Large-to-small Image Resolution Asymmetry in Deep Metric Learning

Pavel Suma Giorgos Tolias
Visual Recognition Group, Faculty of Electrical Engineering, Czech Technical University in Prague
sumapave,toliageo@fel.cvut.cz

Abstract

Deep metric learning for vision is trained by optimizing a representation network to map (non-)matching image pairs to (non-)similar representations. During testing, which typically corresponds to image retrieval, both database and query examples are processed by the same network to obtain the representation used for similarity estimation and ranking. In this work, we explore an asymmetric setup by light-weight processing of the query at a small image resolution to enable fast representation extraction. The goal is to obtain a network for database examples that is trained to operate on large resolution images and benefits from fine-grained image details, and a second network for query examples that operates on small resolution images but preserves a representation space aligned with that of the database network. We achieve this with a distillation approach that transfers knowledge from a fixed teacher network to a student via a loss that operates per image and solely relies on coupled augmentations without the use of any labels. In contrast to prior work that explores such asymmetry from the point of view of different network architectures, this work uses the same architecture but modifies the image resolution. We conclude that resolution asymmetry is a better way to optimize the performance/efficiency trade-off than architecture asymmetry. Evaluation is performed on three standard deep metric learning benchmarks, namely CUB200, Cars196, and SOP. Code: `https://github.com/pavelsuma/raml`

1. Introduction

The performance of deep learning models typically increases with their size and computational complexity. Most work focuses on improving recognition performance and therefore relies on expensive models to train and deploy. Standard deep network architectures [21, 18] are available in different variants that cover a range of the trade-off between performance and efficiency. Optimizing this trade-off forms a particular line of research that attracts a lot of attention since efficient and lightweight deep models allow deployment on mobile and low-resource devices or enable real-time execution. Therefore, powerful yet efficient models are desirable. One of the standard practices is to initially train a large model that is then used to obtain a smaller one, with weight pruning [17] and network distillation [20] being two dominant approaches to achieve that.

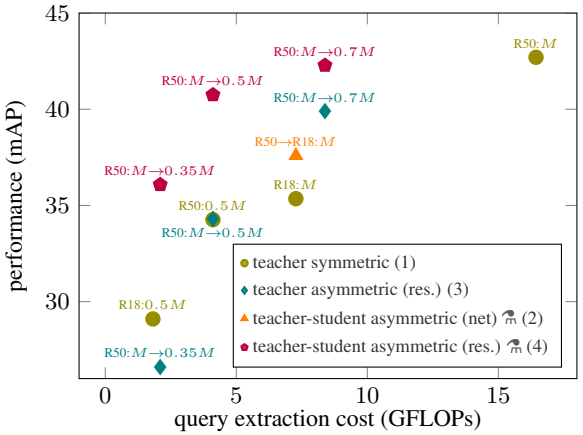

Figure 1. Retrieval performance (mAP) *vs.* extraction cost of the query representation (GFLOPs) for the CUB200 dataset. The notation format used is "database setup"→"query setup", where R50 and R18 are two variants of ResNet architecture. M is equal to 448 and indicates the width and height of images. Contrary to the standard symmetric retrieval (circle), the query in the asymmetric setting is processed by a lighter network (triangle) or in a smaller resolution (diamond, pentagon). ⚗: networks trained with the proposed distillation approach to achieve resolution asymmetry (the focus of this work) and also network asymmetry for comparison.

Network distillation uses the large model as a teacher that guides the training of a smaller student model. Most work in distillation is related to classification tasks, where the teacher logits act as a supervision [20]. Still, some work focuses on metric learning and image retrieval, where either the underlined vector representation [33] or pairwise scalar similarity values [30] function as supervision for distillation. In classification, once the small model is obtained, the large one is no longer used. However, due to the pairwise nature of the retrieval task, a possible asymmetry emerges; the query and database examples are processed by two different networks, with the network of the

former being lightweight to reduce the extraction cost during query time. For small to medium size databases or with the use of fast nearest neighbor search methods [24, 2, 25], the extraction cost of the representation can be the test-time bottleneck. In the asymmetric setup, the two representation spaces corresponding to each of the two networks need to be aligned and compatible. This is the objective of asymmetric metric learning (AML) as introduced by Budnik and Avrithis [6].

AML is studied under the lens of asymmetric network architectures; the student model is a pruned variant of the teacher, a different but lighter architecture possibly discovered by neural architecture search. All these aspects reduce the query time. The resolution of the input images is an important aspect that is overlooked. The use of fully convolutional architectures allows any input resolution, while the representation extraction cost is roughly quadratic in that matter. Metric learning tasks that focus on instance-level recognition are known to benefit from the use of a large image resolution [31, 4]. The same holds for fine-grained recognition where object details matter [35], as also shown in this work. Therefore, input resolution is a critical parameter for the performance/efficiency trade-off.

This work focuses on AML, where the asymmetry comes at the level of input resolution between the database network and the query network. The two architectures are the same, but the query network is trained at a low resolution to match the representation of the database network at a high resolution, which is performed with a distillation process that transfers knowledge from a teacher (database network) to the student (query network). The contributions of this work are summarized as follows:

- Asymmetry in the form of input image resolution is explored for the first time in deep metric learning.
- A distillation approach is proposed to align the representation (absolute distillation) and the pairwise similarities (relational distillation) between student and teacher across task-tailored augmentations of the same image.
- We conclude that resolution asymmetry is a better way to optimize the performance *vs*. efficiency trade-off compared to network asymmetry.
- As a side-effect, the student obtained with distillation noticeably outperforms the deep metric learning baselines in a conventional/symmetric retrieval.

A performance *vs*. efficiency comparison is shown in Figure 1. Compared to the baselines where a single network extracts both database and query examples with the same resolution (circle) and different resolution (diamond), distillation performs much better. The superiority of resolution over network asymmetry is evident too. More details about these experiments are discussed in Section 4.

