3.1 Preliminaries

Given a batch of images, $B = \{I_1, ..I_j..I_B\}$, we obtain teacher and student $l2$ normalized output representations, $f_B^T = \{f_b^T\}_{b=1}^{|B|} \in \mathbb{R}^{N_T \times |B|}$ and $f_B^S = \{f_b^S\}_{b=1}^{|B|} \in \mathbb{R}^{N_S \times |B|}$, where $f_b^T = e^T(T(I_b)) \in \mathbb{R}^{N_T}$, $f_b^S = e^S(S(I_b)) \in \mathbb{R}^{N_S}$. $T(.)$ and $S(.)$ are teacher and student convolutional neural networks respectively, with N_T, N_S being their respective final output dimensionality. As database size in image retrieval problems tends towards millions, it is common practise to store global representations per image by encoding the 3D representation map from the CNNs into 1D vectors. Popular encoding methods from literature include GeM [37], MAC [1], RMAC [26]. As we consider the teacher model as black-box with access to the final vector encoded global representation, we represent the student and teacher encoding functions separately with e^T and e^S respectively. These encoding functions can represent any of the above mentioned encoding methods. As the global representations are $l2$ normalized, a simple dot product is used to compute similarity values.

3.2 Database Augmentation

Acquiring large training datasets and labelling the ground-truth incurs large computational resources, huge memory footprint and high economic costs. We address this using knowledge distillation by replacing large datasets with models trained on them and a small amount of the original training samples. This also addresses practical scenarios where teacher models have limited access rights, or the whole training set is not made public. Furthermore, extracting representations for the whole dataset using both the teacher and student model is inefficient as it leads to increased training costs.

Given a small amount of training samples, D, we extract both teacher and student global representations, f^T and f^S for a given batch, B. We augment the representations from each batch using mixup [11],[12]. These works perform mixup at the local level, while we perform mixup at the global representation level. In particular, given representations for images, $I_i, I_j \in B$, we perform representation mixup as follows:

$$f_{ij} = \lambda f_i + (1 - \lambda) f_j, \tag{1}$$

where $\lambda \sim beta(\alpha, \alpha)$ is the mixing coefficient. Instead of sampling λ per training sample, we only sample a single value of λ per batch. The mixed representations are further $l2$ normalized.

There are several benefits to performing mixup at the global representation level. It is to be noted that since we consider black-box teacher models, we can only consider *InputMixup* [11] and not *ManifoldMixup* [12] as the later requires access to intermediate representation maps. However, *InputMixup* which performs mixup at the input image level, requires a new feed-forward pass of the mixed input image through the network to obtain representations. This increases the number of queries to the teacher model to be much more than $|D|$. The same applies for the student network, which in total significantly increase the training cost. The same costs applies for *ManifoldMixup* considering white-box teacher models. In contrast, global representation mixup only requires at most $|D|$ queries to the teacher model, and large amount additional representations can be obtained at a marginal overhead cost. To give the reader an estimate, given a batch of $B = 1000$ images, mixing each image with $R = 10$ other images from the batch will result in 10000 samples. For *InputMixup* this will require 10000 $\times ep$ queries to the teacher model where ep is the number of training epochs. Our approach will only require 1000 queries.

3.3 Label Generation

Previously, mixup has been addressed in classification domains [11, 12, 38]. In such settings, the label of the mixed sample is obtained by linear interpolation of respective labels of the original samples by λ. In this section, we show how to generate labels for student model using global teacher representations.

Let the joint set of representations be $F^T = f^T_B \bigcup f^T_{B'}$, $F^S = f^S_B \bigcup f^S_{B'}$, where $f^T_{B'}$,$f^S_{B'}$ are the augmented teacher and student representations. Let $Z = \{1, 2, ..., (|B|+|B'|)\}$ be the joint sample index set. We are interested in computing a binary label matrix $Y \in \mathbb{R}^{|Z \times |Z|}$, where each row $Y_q \in \mathbb{R}^{1\times|Z|}$ represents the label vector corresponding to the q^{th} representation. Before we explain how the binary values are computed for Y_q, we first formally define a positive index set $P_q \subseteq Z$ such that $\forall z \in Z$

$$Y_q(z) = \begin{cases} 1, & \text{if } z \in P_q \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

The matrix Y is symmetric ($Y = Y^\intercal$) i.e. $Y_q(z) = Y_z(q)$. The binary label 1 signifies the corresponding representations, f^T_q, f^T_z are similar. Consequently, the corresponding student representations, f^S_q, f^S_z are trained to be similar. The measure of similarity is defined next where we present methods to compute the elements of P_q.

