The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

DarkRank: Accelerating Deep Metric Learning via Cross Sample Similarities Transfer

Yuntao Chen,[1,5] **Naiyan Wang,**[2] **Zhaoxiang Zhang**[1,3,4,5,*]
[1]Research Center for Brain-inspired Intelligence, CASIA [2]TuSimple
[3]National Laboratory of Pattern Recognition, CASIA
[4]Center for Excellence in Brain Science and Intelligence Technology, CAS
[5]University of Chinese Academy of Sciences
{chenyuntao2016, zhaoxiang.zhang}@ia.ac.cn winsty@gmail.com

Abstract

We have witnessed rapid evolution of deep neural network architecture design in the past years. These latest progresses greatly facilitate the developments in various areas such as computer vision and natural language processing. However, along with the extraordinary performance, these state-of-the-art models also bring in expensive computational cost. Directly deploying these models into applications with real-time requirement is still infeasible. Recently, Hinton *et al.* (**?**) have shown that the dark knowledge within a powerful teacher model can significantly help the training of a smaller and faster student network. These knowledge are vastly beneficial to improve the generalization ability of the student model. Inspired by their work, we introduce a new type of knowledge – cross sample similarities for model compression and acceleration. This knowledge can be naturally derived from deep metric learning model. To transfer them, we bring the "learning to rank" technique into deep metric learning formulation. We test our proposed DarkRank method on various metric learning tasks including pedestrian re-identification, image retrieval and image clustering. The results are quite encouraging. Our method can improve over the baseline method by a large margin. Moreover, it is fully compatible with other existing methods. When combined, the performance can be further boosted.

Introduction

Metric learning is the basis for many computer vision tasks, including face verification(**?**; **?**) and pedestrian re-identification(**?**; **?**). In recent years, end-to-end deep metric learning method which learns feature representation by the guide of metric based losses has achieved great success(**?**; **?**; **?**). A key factor for the success of these deep metric learning methods is the powerful network architectures(**?**; **?**; **?**). Nevertheless, along with more powerful features, these deeper and wider networks also bring in heavier computation burden. In many real-world applications like autonomous driving, the system is latency critical with limited hardware resources. To ensure safety, it requires (more than) real-time responses. This constraint prevents us from benefiting from the latest developments in network design.

To mitigate this problem, many model acceleration methods have been proposed. They can be roughly categorized into three types: network pruning(**?**; **?**), model quantization(**?**; **?**) and knowledge transfer(**?**; **?**; **?**). Network pruning iteratively removes the neurons or weights that are less important to the final prediction; model quantization decreases the representation precision of weights and activations in a network, and thus increases computation throughput; knowledge transfer directly trains a smaller student network guided by a larger and more powerful teacher. Among these methods, knowledge transfer based methods are the most practical. Compared with other methods that mostly need tailor made hardwares or implementations, they can archive considerable acceleration without bells and whistles.

Knowledge Distill (KD)(**?**) and its variants(**?**; **?**) are the dominant approaches among knowledge transfer based methods. Though they utilize different forms of knowledges, these knowledges are still limited within a single sample. Namely, these methods provide more precise supervision for each sample from teacher networks at either classifier or intermediate feature level. However, all these methods miss another valuable treasure – the relationships (similarities or distances) across different samples. This kind of knowledge also encodes the structure of the embedded space of teacher networks. Moreover, it naturally fits the objective of metric learning since it usually utilizes similar instance level supervision. We elaborate our motivation in the sequel, and depict our method in Fig. **??**. The upper right corner shows that the student better captures the similarity of images after transferring. The digit 0 which are more similar to 6 than 3, 4, 5 are now ranked higher.

To summarize, the contributions of this paper are three folds:

- We introduce a new type of knowledge – cross sample similarities for knowledge transfer in deep metric learning.
- We formalize it as a rank matching problem between teacher and student networks, and modify classical list-wise learning to rank methods(**?**; **?**) to solve it.
- We test our proposed method on various metric learning tasks. Our method can significantly improve the performance of student networks. And it can be applied jointly with existing methods for a better transferring performance.