2. Related work

Asymmetric embedding compatibility. In image retrieval, embedding compatibility has to be ensured when the database examples are processed by a different network than the query examples. To this end, AML [6] redefines standard metric learning losses in an asymmetric way, *i.e*. the anchor example is processed by the query network, while the database network processes the corresponding positive and negative examples. However, for the objective of representation space alignment, these losses are outperformed by a simple unsupervised regression loss on the embeddings, a form of knowledge distillation between teacher and student. Using the supervised losses, on the other hand, boosts the performance of symmetric retrieval with the student, which even surpasses its teacher. Following the paradigm of unsupervised distillation, another recent approach compels the student to mimic the contextual similarities of image neighbors in the teacher embedding space [50]. Other than optimizing the weights of the student network, a generalization includes using neural architecture search to additionally optimize the network architecture [11] in a work that focuses on training for classification rather than in a metric learning manner.

Classification-based training is the dominant approach in a relevant task to ours, called backward compatible learning (BCT) [36]. However, the underlined task assumptions are different. Its objective is to add new data processed with a stronger backbone version without back-filling the current database. Compatibility is established with cross-entropy loss of the old classifier on old and new embeddings of the same input image. This is extended to the compatibility of multiple embedding versions [23] or to tackling open-set backward compatibility with a continual learning approach [42]. Similarly, forward compatible learning [32] stores side information during training, which is leveraged in the future to transfer the old embeddings to another task. Other methods for fixing inconsistent representation spaces include class prototypes alignment [3, 52], and transformation of both spaces, rather than a single one [43, 22].

Asymmetry also emerges when embeddings are collected from diverse devices which use different models, *e.g*. in the domain of faces where the recognition should be compatible with all models [8], or in localization and mapping task with multiple agents [12].

Distillation and small image resolution. Image downsampling remains the primary pre-processing step even at present times when the average visual memory of GPU allows processing bigger resolutions during training and testing. It is observed that using larger images reliably translates to higher performance regardless of the objective or dataset [35]. Yet there are still many valid use cases where the inference has to be done with limited resources. In this context, distillation is used to align embeddings of high- and

low-resolution images in the form of feature regression [15] or KL divergence [26].

Some networks are specifically trained to facilitate dynamic input resolution changes for an optimal speed-accuracy trade-off. Examples include distillation with mutual learning [51] or ensembling with the teacher learned on the fly [47]. Distillation is also popular in the reverse direction going from small to large resolution, such as in the area of single image super-resolution [54, 14, 19, 28].

3. Method

Let $\mathcal{X}$ be the space of all images, and $s : \mathcal{X} \times \mathcal{X} \rightarrow [-1, 1]$ be a similarity function that estimates scalar similarity $s(x, q)$ for images $x, q \in \mathcal{X}$, also called examples in the following.

During testing, *i.e*. for image retrieval, a similarity is estimated between the query example q and each example in the database. Retrieval is then performed by ranking similarities in descending order. In order to perform the retrieval efficiently, representation function $f : \mathcal{X} \rightarrow \mathbb{R}^d$ is used that maps input examples from $\mathcal{X}$ to a d-dimensional representation vector. These real-valued vectors, referred to as embeddings, are ℓ_2-normalized and are used in combination with a standard similarity measure in $\mathbb{R}^d$ to obtain $s(\cdot)$. In this work, function $f(\cdot)$ is assumed to be implemented by a fully convolutional network (FCN), also denoted by $f_\theta(\cdot)$ declaring that the deep network is parametrized by parameter set θ. As a result of using an FCN, the input image can be of any resolution.

3.1. Symmetric retrieval

In the conventional task of symmetric retrieval, similarity $s(x, q)$ is computed by a simple dot product (equivalent to cosine similarity due to ℓ_2-normalized embeddings) given by

$$s(x, q) := s_s(x, q, f_\theta) = f_\theta(x)^T f_\theta(q), \qquad (1)$$

where representation function $f_\theta(\cdot)$ is used to process both the query and database examples in the same, symmetric, way, *i.e*. same network architecture with the same parameters. Weights θ are optimized during the training phase according to semantic labels of pairs so that matching (non-matching) examples are mapped to nearby (faraway) embeddings in the representation space. This kind of training is the so called deep metric learning. Typical examples of losses are contrastive [16], triplet [45], and multi-similarity [46] that involve optimization of $s_s(\cdot)$ for training pairs[1]. A network trained in such symmetric way constitutes the teacher in the following of this manuscript. Nonetheless, the teacher can equivalently be trained with other deep metric losses that do not directly involve pairwise comparison of train examples [40, 13]. Both the testing and the training of the teacher are performed with the use of symmetric similarity $s_s(\cdot)$.

[1]Some of the standard losses involve the use of Euclidean distance, but an alternative formulation with the use of similarity is possible. In this work, we employ similarities rather than distances.

3.2. Asymmetric retrieval - network-wise

This subsection provides background on the prior work which we rely on. Asymmetric similarity is defined as

$$s(x, q) := s_{an}(x, q, f_\theta, g_\phi) = f_\theta(x)^T g_\phi(q), \qquad (2)$$

where $g_\phi : \mathcal{X} \rightarrow \mathbb{R}^d$ denotes a second FCN with parameter set ϕ to process the query only. The asymmetry is with respect to the network architectures processing query and database examples. The architecture of the query network $g(\cdot)$ is meant to be lighter than that of the database network $f(\cdot)$ to make the asymmetry meaningful in terms of query speed up. Following prior work [6], we assume that the database embeddings are fixed and extracted with $f_\theta(\cdot)$ which is trained by optimizing symmetric similarity; there is no option for modifying the database network or the database embeddings.

Budnik and Avrithis [6] perform a distillation process where knowledge from the fixed teacher (database) network is transferred to the student (query) network. This is performed by a regression process, also called absolute distillation, where the student embedding is optimized to match the teacher embedding for a particular training image x such as by minimizing loss function $(1 - s_{an}(x, x, f_\theta, g_\phi))^2$. Their work concludes that this simple distillation process without the use of labels is the best performing approach for asymmetric retrieval as it directly reflects the objective of the task, *i.e*. to align the two representation spaces. Combination with losses that use semantic labels performs worse for asymmetric retrieval as it compromises the alignment process.