Similarity Labelling (SL). The first measure of similarity is based on cosine similarity in the representation space. We first compute the teacher and student similarity matrices, $S^T = (F^T)^\intercal(F^T)$, $S^S = (F^S)^\intercal(F^S) \in \mathbb{R}^{|Z|\times|Z|}$. The positive set P_q constitutes the Euclidean Nearest Neighbors (ENN) and is computed as $P_q = \{z \mid S^T_q(z) > \tau\} \forall z \in Z$, where $\tau \in [0, 1]$ is a similarity threshold. We call this similarity based labelling Similarity Labelling (SL). If τ is too high, P_q will only contain near duplicate representations, while keeping it too low will include

too many false positives. We experiment with different values of τ and found that optimal performance is achieved with moderate values of τ (c.f. Sec. 6).
Mixup Labelling. As observed the positive set, P_q under similarity labelling is constituted by the ENN. Thus for representations falling in low density regions, the positive sets will be empty or have low cardinality. Empty positive sets means zero loss and thus no gradient to train the model. This becomes an issue if most of the samples fall in such low density regions (c.f. Sec. 7). To address this issue we introduce mixup labelling (ML) based on the following assumption: the global representations contain semantic concepts, so mixed representation will be closer to the positive sets of representations being mixed, than the other representations. Formally, if representations, f_k^T, f_r^T are mixed resulting in the corresponding mixed representation, f_{kr}^T indexed at $kr \in Z$, and the corresponding positive sets obtained from similarity labelling be P_k, P_r and P_{kr} respectively. Then $P_{kr} = P_{kr} \cup P_k \cup P_r$. Thereafter, using Eq. 2 the label matrix, Y is computed while maintaining matrix symmetry.

3.4 Loss Function

In this section we show how to compute the loss given the student similarity matrix, S^S and teacher label matrix, Y. To realize this, we use the listwise loss, known as Average Precision Loss (AP) [14]. The loss maximizes the distance between histogram of positive and negative similarity scores. For brevity we elaborate the loss function below:

The similarity interval $S_q^S = [0,1]^{1\times|Z|}$ is divided into $C-1$ bins of width $\Delta c = \frac{2}{C-1}$. Let $c_b = 1-(b-1)\Delta c$, b=1...C represent the center of the b^{th} bin. Average Precision is generally computed at each rank, $r = 1...|Z|$. However, as rank assignment is non-differentiable, the images are instead assigned to bins using the soft bin assignment as follows:

$$p(S_q^S(i), b) = \max\left(1 - \frac{|S_q^S(i) - c_b|}{\Delta}, 0\right). \tag{3}$$

Here, p represents the probability that the i^{th} image occupies the b^{th} bin conditioned on its similarity to the query, q based on S_q^S. The AP is then computed in each bin as follows:

$$Pr(S_q^S, Y_q, b) = \frac{\sum_{b'=1}^{b} p(S_q^S, b')^\top Y_q}{\sum_{b'=1}^{b} p(S_q^S, b')^\top \mathbf{1}}, \tag{4}$$

$$\Delta Rc(S_q^S, Y_q, b) = \frac{p(S_q^S, b)^\top Y_q}{N_q}, \tag{5}$$

The AP loss for each q is computed as :

$$AP(q) = 1 - \sum_{b=1}^{C} Pr(S_q^S, Y_q, b)\ \Delta Rc(S_q^S, Y_q, b). \tag{6}$$