*corresponding author

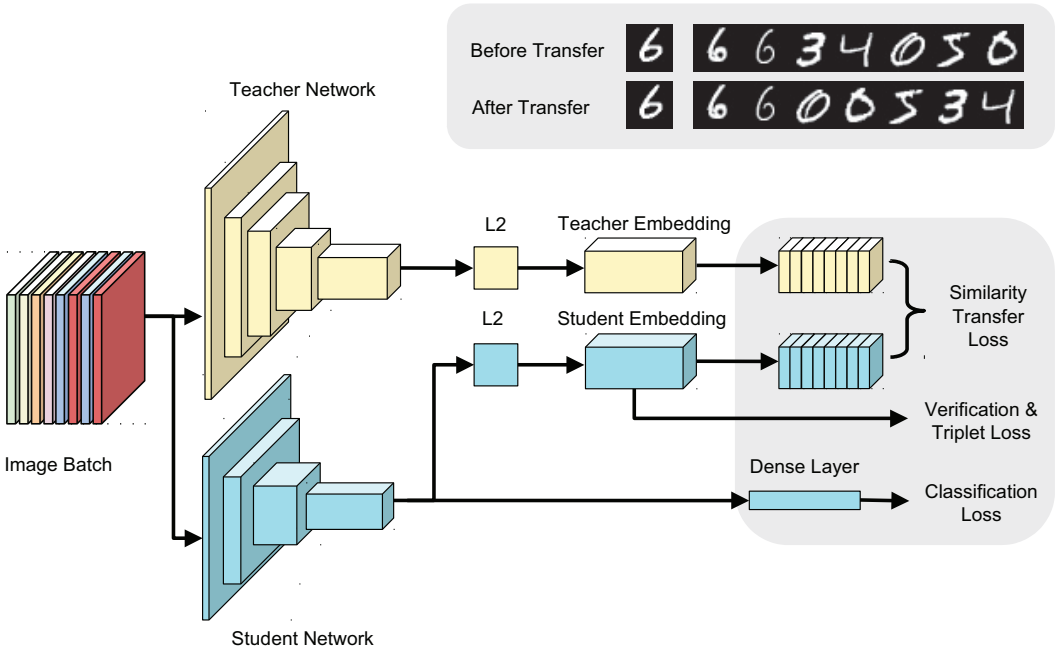


Figure 1: The network architecture of our DarkRank method. The student network is trained with standard classification loss, contrastive loss and triplet loss as well as the similarity transfer loss proposed by us.

Related works

In this section, we review several previous works that are closely related to our proposed method.

Deep Metric Learning

Different from most traditional metric learning methods that focus on learning a Mahalanobis distance in Euclidean space(**?**; **?**) or high dimensional kernel space(**?**), deep metric learning usually transforms the raw features via DNNs, and then compare the samples in Euclidean space directly.

Despite the rapid evolution of network architectures, the loss functions for metric learning are still a popular research topic. The key point of metric learning is to separate inter-class embeddings and reduce the intra-class variance. Classification loss and its variants(**?**; **?**) can learn robust features that help to separate samples in different classes. However, for out-of-sample identities, the performance cannot be guaranteed since no explicit metric is induced by this approach. Another drawback of classification loss is that it projects all samples with the same label to the same direction in the embedding space, and thus ignores the intra-class variance. Verification loss(**?**) is a popular alternative because it directly encodes both the similarity ans dissimilarity supervisions. The weakness of verification loss is that it tries to enforce a hard margin between the anchor and negative samples. This restriction is too strict since images of different categories may look very similar to each other. Imposing a hard margin on those samples only hurts the learnt representation. Triplet loss and its variants(**?**; **?**) overcome this disadvantage by imposing an order on the embedded triplets instead. Triplet loss is the exact reflection of desired retrieval results: the positive samples are closer to anchor than the negative ones. But its good performance requires a careful design of the sampling and the training procedure(**?**; **?**). Other related work includes center loss (**?**) which maintains a shifting template for each class to reduce the intra-class variance by simultaneously drawing the template and the sample towards each other. Besides loss function design, Bai *et al.* (**?**) introduce smoothness of metric space with respect to data manifold as a prior.