3.3. Asymmetric retrieval - resolution-wise

In this subsection, we first highlight the impact of image resolution for symmetric similarity and then introduce resolution asymmetry.

Even though the input resolution can be arbitrary, the level of details seen during training influences the ability to capture information in the representation. In practice, we make the following two observations: (i) the resolution used in training imposes restrictions on the resolution used during test time [41]; we presume the network gets adjusted to a specific level of details and scale of objects or their parts. (ii) fine-grained recognition, which is the main focus of this work, benefits from larger resolutions than the ones typically used for image classification tasks. Results in Table 1 support these observations; performance declines

CUB200

train resolution \ test resolution	634	448	317	224	158	112
634	43.9	42.2	37.3	30.3	22.7	15.3
448	42.0	42.7	39.2	32.8	25.0	17.5
317	35.9	40.7	39.6	34.1	26.3	18.3
224	21.6	31.6	36.3	34.3	27.9	20.7
158	11.4	19.1	27.5	31.7	28.3	21.9
112	10.9	16.8	22.9	28.0	26.8	21.9

Cars196

train resolution \ test resolution	634	448	317	224	158	112
634	40.4	35.5	26.7	16.7	9.3	5.0
448	42.3	41.6	34.5	23.0	13.0	6.3
317	31.9	36.2	33.8	25.2	15.4	7.6
224	22.8	31.3	33.3	29.3	20.7	10.9
158	12.7	21.0	27.3	29.1	25.2	15.4
112	6.0	10.4	16.7	22.1	22.1	16.6

SOP

train resolution \ test resolution	634	448	317	224	158	112
634	60.7	58.1	52.9	45.1	37.4	30.0
448	61.0	61.6	57.5	50.8	42.4	33.9
317	59.1	59.4	60.4	51.8	43.7	34.1
224	55.9	60.0	60.0	57.4	50.8	41.0
158	51.3	55.2	55.9	55.2	53.2	41.5
112	45.8	49.6	51.8	51.9	48.7	47.2

Table 1. Retrieval performance with the use of symmetric similarity when the network is trained and tested on different image resolution. This experiment does not include any asymmetry between database and query. ResNet-50 is used as the backbone and trained with labels and triplet loss (equivalent to the teacher network in our approach). Mean Average Precision is reported.

when there is a test-train resolution discrepancy. An exception appears for small training resolution where asymmetry with a slightly larger test resolution performs better; the benefit of the larger train image resolution is higher than the harm of the asymmetry. As shown in the work of Berman *et al*. [4], the mentioned discrepancy can be alleviated by re-parametrization of a non-linear pooling operation, but this does not apply to the asymmetric task of this work.

The asymmetry in Section 3.2 is caused by using two different network architectures to process query and database examples. This work introduces an asymmetry in the resolution of each model's input images. In that case, the asymmetric similarity is firstly defined in a simplistic way given by

$$s(x, q) := s_{ar}(x, q, f_\theta) = f_\theta(x)^T f_\theta(r(q)), \qquad (3)$$

where $r : \mathcal{X} \to \mathcal{X}$ is an image down-sampling function. In the case of network asymmetry, two different networks cannot be used without any training for alignment of the representation spaces. Conversely, in the case of resolution asymmetry, the same network with the same parameters can be used, as in (3), at different resolutions. Therefore, embeddings of two different resolutions are matched to each other despite the discrepancy in the average object scale, which is expected to harm performance compared to the symmetric case.

We go one step further from (3) and re-parametrize the query network and define asymmetric similarity by

$$s(x, q) := s_{ar}(x, q, f_\theta, f_\phi) = f_\theta(x)^T f_\phi(r(q)), \qquad (4)$$

where the architecture for the database and query networks are identical, but their parameters differ. In the following, we discuss how to optimize ϕ for resolution-wise asymmetric similarity.

Training. The teacher model f_θ is given and is pre-trained on the training examples at the large resolution. Teacher parameters are frozen for the entirety of our training. We initialize the student f_ϕ with the parameters θ of the teacher; this is a good initialization aligning the representation spaces up to the resolution discrepancy. Note that there is no such good initialization for network-wise asymmetry. We use two types of losses, one to perform absolute distillation and one to perform relational distillation [34, 30]. A visual overview of the proposed method is given in Figure 2. We also refer to teacher and student networks by $T(\cdot)$ and $S(\cdot)$ for brevity, where the latter already includes the down-sampling process.

Absolute distillation is performed, as in the original AML work [6], by regression between the teacher and student outputs given by loss

$$\ell_{abs}(x, \theta; \phi) = \Big(1 - s_{ar}(x, x, f_\theta, f_\phi)\Big)^2. \qquad (5)$$

Note that ℓ_{abs} is bounded between 0 and 4. This loss does not involve any labels and is applicable to any example in the training set, allowing for alignment of the representation space at a large number of training embeddings.

Coupled augmentations. A common trick to introduce curated noise is to randomly alter each network input by handcrafted, domain-specific functions, *i.e*. random image augmentations. During supervised training, the model learns to be invariant to such noise since the alterations of a single image correspond to the same label. Our proposed method is fully unsupervised but utilizes random augmentations to virtually increase the training set and the number of training embeddings used for alignment. We empirically observe that it is essential to randomly perturb the input in the same way for both teacher and student involved in the asymmetric similarity. These are what we refer to as coupled augmentations in this work. Knowledge distillation is known to be highly compatible and greatly benefit from data augmentations [44]; this is observed in our case too but only with coupled augmentations. Experiments and discussion for coupled and non-coupled augmentations are included in Section 4.