Algorithm 1 Rank Distillation

```
REQUIRE: Teacher & Student representation $f_B^T = \{f_i^T\}_{i=1}^{B}, f_B^S = \{f_i^S\}_{i=1}^{B}$
REQUIRE:Labelling functions: $SL(.), ML(.)$
REQUIRE:Loss function: $AP(.)$
REQUIRE: $\tau, R, \lambda$
OUTPUT: Loss value
Initialize $L = \{\}$ ▷ store loss values.
$S_o^T \leftarrow (f_B^T)^\intercal(f_B^T)$
$P_o \leftarrow SL(S_o^T, \tau)$. ▷ Initial positive set.
for  $r = 1, 2..., R$ do
    Initialize $F^T = \{\}$, $F^S = \{\}$ ▷ store original & augmented samples.
    Initialize $MIX = \{\}$ ▷ store indices of mixing & mixed samples.
    $\{F_i^T\}_{i=1}^{|B|} \leftarrow \{f_i^T\}_{i=1}^{|B|}$, $\{F_i^S\}_{i=1}^{|B|} \leftarrow \{f_i^S\}_{i=1}^{|B|}$
    for $k = 1, 2..., |B|$ do
        Sample index $r_k$ from range $(1, |B|)$
        $F_{|B|+K}^T \leftarrow \lambda f_k^T + (1 - \lambda) f_{r_k}^T$
        $F_{|B|+K}^S \leftarrow \lambda f_k^S + (1 - \lambda) f_{r_k}^S$
        $F_{|B|+K}^S$.requires_grad = False ▷ Restrict back-propagation.
        $MIX$.store($k, r_k, |B| + k$) ▷ Store mixing information.
    end for
    $S^T \leftarrow (F^T)^\intercal(F^T)$, $S^S \leftarrow (F^S)^\intercal(F^S)$ ▷ Compute teacher & student similarity matrix.
    $P_s \leftarrow SL(S^T, \tau)$. ▷ Positive set based on SL.
    $P_m \leftarrow ML(P_o, MIX)$ ▷ Positive set based on ML.
    $P \leftarrow P_s \cup P_m$. ▷ Final positive set based.
    Compute $Y$ using $P$▷ Eq. 2
    $L_r \leftarrow AP(Y, S^S)$▷ Compute Average Precision Loss
end for
$L \leftarrow 1/R\sum_{r=1}^{R} L_r$ ▷ Total loss
return  $L$
```

The final loss function to be optimized is defined as :

$$L = \frac{1}{|Z|} \sum_{q=1}^{|Z|} AP(q) \tag{7}$$

3.5 Algorithm

Given the training set, D we first extract all the teacher representations, $f_D^T = \{f_i^T\}_{i=1}^{|D|}$. Thereafter for each epoch, *ep*, we sample a batch of B images from D. We then extract the student representations, $f_B^S = \{f_b^S\}_{b=1}^{|B|}$ and from f_D^T obtain the teacher representations $f_B^T = \{f_b^T\}_{b=1}^{|B|}$. We now proceed to compute the loss as described in Algorithm 1.

First, we compute the initial positive set, P_o based on similarity labelling (SL) and teacher representations, f_B^T. Next, we introduce the mixing iterator,

R. In each iteration, $r = 1, 2, ...R$ we iterate over the following steps: 1) Mix each student and teacher representation, $f_k^T, f_k^S, k = 1, 2..|B|$ with the representation, $f_r^T \in f_B^T, f_{r_k}^S \in f_B^S$ of a random sample r_k. The mixed representations are concatenated with the original representations resulting in the joint representation set, $F^T \in \mathbb{R}^{N_T \times 2|B|}$, $F^S \in \mathbb{R}^{N_S \times 2|B|}$, where the first $N \times |B|$ are the original representations while the bottom $N \times |B|$ are mixed representations respectively. 2) Simultaneously with the previous step we store the index information of mixing samples (k, r_k) and the mixed sample $(|B| + k)$, $\forall k = 1...|B|$ in the variable, MIX. 3) Given F^T, F^S we compute the teacher and student similarity matrices, S^T, S^S. 4) Next we proceed to label generation. Using the similarity threshold, τ and S^T we first compute the positive set, $P_s = \{P_k\}_{k=1}^{2|B|}$ based on similarity labelling (SL). Thereafter, using the mixing information in MIX and P_o we perform mixup labelling (ML) to compute the mixup positive set, $P_m = \{P_k\}_{k=|B|}^{2|B|}$. The final positive set, P is obtained by combining similarity and mixup positive sets. The label matrix, $Y \in \mathbb{R}^{2|B| \times 2|B|}$ is formed using P. 5) Finally, we compute the Average Precision (AP) loss, L_r using teacher label matrix, Y and student similarity matrix, S^S. After r iterations we have R loss values, $\{L_r\}_{r=1}^{R}$ which are then averaged followed by back-propagation. It is to be noted that the mixed representations are not used to back-propagate gradients (line 12 in Algorithm 1).

We now explain the rationale behind introducing R. Under the current setting, the number of mixed representations used to compute the final loss is $|B|R$. If these mixed representations were jointly used in computing the final loss the size of the similarity and label matrices will be $((|B| + |B|R)^2)$. For values of $|B|$=1000, R=10 used in this work, the size will be $\sim 10000^2$. Loss and gradient computation becomes considerably slow under this setting. Instead, by dividing the loss computation into R steps, we are still able to leverage $|B|R$ mixed representations, while the final similarity matrices are of size $(2|B|)^2 \sim 2000^2$. This leads to comparable performance while increasing training efficiency.