Knowledge Transfer for Model Acceleration and Compression

In (**?**), Bucila *et al.* first proposed to approximate an ensemble of classifiers with a single neural network. Recently, Hinton *et al.* revived this idea under the name knowledge distill(**?**). The insight comes from that the softened probabilities output by classifiers encode more accurate embedding of each sample in the label space than one-hot labels. Consequently, in addition to the original training targets, they proposed to use soft targets from teacher networks to guide the training of student networks. Through this process, KD transfers more precise supervision signal to student networks, and therefore improves their generalization ability. Subsequent works FitNets(**?**), Attention Transfer(**?**) and Neuron Selectivity Transfer(**?**) tried to exploit other knowledges in intermediate feature maps of CNNs to improve the performance. Instead of using forward input-output pairs, Czarnecki *et al.* tried to utilize the gradients with respect to input of teacher network for knowledge transfer with Sobolev training(**?**). In this paper, we exploit a unique type of knowledge inside deep metric learning model – cross sample similarities to train a better student network.

Learning to Rank

Learning to rank refers to the problem that given a query, rank a list of samples according to their similarities. Most learning to rank methods can be divided into three types: pointwise, pairwise and listwise, according to the way of assembling samples. Pointwise approaches (**?**; **?**) directly optimize the relevance label or similarity score between the query and each candidate; while pairwise approaches compare the relative relevance or similarity of two candidates. Representative works of pairwise ranking include Ranking SVM (**?**) and Lambda Rank (**?**). Listwise methods either directly optimize the ranking evaluation metric or maximize the likelihood of the ground-truth rank. SVM MAP (**?**), ListNet (**?**) and ListMLE (**?**) fall in this category. In this paper, we introduce listwise ranking loss into deep metric learning, and utilize it to transfer the soft similarities between candidates and the query into student models.

Background

In this section, we review ListNet and ListMLE which are classical listwise learning to rank methods introduced by Cao *et al.* (**?**) and Xia *et al.* (**?**) for document retrieval task. These methods are closely related to our proposed method that will be elaborated in the sequel.

The core idea of these methods is to associate a probability with every rank permutation based on the relevance or similarity score between candidate $\mathbf{x}$ and query $\mathbf{q}$.

We use π to denote a permutation of the list indexes. For example, a list of four samples can have a permutation of $\pi = \{\pi(1), \pi(2), \pi(3), \pi(4)\} = \{4, 3, 1, 2\}$, which means the forth sample in the list is ranked first, the third sample second, and so on. Formally, We denote the candidate samples as $\mathbf{X} \in \mathbb{R}^{p \times n}$ with each column i being a sample $\mathbf{x}_i \in \mathbb{R}^p$. Then the probability of a specific permutation π is given as:

$$P(\pi|\mathbf{X}) = \prod_{i=1}^{n} \frac{\exp[S(\mathbf{x}_{\pi(i)})]}{\sum_{k=i}^{n} \exp[S(\mathbf{x}_{\pi(k)})]} \tag{1}$$

where $S(\mathbf{x})$ is a score function based on the distance between $\mathbf{x}$ and $\mathbf{q}$. After the probability of a single permutation is constructed, the objective function of ListNet can be defined as:

$$L_{\text{ListNet}}(\mathbf{x}) = -\sum_{\pi \in \mathcal{P}} P(\pi|\mathbf{s}) \log P(\pi|\mathbf{x}) \tag{2}$$

where $\mathcal{P}$ denotes all permutations of a list of length n, and $\mathbf{s}$ denotes the ground-truth.