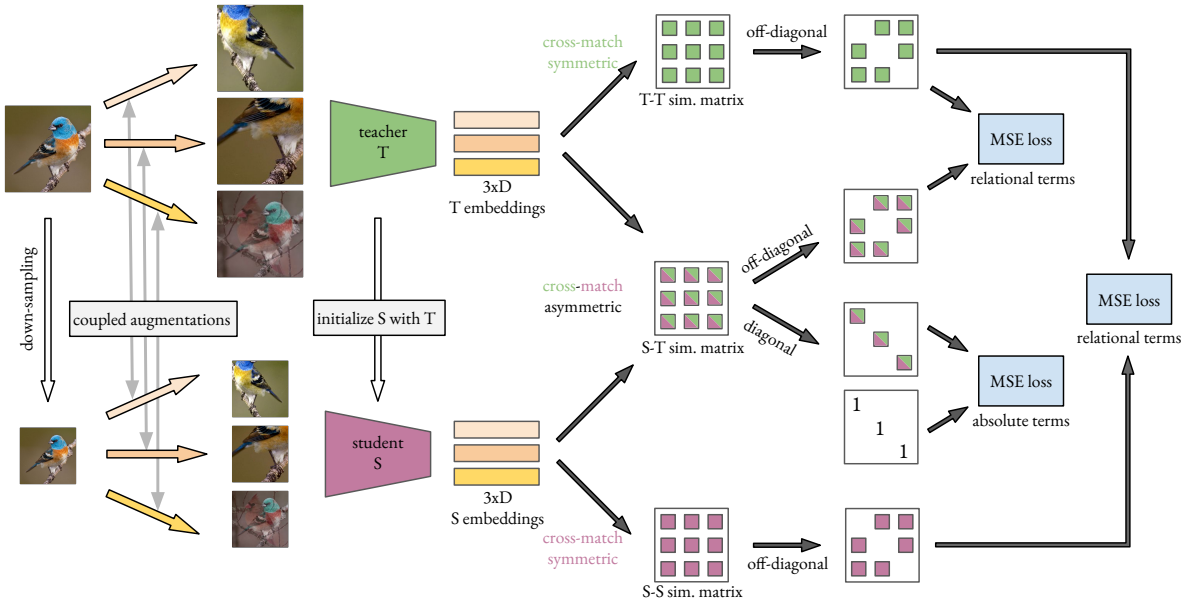

Figure 2. An overview of the proposed approach. The teacher network, which processes database images during testing, is pre-trained to operate on large image resolution and is now fixed and used to distill knowledge to the student network that operates on small image resolution which processes queries during testing. The similarity is computed between all embeddings of different augmentations of the same image with the same network and across networks. The resulting similarity matrices are used in three distillation losses during training. MSE: mean squared error.

More formally, we define $a(x)$ as the set of examples obtained by applying different image augmentations to x. Then, the average absolute distillation loss[2] is given by

$$\mathcal{L}_{abs}(x, a, \theta; \phi) = \frac{1}{|a(x)|} \sum_{z \in a(x)} \ell_{abs}(z, \theta; \phi). \quad (6)$$

Relational distillation is performed by using scalar similarities, instead of the embeddings, to guide the student. What is preserved is the relative comparison between two examples, therefore the name relational [30]. The two examples involved are different augmentations of the same original example, resulting in a loss being applied per original example separately. We propose two alternatives where the teacher-to-teacher relation is distilled to a teacher-to-student relation or student-to-student relation. The former is given by

$$\mathcal{L}_{rel-ts}(x, a, \theta; \phi) = \frac{1}{n} \sum_{\substack{y,z \in a(x) \\ z \neq y}} \ell_{rel-ts}(y, z, \theta; \phi), \quad (7)$$

where $n = |a(x)|^2 - |a(x)|$ and

$$\begin{aligned} \ell_{rel-ts}(y, z, \theta; \phi) &= \Big(s_s(y, z, f_\theta) - s_{ar}(y, z, f_\theta, f_\phi)\Big)^2 \\ &= \Big(f_\theta(y)^\top f_\theta(z) - f_\theta(y)^\top f_\phi(r(z))\Big)^2 \\ &= \Big(T(y)^\top T(z) - T(y)^\top S(z)\Big)^2. \end{aligned} \quad (8)$$

[2]In the mathematical formulation, we first augment and then down-sample, which is a different order than in Figure 2, but identical in practice.

Equivalently to $\mathcal{L}_{rel-ts}$ in (7), we use $\mathcal{L}_{rel-ss}$ by defining

$$\begin{aligned} \ell_{rel-ss}(y, z, \theta; \phi) &= \Big(s_s(y, z, f_\theta) - s_s(r(y), r(z), f_\phi)\Big)^2 \\ &= \Big(f_\theta(y)^\top f_\theta(z) - f_\phi(r(y))^\top f_\phi(r(z))\Big)^2 \\ &= \Big(T(y)^\top T(z) - S(y)^\top S(z)\Big)^2. \end{aligned} \quad (9)$$

Relational distillation in the form of (9) is shown to fail in prior work for asymmetric retrieval [6] as it does not satisfy the alignment objective, while in the form of (8) is shown effective [50]. In our case, such distillation loss terms are effective even by themselves under particular conditions, which is shown and discussed in Section 4. Note that one of our relational terms, *i.e.* (8) is similar to that in the work of Wu *et al*. [50]. Nevertheless, we do not require a costly nearest neighbors process and simply use random augmentations.

Each of the three losses is averaged over all examples in a batch, resulting to $\mathcal{L}_{abs}$, $\mathcal{L}_{rel-ts}$, and $\mathcal{L}_{res-ss}$, respectively. The the total loss is given by

$$\mathcal{L} = \mathcal{L}_{abs} + \lambda_t \mathcal{L}_{rel-ts} + \lambda_s \mathcal{L}_{rel-ss}, \quad (10)$$

where λ_t and λ_s are hyper-parameters tuned during cross-validation.

4. Experiments

In this section, we provide the implementation details and present the experimental results.

4.1. Datasets

We use three standard deep metric learning datasets for fine-grained recognition, namely Caltech-UCSD Birds (*CUB200*) dataset [48], Stanford Cars dataset (*Cars196*) [27], and Stanford Online Products (*SOP*) [38]. They comprise 11,788 images of birds from 200 classes, 16,185 images of cars from 196 classes, and 120,053 images of products from 22,634 classes, respectively. Following common practice, for CUB200 and Cars196 we use the first half of the classes for training and the other half for testing, while for SOP we use the provided train/test sets.