4 Implementation details

Training dataset. We use the training dataset used in [1]. The dataset was initially introduced in [24] and consists of 7.4 million internet photo collections of popular landmarks around the world. The images are passed through an SfM pipeline [1] to create clusters of images (class labels). This process results in 163k images clustered into about 700 classes. Training dataset consists of randomly selected 550 classes containing 133k images. We refer to this dataset as *SfMFr.*
Network Training. We used the publicly available trained Resnet101 models by Radenovic *et al.* [37] trained on *SfMFr* as teacher models, T. MobileNetV2 (MVNetV2) and Resnet34 pre-trained on ImageNet [39] are used as the student models S1 and S2 repsectively. We randomly sample $D = 4000$ images from *SfMFr*. The network is trained with a batch size, $B = 1000$ which are randomly sampled from the training set, D. We used Adam [40] optimizer with an initial learning rate of $l_0 = 1 \times 10^{-4}$, exponential decay $l_0 \exp(-0.01 \times ep)$ every epoch,

Table 1: Different networks with the number of respective parameters and time taken to process 1 image in multi-scale mode.

Method	Param	Time/Image
ResNet101	42M	60ms
ResNet34	21M	20ms
MVNetV2	1.8M	10ms

ep. Weight decay was set to 1×10^{-6}. Images were rescaled to 362 pixels on the longest side while maintaining the aspect ratio. Training is done for 30 epochs on GeForce RTX GPU with 11GB memory. We use generalized mean pooling (GeM) [37] to obtain global representations for each image. The global descriptors are subsequently $l2$ normalized.

We list the hyper-parameters associated with our algorithm are $\tau = 0.75$, $R = 10$. We use the same hyper-parameter settings for both the student networks, S1 and S2.

Baselines. We train MVNetV2, Resnet34 using contrastive loss (CL) and Average Precision (AP) loss on *SfMFr* dataset. For CL, we mine hard negatives every epoch from a random pool of 22K images, and keep top 5 negatives. Margin is set to 0.65. Batch size are 5 and 4000 for CL and AP respectively. The learning rate was set to 5×10^{-7} for CL and 1×10^{-4} for AP.

Test dataset We evaluate our approaches on standard image retrieval benchmark datasets, Oxford5k (Oxf) [41],Paris6k (Par) [42], ROxford5k (ROxf) [23], and RParis (RPar) [23] datasets. The evaluation metric is mean Average Precision (mAP). The test sets consists of 55 queries and several thousand database images (Oxf: 5k, Par: 6k), while their revisited counterparts have 70 queries each with 4k and 6k database images respectively. The revisited datasets also have 3 splits: Easy (E), Medium (M), and Hard (H) defining the difficulty level of retrieving the corresponding database images in the set. The queries are annotated with a bounding box specifying the landmark of interest. Similar to prior works, we crop the query images with bounding box.

During evaluation, we extract multi-scale global representations with the scales: 1, $1/\sqrt{2}$, and $1/2$. The resulting descriptors are combined using GeM pooling. The resulting vector is $l2$ normalized. Furthermore, due to low sample complexity we do not use any validation data during training. Instead, we perform weight averaging [43] to combine model performances from different epochs. In particular, the final student network used for evaluation is obtained by averaging the weights of the trained student models from the 20^{th} and 30^{th} epoch.

The number of parameters and average multi-scale inference time during evaluation are presented in Tab. 1 for teacher and student models.

Table 2: Performance comparison of compact student networks MobileNetV2 (S1,MVnetV2) and Resnet34 (S2,ResNet34) trained using our method, and baseline methods : without augmentation (no -aug), Average Precision (AP), contrastive loss (CL). Evaluation is done on image retrieval datasets : Oxford (Oxf), Paris (Par), ROxford (ROxf), RParis (RPar). The revisited datasets, ROxf, RPar are evaluated using Easy (E), Medium (M) and Hard (H) splits. Evaluation metric is mAP. T/S denotes the teacher /student role of the model. Our method does not require training labels.