Another closely related method is ListMLE(**?**). Unlike ListNet, as its name states, ListMLE aims at maximizing the likelihood of a ground truth ranking π_y. The formal definition is as follow:

$$L_{\text{ListMLE}}(\mathbf{x}) = -\log P(\pi_y|\mathbf{x}) \tag{3}$$

Our Method

In this section, we first introduce the motivation of our DarkRank by an intuitive example, then followed by the formulation and two variants of our proposed method.

Motivation

We depict our framework in Fig. **??** along with an intuitive illustration to explain the motivation of our work. In the example, the query is a digit 6, and there are two relevant digits and six irrelevant digits. Through training with such supervision, the original student network can successfully rank the relevant digits in front of the irrelevant ones. However, for the query 6, there are two 0s which are more similar than other digits. Simply using hard labels (similar or dissimilar) totally ignores such dark knowledge. However, such knowledge is crucial for the generalization ability of student models. A powerful teacher model may reflect these similarities its the embedded space. Consequently, we propose to transfer these cross sample similarities to improve the performance of student networks.

Formulation

We denote the embedded features of each mini-batch after an embedding function $f(\cdot)$ as $\mathbf{X}$. Here the choice of $f(\cdot)$ depends on the problem at hand, such as CNN for image data or DNN for text data. We further use $\mathbf{X}^s$ to denote the embedded features from student networks, and similarly $\mathbf{X}^t$ for those from teacher networks. We use one sample in the mini-batch as the anchor query $\mathbf{q} = \mathbf{x}_1$, and the rest samples in the mini-batch as candidates $\mathbf{C} = \{\mathbf{x}_2, \mathbf{x}_3, \cdots, \mathbf{x}_n\}$. We then construct a similarity score function $S(\mathbf{x})$ based on the Euclidean distance between two embeddings. The α and β are two parameters in the score function to control the scale and "contrast" of different embeddings:

$$S(\mathbf{x}) = -\alpha \|\mathbf{q} - \mathbf{x}\|_2^\beta. \tag{4}$$

After that, we propose two methods for the transfer: soft transfer and hard transfer. For soft transfer method, we construct two probability distributions $P(\pi \in \mathcal{P} \mid \mathbf{X}^s)$ and $P(\pi \in \mathcal{P} \mid \mathbf{X}^t)$ over all possible permutations (or ranks) $\mathcal{P}$ of the mini-batch based on Eqn. **??**. Then, we match these two distributions with KL divergence. For hard transfer method, we simply maximize the likelihood of the ranking π_y which has the highest probability by teacher model. Formally, we have

$$\begin{aligned} L_{\text{soft}}(\mathbf{X}^s, \mathbf{X}^t) &= D_{\text{KL}}[P(\pi \in \mathcal{P} \mid \mathbf{X}^t) \parallel P(\pi \in \mathcal{P} \mid \mathbf{X}^s)] \\ &= \sum_{\pi \in \mathcal{P}} P(\pi \mid \mathbf{X}^t) \log \frac{P(\pi \mid \mathbf{X}^t)}{P(\pi \mid \mathbf{X}^s)}, \\ L_{\text{hard}}(\mathbf{X}^s, \mathbf{X}^t) &= -\log P(\pi_y \mid \mathbf{X}^s, \mathbf{X}^t). \end{aligned} \tag{5}$$

Soft transfer considers all possible rankings. It is helpful when there are several rankings with similar probability. However, there are $n!$ possible ranking in total. It is only feasible when n is not too large. Whereas, hard transfer only considers the most possible ranking labeled by the teacher. As demonstrated in the experiments, hard transfer is a good approximation of soft transfer in the sense that it is much faster with long lists but has similar performance.