4.2. Experimental setup

As the first step, we use the labeled training set to train the teacher. All training images are re-sampled to have their largest dimension equal to what we refer to as database resolution or large resolution. Then, the teacher weights are used to initialize the student, while we freeze the teacher weights and optimize the student with the proposed distillation losses on the whole training set. During this process, the student receives input images at what we refer to as query resolution or small resolution. We perform the student training three times with different, but fixed, seeds. In the evaluation phase, performance is evaluated by mean Average Precision (mAP) and by recall at 1 (R@1) which is equal to 1 if the top retrieved image is from the correct class. We report average performance across seeds.

The student network is evaluated in two ways. Firstly, for asymmetric retrieval where the database (query) examples are processed by the teacher (student) network in the large (small) resolution. Secondly, we drop the assumption that the database embeddings are fixed and evaluate the student network in a symmetric manner where it processes both database and query examples in the small resolution. The teacher network, other than participating in the distillation process and in the aforementioned asymmetric testing for retrieval, is also evaluated in the following two ways to provide a baseline. Firstly, it is evaluated in a symmetric way by processing both database and query examples at the resolution it was trained with. Secondly, it is evaluated in an asymmetric way by processing database (query) examples at the large (small) resolution (a single network is processing images at two different resolutions). To summarize, our evaluation setups are *teacher-student asymmetric* (two networks, two resolutions, same architecture) by (4), *student symmetric* by (1), *teacher symmetric* by (1), *teacher asymmetric* (one network, two resolutions) by (3).

4.3. Implementation details

ResNet-50 (R50) and ResNet-18 (R18) [18] are used as backbone FCN, with initial weights obtained from ImageNet [10] pre-training. Generalized mean pooling (GeM) [31] and a fully connected layer reducing the final embedding dimension to 512 are added on top of the FCN. Optimization is performed with AdamW [29] and a one cycle learning rate scheduler [37] using PyTorch default values. Teacher networks are trained with triplet loss and distance weighed negative mining [49]. Following standard protocol [35, 5, 7], we perform random crop [39], resize to fixed resolution and flip horizontally with probability 0.5. The number of epochs is set to 200. Batch size equals 200 for CUB200 and Cars196 and 4000 for SOP.

During student training with distillation, the same augmentation strategy is used with the addition of color distortion with the strength of 0.5 [9] and image mixup [53] with $\alpha = 0.2$. For mixup, each image is mixed with the next image in the batch. We train for 200 epochs with batch size equal to 200, limit the number of examples to 8000 per epoch, and use 8 different augmentations per image unless otherwise stated. For proper hyper-parameter tuning, we use Optuna library [1] and half of the training set as validation set [40]. We tune learning rate, λ_t, and λ_s; indicative values are 1.1e-4, 0.7, and 0.7, respectively.

The evaluation policy is identical for all experiments. The input image is re-sampled to the large or small resolution depending on the setup and network used while preserving the aspect ratio. Following common practice, the center square area is cropped.

4.4. Results

Performance comparison at multiple query resolutions. In Table 2 we show the performance of our distillation method and compare it with baselines on three smaller image resolutions. Compared with the respective (same query resolution) baselines for symmetric retrieval (1st block), the distilled students (3rd block) are on all datasets higher in terms of their asymmetric retrieval performance. The benefit of asymmetry and using large resolution database images gets larger for decreasing query resolution (increasing query extraction savings). We also substantially improve over the naive asymmetric approach (2nd block *vs*. 3rd block), where the teacher network processes the small resolution queries without any appropriate training.

If the assumption of fixed database embeddings is dropped, it makes sense to look at the student symmetric retrieval performance at small resolutions; database extraction cost is also reduced now. First, we observe that through distillation, symmetric retrieval performance (4th block) is much higher than training with the standard deep metric learning way (1st block). Note that the same is achieved in earlier work on asymmetric metric learning, but only with the use of labels [6], while we do not use labels for student training. Secondly, we observe that the student's symmetric performance (measured via mAP) is larger than its asymmetric one if the query resolution

QR	DR	QN	DN	CUB200 mAP	CUB200 R@1	Cars196 mAP	Cars196 R@1	SOP mAP	SOP R@1
teacher symmetric: teacher trained & tested @ DR, different network per row									
448	448	T	T	42.7	73.7	41.6	87.9	62.6	82.1
317	317	T	T	39.6	71.6	33.8	81.4	60.4	81.2
224	224	T	T	34.3	64.6	29.6	75.2	57.4	79.2
158	158	T	T	28.3	55.9	25.2	64.1	53.2	76.0
teacher asymmetric: teacher trained @ 448, same network for all rows									
317	448	T	T	39.9	69.5	36.5	81.3	58.4	80.1
224	448	T	T	34.3	61.0	25.7	61.0	51.3	74.3
158	448	T	T	26.6	47.6	14.8	32.3	38.2	60.1
teacher-student asymmetric: teacher trained @ 448, student trained with distillation @ QR same teacher for all rows, different student per row									
317	448	S	T	42.3	72.9	41.0	87.1	61.3	81.9
224	448	S	T	40.8	70.5	38.1	83.4	59.7	80.8
158	448	S	T	36.9	63.7	31.8	71.3	56.4	78.1
student symmetric: student trained with distillation @ QR, different student per row									
317	317	S	S	43.6	74.9	41.8	88.1	61.6	82.0
224	224	S	S	40.9	71.8	38.1	85.0	59.5	80.8
158	158	S	S	36.0	66.8	30.6	76.3	55.3	77.9

Table 2. Performance results for resolution-wise asymmetric and symmetric retrieval at different query resolution. QR,DR: query and database resolution. QN,DN: query and database network. S, T: student and teacher.

is relatively large. This does not hold for smaller query resolutions where asymmetry is still more meaningful. Focusing on recall at 1, the student performs in most cases higher in the symmetric setting than in the asymmetric, indicating a different behavior for the top-ranked examples; the asymmetric setup works better considering all relevant examples but worse if we consider the most similar ones.