Method	$\|D\|$	Oxf	ROxf			Par	RPar		
			E	M	H		E	M	H
Resnet101 (T,CL)	120k	81.2	73.8	55.8	27.4	87.8	86.5	70.0	44.8
Compact student networks									
MVnetV2 (CL)	120k	74.5	66.5	48.9	20.8	85.7	**84.6**	**66.2**	39.0
MVnetV2 (AP)	120k	74.2	67.2	51.0	24.3	85.0	83.7	65.6	39.5
MVNetV2 (S1, no-aug)	4k	76.1	67.0	51.0	25.8	**84.6**	84.0	66.1	**40.4**
MVnetV2 (S1)	4k	**78.7**	**70.8**	**53.8**	**26.9**	84.0	82.1	65.0	39.4
ResNet34 (CL)	120k	77.9	70.7	51.9	23.1	**86.5**	**85.9**	**69.5**	**44.0**
ResNet34 (AP)	120k	79.6	70.5	53.3	24.9	86.3	**86.4**	69.2	43.1
ResNet34 (S2,no-aug)	4k	77.3	70.7	50.8	22.5	84.9	83.8	68.0	42.8
ResNet34 (S2)	4k	**78.1**	**74.1**	**55.1**	**25.9**	85.4	84.2	68.6	43.8

5 Results

In this section we compare our proposed algorithms on the standard retrieval datasets. In addition we also compare with baseline methods and perform detailed ablation study.

Baseline comparison. We compare the performance of the student models, S1: MVNetV2, S2: ResNet34 trained using our proposed method with the teacher model, T: ResNet101 and the same student models trained without the proposed augmentation. In addition we also consider student models trained with ground-truth labels on the full dataset with loss functions: contrastive loss (CL) and average precision (AP). Results are presented in Tab. 2. Results show that student models using our proposed method are able to match the performance of the supervised counterparts. It is to be noted that our method was only trained on 4k images. Compared to it, the supervised models based on CL and AP losses were trained using the full dataset of 120k images. Among the student models, ResNet34 outperforms MVNetV2 by 2-3% on ROxford and RParis datasets. This can be attributed to the higher capacity of ResNet34 model (c.f. Tab. 1).

Furthermore, student models trained without the proposed global representation augmentation performs poorly compared to our proposed method with augmentation. Baseline student models trained only on $D = 4k$ dataset with AP loss and full ground-truth label supervision has similar performance to the no-augmentations setting. The decrease in performance in low sample setting

Table 3: Comparison of different mixup methods in terms of performance and computational costs. Mixed indicates the number of mixed samples generated. Cost represents the number of feed-forward pass required through the teacher model. Similar costs also extend to student model. The number of original training samples is 4k.

Method	Mixed	Cost	ROxf			RPar		
			E	M	H	E	M	H
Ours,no-aug	0k	4k	70.7	50.8	22.5	83.8	68.0	42.8
Inp Mix	1k	5k	71.9	52.3	24.4	83.1	67.1	40.5
Inp Mix	8k	12k	72.6	52.7	24.9	82.2	64.7	37.1
Inp Mix	32k	36k	73.4	53.1	24.5	81.7	65.0	38.2
Ours, with-aug	600k	4k	**74.1**	**55.1**	**25.9**	**84.2**	**68.6**	**43.8**

can be attributed to the fact that in a randomly selected training set D, large number of samples, $q \in D$ have empty or very small sized positive set P_q. Thus, without any positives there is no error signal that can lead to learning representations from these images. Augmentation addresses this issue by generating positives from the mixed samples.

The computational gains of the proposed global mixup over the baseline input mixup approach is shown in Tab. 3. Generating more samples using input mixup leads to improvement in ROxford dataset while it decreases on RParis dataset. We hypothesize this is due to limited number of mixed samples. However, increasing the number of input mixup augmentations incurs significant computational costs.

Table 4: mAP performance on ROxford (ROxf) and RParis (RPar) datasets. We present alongside each method, the model architecture (R:ResNet101, V:VGG16,A:AlexNet,R34:ResNet34 and M:MobileNetV2). In addition we also show the dimension of the final global representation from each model. It is to be noted that our proposed method only requires a fraction of the full supervised dataset and a trained teacher.

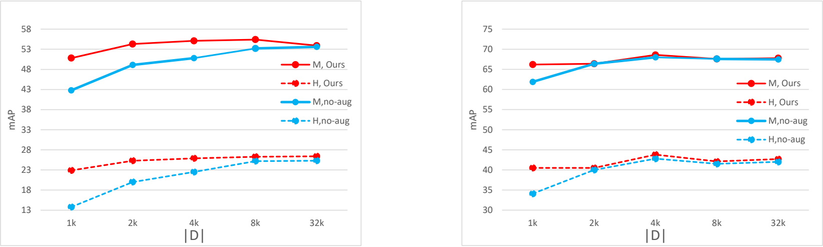
Method	Nw	Dim	ROxf				RPar			
			M		H		M		H	
			mAP	mP@10	mAP	mP@10	mAP	mP@10	mAP	mP@10
GeM [37]	R101	2048	64.7	84.7	38.5	53.0	76.9	98.1	55.4	89.1
AP [14]	R101	2048	**67.5**	-	**42.8**	-	**80.1**	-	**60.5**	-
GeM [37]	V	512	60.9	82.7	32.9	51.0	69.3	97.9	44.2	83.7
NetVLAD [22]	V	512	37.1	56.5	13.8	23.3	59.8	94.0	35.0	73.7
Ours	R34	512	55.4	79.9	29.1	46.3	68.7	96.6	43.7	83.4
Ours	M	320	51.1	74.0	24.9	38.6	67.3	96.1	41.1	80.0