For the gradient calculation, we first use S_i to denote $S(\mathbf{x}_{\pi(i)})$ for better readability, then the gradient is calculated as below:

$$\frac{\partial P}{\partial S_i} = \prod_{k=2}^{n} \frac{\exp(S_k)}{\sum_{m=k}^{n} \exp(S_m)} - \sum_{j=1}^{i} \left[\left(\prod_{k=2}^{n} \frac{\exp(S_k)}{\sum_{m=k}^{n} \exp(S_m)} \right) \frac{\exp(S_i)}{\sum_{m=j}^{n} \exp(S_m)} \right]. \quad (6)$$

For the gradient of S_i with respect to $\mathbf{x}$, it is trivial to calculate. So we don't expand it here.

The overall loss function for the training of student networks consists both losses from ground-truth and loss from teacher knowledge. In specific, we combine large margin softmax loss (**?**), verification loss (**?**) and triplet loss (**?**) and the proposed DarkRank loss which can either be its soft or hard variant.

Experiments

In this section, we test the performance of our DarkRank method on several metric learning tasks including person re-identification, image retrieval and clustering, and compare it with several baselines and closely related works. We also conduct ablation analysis on the influence of the hyper-parameters in our method.

Datasets

We briefly introduce the datasets will be used in the following experiments.

CUHK03 CUHK03(**?**) is a large scale data for person re-identification. It contains 13164 images of 1360 identities. Each identity is captured by two cameras from different views. The author provides both detected and hand-cropped annotations. We conduct our experiments on the detected data since it is closer to the real world scenarios. Furthermore, we follow the training and evaluation protocol in(**?**). We report Rank-1, 5 and 10 performance on the first standard split.

Market1501 Market1501(**?**) contains 32668 images of 1501 identities. These images are collected from six different camera views. We follow the training and evaluation protocol in (**?**), and report mean Average Precision (mAP) and Rank-1 accuracy in both single and multiple query settings.

CUB-200-2011 The Caltech UCSD Birds-200-2011 (CUB-200-2011) dataset contains 11788 images of 200 bird species. Following the setting in (**?**), we train our network on the first 100 species (5864 images) and then perform image retrieval and clustering on the rest 100 species (5924 images). Standard F_1, NMI and Recall@1 metrics are reported.

Implementation Details

We choose Inception-BN(**?**) as our teacher network and NIN-BN(**?**) as our student network. Both networks are pre-trained on the ImageNet LSVRC image classification dataset(**?**). We first remove the fully connected layers specific to the pre-trained task, and then globally average pool the features. The output is then connected to a fully connected layer followed a L2 normalization layer to generate the final embeddings. The large margin softmax loss is directly connected to the fully connected layer. All other losses including the proposed transfer loss are built upon the L2 normalization layer. Figure **??** illustrates the architecture of our system.

We set the margin in large margin softmax loss to 3, and set the margin to 0.9 in both triplet and verification loss. We set the loss weights of verification, triplet and large margin softmax loss to 5, 0.1, 1, respectively. We choose the stochastic gradient descent method with momentum for optimization. We set the learning rate to 0.01 for the Inception-BN and 5×10^{-4} for the NIN-BN, and set the weight decay to 10^{-4}. We train the model for 100 epochs, and shrink the learning rate by a factor of 0.1 at 50 and 75 epochs. The batch size is set to 8.

For person ReID tasks, we resize all input images to 256×128 and randomly crop to 224×112. We first construct all possible cross view positive image pairs, and randomly shuffle them at the start of each epoch. For image retrieval and clustering, we resize all input images to 256×256 and randomly crop to 224×224. In addition, we flip the images in horizontal direction randomly during the training of both tasks. We implement our method in MXNet (**?**). We train our model from scratch when experimenting with CUB-200-2011 dataset, since the authors discourage the use of ImageNet pre-trianed model due to sample overlap.