Ablation study. Our work heavily relies on the use of augmentations. We dissect their contribution in Table 3. They are organized into three groups: *coupled* applies the same image transformations in the teacher and student (see Figure 2), *geometric augmentations* correspond to random resized crop, color jitter, and horizontal flip, and *MX* constitutes image mixup. The setup with augmentations performed only separately in the teacher input and student input is clearly worse and even harms the alignment. Such non-coupled augmentations can potentially increase the invariance of the asymmetric similarity; its failure signifies that representation space alignment is the important objective and not invariance, which is inherited from the teacher anyway. Mixup by itself is not as good as the more standard augmentations but still improves the final method. It seems that non-standard augmentations are effective if applied in a coupled way.

We additionally show the performance of each loss term from (10) separately along with their combination. We confirm that absolute distillation by itself is already a strong performer for our asymmetric resolution setting. It outperforms the other two relative loss terms. Nevertheless, their combination is the top-performing approach. Note that the relational loss terms have negligible added cost in the training since all the embeddings are already obtained for the needs of the absolute distillation loss term. To our surprise, $\mathcal{L}_{rel-ss}$ by itself improves the asymmetric performance, although the individual loss serves only as a regular relational knowledge distillation previously shown to fail for asymmetric retrieval [6]. We investigate this in the next experiment.

loss	coupled	G	MX	*asymmetric* mAP	*asymmetric* R@1	*symmetric* mAP	*symmetric* R@1
$\mathcal{L}_{abs}$				37.7	66.1	34.8	65.1
$\mathcal{L}_{abs}$		✓		31.3	55.6	30.7	60.3
$\mathcal{L}_{abs}$			✓	20.9	40.1	29.8	60.0
$\mathcal{L}_{abs}$	✓	✓		40.1	69.8	40.6	71.5
$\mathcal{L}_{abs}$	✓		✓	39.5	68.5	38.8	68.9
$\mathcal{L}_{abs}$	✓	✓	✓	40.3	69.8	40.5	71.2
$\mathcal{L}_{rel-ts}$	✓	✓	✓	39.7	68.9	39.1	69.2
$\mathcal{L}_{rel-ss}$	✓	✓	✓	37.9	65.1	40.0	71.4
$\mathcal{L}$	✓	✓	✓	40.8	70.5	40.9	71.8

Table 3. Impact of the loss function and of different augmentation strategies on performance for teacher-student asymmetric and student symmetric retrieval after distillation. Results reported on CUB200. G: geometric augmentations. MX: mixup at the input level. Teacher (student) operates at 448 (224) resolution.

Impact of student initialization. If we initialize the student network with the teacher weights, an initial alignment of the two embedding spaces is provided, which plays a crucial role, as seen in Table 4. Note that no such initialization exists in the case of network-wise asymmetry. It is possible in our case because it corresponds to matching objects at different resolutions with the same network, which is a good starting point that is then improved via distillation. We compare such initialization to the one using weights from a teacher trained for the small resolution and the one with ImageNet weights. Both alternatives perform worse than the suggested and are on par with each other. Even using each of the losses separately performs much better given a good initialization. We suspect the reason is that the embedding spaces in these two cases are initially entirely misaligned.

Noticeably, the relational distillation with $\mathcal{L}_{rel-ss}$, which does not involve any asymmetric term, completely fails without this initialization but performs well with it. We presume that such a relational distillation only works if we are already close to a good solution. Otherwise, satisfying its objective does not meet the representation space alignment objective at all.

Impact of number of augmentations. The amount of terms in the relative distillation loss grows quadratically with the number of augmentations per image, while training

initialization	loss	*asymmetric* mAP	R@1	*symmetric* mAP	R@1
teacher@448	$\mathcal{L}$	40.8	70.5	40.9	71.8
teacher@224	$\mathcal{L}$	36.3	62.9	37.0	66.8
ImageNet	$\mathcal{L}$	36.3	62.4	36.9	67.3
teacher@224	$\mathcal{L}_{abs}$	35.7	61.6	35.9	65.5
teacher@224	$\mathcal{L}_{rel-ts}$	36.3	60.8	38.9	68.5
teacher@224	$\mathcal{L}_{rel-ss}$	1.3	1.1	39.6	69.8
teacher@448	$\mathcal{L}_{abs}$	40.3	69.8	40.5	71.2
teacher@448	$\mathcal{L}_{rel-ts}$	39.7	68.9	39.1	69.2
teacher@448	$\mathcal{L}_{rel-ss}$	37.9	65.1	40.0	71.4

Table 4. Impact of the student initialization during our distillation process with the different losses used all together or separately. Performance is evaluated on CUB200 for teacher-student asymmetric retrieval and for student symmetric retrieval. Initialization is performed by the teacher trained at the large or small resolution, or with ImageNet pre-trained weights. Teacher (student) operates at 448 (224) resolution.

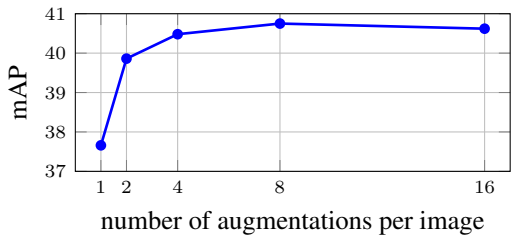

Figure 3. Impact of the number of augmentations on the performance of teacher-student asymmetric retrieval after our distillation on CUB200.

time increases more or less linearly. Figure 3 shows a noticeable increase in performance when using more than one augmentation of the same image. The gain saturates after 8, which is the number of augmentations we use in the rest of the experiments.