State-of-the-art. State-of-the-art methods are compared in Tab. 4. Whitening is a standard post-processing step in all standard image retrieval methods as it reduces the impact of correlated features. While some [14] use unsupervised whitening based on PCA, others [37] use supervised whitening. We use PCA based whitening. In particular we use the square rooted PCA [44]. Similar to traditional practices, we learn PCA on Paris6k for evaluating the network on Oxford5k and vice-versa. In addition to mAP, we also report mean precision @10 (mP@10). In RPar PCA does not bring any improvement. However, in ROxf, PCA brings significant improvement both in terms of mAP : ResNet34 (ROxf,M : 55.1 → 55.4, ROxf,H : 25.9 → 29.1), and mP@10 : ResNet34 (ROxf,M : 76.6 → 79.9, ROxf,H : 39.8 → 46.3). However, there is an increase in the performance gap with the teacher model (GeM [37]). The performance difference can be attributed to the supervision in the whitening process. Compared to supervised models with similar dimensionality such as (GeM [37], V) that uses a VGG16 architecture, the performance gap is much smaller. Overall, the difference in student performance is well compensated by the reduced number of parameters and computation time for processing a single image in current multi-scale mode as shown in Tab. 1.

6 Hyper-parameter Ablation

In this section, we analyze and present detailed analysis on the impact of different hyper-parameters in retrieval performance. This is done by varying the concerned hyper-parameter while keeping the rest same as detailed in Sec. 4.

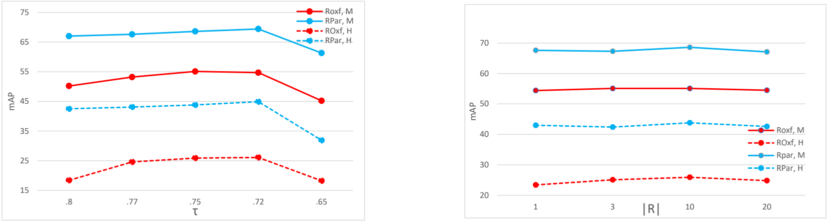
First we analyze the image retrieval performance by varying the size of the training set, D. In Fig. 1 we observe that our method consistently outperforms the baseline method trained without the representation augmentations. As the sample size increases, both the methods converge in performance.



(a) 1a

(b) 1b

Fig. 1: Figure shows the impact of sample size, $|D|$ on (a)Roxford and (b)RParis datasets for methods with and without the proposed augmentation based global mixup method.



(a) 1a

(b) 1b

Fig. 2: Figure shows the impact of different hyper-parameters τ (a) and R (b) in retrieval performance.

In Fig. 2a we study the impact of size of similarity threshold τ on the retrieval performance. As mentioned earlier, τ controls the amount of semantic information that is distilled from teacher onto student model. From Fig. 2a we observe that retrieval performance increases as τ is decreased. As explained earlier, high values of τ selects easy positives in the positive set, P. As we decrease this threshold, the hardness of positives increases. However, decreasing τ too much will allow false positives to get included in P which will be detrimental to the learning process. This is evident by the sharp decrease in retrieval performance for $\tau = 0.65$.

Finally in Fig. 2b we study the impact of R. We notice a marginal but consistent improvement in retrieval performance across both datasets as R is increased. In particular for the Hard (H) setting, the performance improves by 2-3 % as R increases from 1 to 10. Beyond $R = 10$ there is a marginal drop in performance.

The above experiments are in line with our motivation to apply the given hyper-parameters and also shows that beyond certain values, our proposed methods are not sensitive to the choice of hyper-parameter values.

7 Training Ablation

In this section we study some of the key components in the training algorithm. In our experiments, the teacher model uses GeM encoding method. Both encoding methods GeM/MAC produce comparable performance on ROxf,M : 55.1/55.0, ROxf,H : 25.9/25.4, RPar,M : 68.6/67.1, and RPar,H : 43.8/42.2. This demonstrates that our algorithm is robust to the selected global encoding method.