Performance *vs*. efficiency. We measure the query extraction cost with FLOPs for combinations of networks and specific query resolutions. The trade-off between performance and efficiency for Cars196 and SOP is summarized in Figure 4 (in Figure 1 for CUB200). For the case of teacher symmetric testing, the teacher is trained at the resolution it is tested on. In all other cases, the teacher is trained at resolution equal to 448. Our distillation approach is applicable in an off-the-shelf way to achieve network asymmetry too. We additionally perform distillation for network asymmetry using only (5) (no augmentations) to be as close as possible to AML [6] for reference. We observe that resolution asymmetry is outperforming network asymmetry for the same cost and that distillation noticeably outperforms the naive baseline that uses (3).

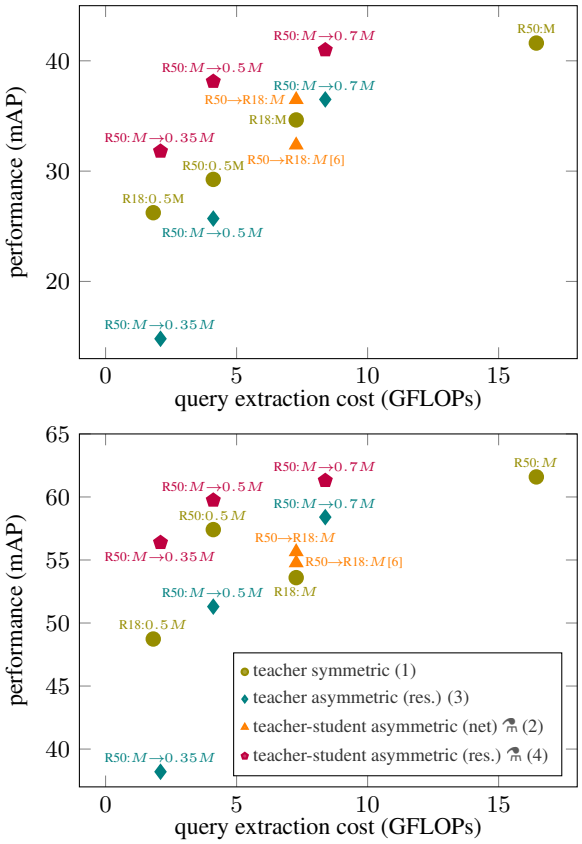

Figure 4. Retrieval performance (mAP) *vs*. extraction cost of the query representation (GFLOPs) for Cars196 (top) and SOP (bottom). The notation format used is "database setup"→"query setup", where R50 and R18 are two variants of ResNet architecture. M is equal to 448 and indicates the width and height of images. Contrary to the standard symmetric retrieval (circle), the query in the asymmetric setting is processed by a lighter network (triangle) or in a smaller resolution (diamond, pentagon). ⚗: networks trained with the proposed distillation approach to achieve resolution asymmetry (the focus of this work) and also network asymmetry for comparison.

5. Conclusions

In this work[3], we explore resolution asymmetry in deep metric learning and conclude that it forms a better way to optimize the performance *vs*. efficiency trade-off than network asymmetry studied in prior work. The proposed distillation approach performs well without the use of any labels and allows us to get useful insight into task-tailored augmentations, proper student initialization, and the importance of its different loss terms, namely absolute and relational. The combination of network and resolution asymmetry is theoretically feasible, possibly even in straightforward ways, but remains as future work. So does the case of dropping the fixed database embeddings assumption and jointly optimizing the database and query networks.

[3]Work supported by Junior Star GACR grant No. GM 21-28830M, and Grant Agency of the Czech Technical University in Prague grant No. SGS20/171/OHK3/3T/13.

References

[1] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *ACM SIGKDD*, 2019.

[2] Artem Babenko and Victor Lempitsky. Efficient indexing of billion-scale datasets of deep descriptors. In *CVPR*, 2016.

[3] Yan Bai, Jile Jiao, Shengsen Wu, Yihang Lou, Jun Liu, Xuetao Feng, and Ling-Yu Duan. Dual-tuning: Joint prototype transfer and structure regularization for compatible feature learning. In *arXiv*, 2021.

[4] Maxim Berman, Hervé Jégou, Vedaldi Andrea, Iasonas Kokkinos, and Matthijs Douze. MultiGrain: a unified image embedding for classes and instances. In *arXiv*, 2019.

[5] Andrew Brown, Weidi Xie, Vicky Kalogeiton, and Andrew Zisserman. Smooth-ap: Smoothing the path towards large-scale image retrieval. In *ECCV*, 2020.

[6] Mateusz Budnik and Yannis Avrithis. Asymmetric metric learning for knowledge transfer. In *CVPR*, 2021.

[7] Fatih Cakir, Kun He, Xide Xia, Brian Kulis, and Stan Sclaroff. Deep metric learning to rank. In *CVPR*, 2019.

[8] Ken Chen, Yichao Wu, Haoyu Qin, Ding Liang, Xuebo Liu, and Junjie Yan. R3 adversarial network for cross model face recognition. In *CVPR*, 2019.

[9] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020.

[10] Wei Dong, Richard Socher, Li Li-Jia, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009.

[11] Rahul Duggal, Hao Zhou, Shuo Yang, Yuanjun Xiong, Wei Xia, Zhuowen Tu, and Stefano Soatto. Compatibility-aware heterogeneous visual search. In *CVPR*, 2021.

[12] Mihai Dusmanu, Ondrej Miksik, Johannes L Schönberger, and Marc Pollefeys. Cross-descriptor visual localization and mapping. In *ICCV*, 2021.

[13] Ismail Elezi, Sebastiano Vascon, Alessandro Torcinovich, Marcello Pelillo, and Laura Leal-Taixé. The group loss for deep metric learning. In *ECCV*, 2020.

[14] Qinquan Gao, Yan Zhao, Gen Li, and Tong Tong. Image super-resolution using knowledge distillation. In *ACCV*, 2019.

[15] Shiming Ge, Shengwei Zhao, Chenyu Li, Yu Zhang, and Jia Li. Efficient low-resolution face recognition via bridge distillation. IEEE *Transactions on Image Processing*, 29:6898–6908, 2020.

[16] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *CVPR*, 2006.