Next, we analyze the retrieval performance of models trained on a dataset of size, $|D| = 2000$ with/without back-propagating gradients beyond the level of mixed global representations : ROxf,M : 54.3/49.7, ROxf,H : 25.2/21.8, RPar,M : 66.4/65.5, and RPar,H : 40.5/37.6. Back-propagating beyond mixed representations results in over-fitting and decrease in retrieval performance.

Thirdly, we analyze the scenario where the original teacher representations are sparesely conncected. As such the mixed samples are located in low den-

Table 5: mAP performance with and without mixup labelling (ML). Note that for this experiment, D was set to 2000.

Method	ROxf			RPar		
	E	M	H	E	M	H
ResNet34 ($\tau = 0.75$)	**71.5**	**50.9**	**21.9**	**83.6**	**64.6**	37.6
ResNet34 (no-ML,$\tau = 0.75$)	66.7	47.2	18.5	82.8	63.9	**37.8**
ResNet34 (no-ML,$\tau = 0.65$)	64.2	46.3	17.7	78.0	59.5	30.4
ResNet34 (no-ML,$\tau = 0.5$)	57.1	39.5	13.9	73.0	56.0	26.5
ResNet34 (no-aug,$\tau = 0.65$)	64.4	46.8	20.9	79.6	62.4	36.1

sity regions resulting in most samples having empty positive sets from similarity labels (SL) alone. In such settings we expect the mixup labeling (ML) to provide training signals that can drive the learning process. Results are shown in Tab. 5. For this setting, we sample 2000 images from the full dataset such that each sample has atmost 3 Euclidean Nearest Neighbors. Results demonstrate that the mixup labelling significantly improves the retrieval performance across all settings in ROxford5k dataset. On RParis6k, both methods have comparable performance. No ML setting is compared under different teacher similarity thresholds to demonstrate that simply decreasing τ to increase occupancy of positive set, P does not lead to improvement in performance. In addition, our method also outperforms the baseline setting without the proposed augmentations across both datasets.

8 Conclusion

We have presented a knowledge distillation approach based on ranking distillation. The proposed approach transfers the ranking knowledge of a list of images from a cumbersome teacher onto a compact student model. The proposed method introduces key algorithmic design choices that make the approach data-efficient under budget constraints w.r.t access to black-box teacher model and the number of training samples.

Our results are comparable or better than the standard supervised methods with the same network architecture that are trained using full dataset. Under the training budget constraints, the proposed method clearly outperforms the baseline methods on challenging image retrieval datasets. Our approach finds use case in settings where teacher models are hosted as public APIs with limited access.

References

1. Radenović, F., Tolias, G., Chum, O.: Cnn image retrieval learns from bow: Unsupervised fine-tuning with hard examples. In: Proc. ECCV. (2016)

2. Gordo, A., Almazán, J., Revaud, J., Larlus, D.: Deep image retrieval: Learning global representations for image search. In: Proc. ECCV. (2016)
3. Teichmann, M., Araujo, A., Zhu, M., Sim, J.: Detect-to-retrieve: Efficient regional aggregation for image search. In: Proc. CVPR. (2019)
4. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
5. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proc. CVPR. (2016)
6. Yang, J., Shen, X., Xing, J., Tian, X., Li, H., Deng, B., Huang, J., Hua, X.s.: Quantization networks. In: Proc. CVPR. (2019)
7. Buciluǎ, C., Caruana, R., Niculescu-Mizil, A.: Model compression. In: Proc. SIGKDD. (2006)
8. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. Proc. NIPSW (2015)
9. Chen, Y., Wang, N., Zhang, Z.: Darkrank: Accelerating deep metric learning via cross sample similarities transfer. In: Proc. AAAI. (2018)
10. Park, W., Kim, D., Lu, Y., Cho, M.: Relational knowledge distillation. In: Proc. CVPR. (2019)
11. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: Beyond empirical risk minimization. In: Proc. ICLR. (2018)
12. Verma, V., Lamb, A., Beckham, C., Najafi, A., Mitliagkas, I., Lopez-Paz, D., Bengio, Y.: Manifold mixup: Better representations by interpolating hidden states. In: Proc. ICML. (2019)
13. Wang, D., Li, Y., Wang, L., Gong, B.: Neural networks are more productive teachers than human raters: Active mixup for data-efficient knowledge distillation from a blackbox model. In: Proc. CVPR. (2020)
14. Revaud, J., Almazán, J., Rezende, R.S., Souza, C.R.d.: Learning with average precision: Training image retrieval with a listwise loss. In: Proc. ICCV. (2019)
15. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International journal of computer vision (2004)
16. Sivic, Zisserman: Video google: a text retrieval approach to object matching in videos. In: Proc. ICCV. (2003)
17. Sánchez, J., Perronnin, F., Mensink, T., Verbeek, J.: Image classification with the fisher vector: Theory and practice. International journal of computer vision (2013) 222–245
18. Arandjelovic, R., Zisserman, A.: All about vlad. In: Proc. CVPR. (2013)
19. Perd'och, M., Chum, O., Matas, J.: Efficient representation of local geometry for large scale object retrieval. In: Proc. CVPR. (2009)
20. Sharif Razavian, A., Azizpour, H., Sullivan, J., Carlsson, S.: Cnn features off-the-shelf: an astounding baseline for recognition. In: Proc. CVPRW. (2014)
21. Babenko, A., Slesarev, A., Chigorin, A., Lempitsky, V.: Neural codes for image retrieval. In: Proc. ECCV. (2014)
22. Arandjelovic, R., Gronat, P., Torii, A., Pajdla, T., Sivic, J.: Netvlad: Cnn architecture for weakly supervised place recognition. In: Proc. CVPR. (2016)
23. Radenović, F., Iscen, A., Tolias, G., Avrithis, Y., Chum, O.: Revisiting oxford and paris: Large-scale image retrieval benchmarking. In: Proc. CVPR. (2018)
24. Schonberger, J.L., Radenovic, F., Chum, O., Frahm, J.M.: From single image query to detailed 3d reconstruction. In: Proc. CVPR. (2015)
25. Perronnin, F., Larlus, D.: Fisher vectors meet neural networks: A hybrid classification architecture. In: Proc. CVPR. (2015)