[17] Song Han, Jeff Pool, John Tran, and William J. Dally. Learning both weights and connections for efficient neural networks. In *NeurIPS*, 2015.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[19] Zibin He, Tao Dai, Jian Lu, Yong Jiang, and Shu-Tao Xia. Fakd: Feature-affinity based knowledge distillation for efficient image super-resolution. In *ICIP*, 2020.

[20] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. In *arXiv*, 2015.

[21] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, Quoc V. Le, and Hartwig Adam. Searching for mobilenetv3. In *ICCV*, 2019.

[22] Jie Hu, Rongrong Ji, Hong Liu, Shengchuan Zhang, Cheng Deng, and Qi Tian. Towards visual feature translation. In *CVPR*, 2019.

[23] Weihua Hu, Rajas Bansal, Kaidi Cao, Nikhil Rao, Karthik Subbian, and Jure Leskovec. Learning backward compatible embeddings. In *arXiv*, 2022.

[24] Hervé Jégou, Matthijs Douze, and Cordelia Schmid. Product quantization for nearest neighbor search. *PAMI*, 33(1):117–128, Jan. 2011.

[25] Yannis Kalantidis and Yannis Avrithis. Locally optimized product quantization for approximate nearest neighbor search. In *CVPR*, 2014.

[26] Syed Safwan Khalid, Muhammad Awais, Zhen-Hua Feng, Chi-Ho Chan, Ammarah Farooq, Ali Akbari, and Josef Kittler. Resolution invariant face recognition using a distillation approach. IEEE *Transactions on Biometrics, Behavior, and Identity Science*, 2(4):410–420, 2020.

[27] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *ICCVW*, 2013.

[28] Wonkyung Lee, Junghyup Lee, Dohyung Kim, and Bumsub Ham. Learning with privileged information for efficient image super-resolution. In *ECCV*, 2020.

[29] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.

[30] Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *CVPR*, 2019.

[31] Filip Radenović, Giorgos Tolias, and Ondřej Chum. Fine-tuning cnn image retrieval with no human annotation. *PAMI*, 41(7):1655–1668, 2019.

[32] Vivek Ramanujan, Pavan Kumar Anasosalu Vasu, Ali Farhadi, Oncel Tuzel, and Hadi Pouransari. Forward compatible training for large-scale embedding retrieval systems. In *CVPR*, 2022.

[33] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. In *ICLR*, 2015.

[34] Karsten Roth, Timo Milbich, Bjorn Ommer, Joseph Paul Cohen, and Marzyeh Ghassemi. Simultaneous similarity-based self-distillation for deep metric learning. In *ICML*, 2021.

[35] Karsten Roth, Timo Milbich, Samarth Sinha, Prateek Gupta, Bjorn Ommer, and Joseph Paul Cohen. Revisiting training strategies and generalization performance in deep metric learning. In *ICML*, 2020.

[36] Yantao Shen, Yuanjun Xiong, Wei Xia, and Stefano Soatto. Towards backward-compatible representation learning. In *CVPR*, 2020.

[37] Leslie N. Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates. In *arXiv*, 2017.

[38] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding. In *arXiv*, 2015.

[39] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, 2015.

[40] Eu Wern Teh, Terrance DeVries, and Graham W Taylor. Proxynca++: Revisiting and revitalizing proxy neighborhood component analysis. In *ECCV*, 2020.

[41] Hugo Touvron, Andrea Vedaldi, Matthijs Douze, and Herve Jegou. Fixing the train-test resolution discrepancy. In *NeurIPS*, 2019.

[42] Timmy S. T. Wan, Jun-Cheng Chen, Tzer-Yi Wu, and Chu-Song Chen. Continual learning for visual search with backward consistent feature embedding. In *CVPR*, 2022.

[43] Chi Wang, Ya-Liang Chang, Shang-Ta Yang, Dong Chen, and Shang-Hong Lai. Unified representation learning for cross model compatibility. In *BMVC*, 2020.

[44] Huan Wang, Suhas Lohit, Michael Jones, and Yun Fu. Knowledge distillation thrives on data augmentation. In *arXiv*, 2020.

[45] Jiang Wang, Yang Song, Thomas Leung, Chuck Rosenberg, Jingbin Wang, James Philbin, Bo Chen, and Ying Wu. Learning fine-grained image similarity with deep ranking. In *CVPR*, 2014.

[46] Xun Wang, Xintong Han, Weilin Huang, Dengke Dong, and Matthew R Scott. Multi-similarity loss with general pair weighting for deep metric learning. In *CVPR*, 2019.

[47] Yikai Wang, Fuchun Sun, Duo Li, and Anbang Yao. Resolution switchable networks for runtime efficient image recognition. In *arXiv*, 2020.

[48] Peter Welinder, Steve Branson, Takeshi Mita, Catherine Wah, Florian Schroff, Serge Belongie, and Pietro Perona. Caltech-ucsd birds 200. Technical report, California Institute of Technology, 2010.

[49] Chao-Yuan Wu, R Manmatha, Alexander J Smola, and Philipp Krahenbuhl. Sampling matters in deep embedding learning. In *ICCV*, 2017.

[50] Hui Wu, Min Wang, Wengang Zhou, Houqiang Li, and Qi Tian. Contextual similarity distillation for asymmetric image retrieval. In *CVPR*, 2022.

[51] Taojiannan Yang, Sijie Zhu, Chen Chen, Shen Yan, Mi Zhang, and Andrew Willis. Mutualnet: Adaptive convnet via mutual learning from network width and resolution. In *ECCV*, 2020.

[52] Binjie Zhang, Yixiao Ge, Yantao Shen, Shupeng Su, Fanzi Wu, Chun Yuan, Xuyuan Xu, Yexin Wang, and Ying Shan. Towards universal backward-compatible representation learning. In *arXiv*, 2022.

[53] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICML*, 2018.

[54] Yiman Zhang, Hanting Chen, Xinghao Chen, Yiping Deng, Chunjing Xu, and Yunhe Wang. Data-free knowledge distillation for image super-resolution. In *CVPR*, 2021.