26. Tolias, G., Sicre, R., Jegou, H.: Particular object retrieval with integral max-pooling of cnn activations. arXiv preprint arXiv:1511.05879 (2015)
27. Gong, Y., Wang, L., Guo, R., Lazebnik, S.: Multi-scale orderless pooling of deep convolutional activation features. In: Proc. ECCV. (2014)
28. Oh Song, H., Xiang, Y., Jegelka, S., Savarese, S.: Deep metric learning via lifted structured feature embedding. In: Proc. CVPR. (2016)
29. Chen, W., Chen, X., Zhang, J., Huang, K.: Beyond triplet loss: a deep quadruplet network for person re-identification. In: Proc. CVPR. (2017)
30. Breiman, L., Shang, N.: Born again trees. In: Citeseer. (1996)
31. Zagoruyko, S., Komodakis, N.: Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. arXiv preprint arXiv:1612.03928 (2016)
32. Huang, Z., Wang, N.: Like what you like: Knowledge distill via neuron selectivity transfer. arXiv preprint arXiv:1707.01219 (2017)
33. Bagherinezhad, H., Horton, M., Rastegari, M., Farhadi, A.: Label refinery: Improving imagenet classification through label progression. arXiv preprint arXiv:1805.02641 (2018)
34. Furlanello, T., Lipton, Z.C., Tschannen, M., Itti, L., Anandkumar, A.: Born again neural networks. arXiv preprint arXiv:1805.04770 (2018)
35. Tokozume, Y., Ushiku, Y., Harada, T.: Between-class learning for image classification. In: Proc. CVPR. (2018)
36. Cho, K., et al.: Retrieval-augmented convolutional neural networks against adversarial examples. In: Proc. CVPR. (2019)
37. Radenović, F., Tolias, G., Chum, O.: Fine-tuning cnn image retrieval with no human annotation. IEEE TPAMI (2018)
38. Verma, V., Lamb, A., Kannala, J., Bengio, Y., Lopez-Paz, D.: Interpolation consistency training for semi-supervised learning. In: Proc. IJCAI. (2019)
39. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Proc. NIPS. (2012)
40. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
41. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Lost in quantization: Improving particular object retrieval in large scale image databases. In: 2008 IEEE conference on computer vision and pattern recognition, IEEE (2008) 1–8
42. Radenovic, F., Schonberger, J.L., Ji, D., Frahm, J.M., Chum, O., Matas, J.: From dusk till dawn: Modeling in the dark. In: Proc. CVPR. (2016)
43. Izmailov, P., Podoprikhin, D., Garipov, T., Vetrov, D., Wilson, A.G.: Averaging weights leads to wider optima and better generalization. In: Proc. UAI. (2018)
44. Jégou, H., Chum, O.: Negative evidences and co-occurences in image retrieval: The benefit of pca and whitening. In: Proc. ECCV. (2012)