Compared Methods

We introduce the models and baselines compared in our experiments. Despite the soft and hard DarkRank methods proposed by us, we also test the following methods and the combination of them with our methods:

Knowledge Distill (KD) Since the classification loss is included in our model, we test the knowledge distill with softened softmax target. According to (**?**), we set the temperature T to 4 and the loss weight to 4^2 for softmax knowledge distill method. Formally, KD can be defined as:

$$L_{\text{KD}}(\mathbf{X}^s, \mathbf{X}^t) = \sum_{i=1}^{n} D_{\text{KL}} \left[\text{softmax} \left(\frac{\mathbf{x}_i^t}{T} \right) \| \text{softmax} \left(\frac{\mathbf{x}_i^s}{T} \right) \right]. \quad (7)$$

Direct Match Distances between the query and candidates are the most straightforward form of cross sample similarities knowledge. So we directly match the distances output by teacher and student models as a baseline. Formally, the matching loss is defined as:

$$L_{\text{match}}(\mathbf{X}^s, \mathbf{X}^t) = \sum_{i=2}^{n} \left(\|\mathbf{x}_i^s - \mathbf{q}^s\|_2^2 - \|\mathbf{x}_i^t - \mathbf{q}^t\|_2^2 \right)^2. \quad (8)$$

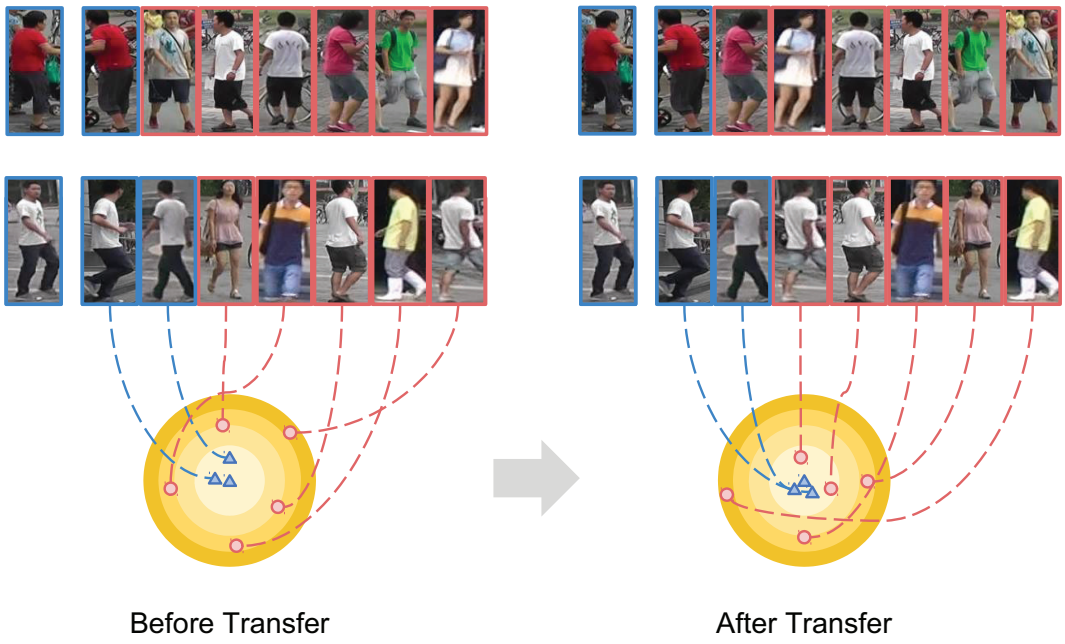


Figure 2: Selected results visualization before and after our DarkRank transfer on Market1501. The border color of image denotes its relation to the query image. With the help of teacher's knowledge, the student model learns a better distance metric that can capture similarities in images.

Person ReID Results

We present the results of Market1501 and CUHK03 in Table. **??** and Table. **??**, respectively.

Method	Single Query		Multiple Query	
	mAP	Rank 1	mAP	Rank 1
Student	58.1	80.3	66.7	86.7
Direct Match	58.5	80.3	68.0	86.7
Hard DarkRank	**63.5**	83.0	71.2	87.4
Soft DarkRank	63.1	**83.6**	**71.4**	**88.8**
KD	66.7	86.0	75.1	90.4
KD + HardRank	**68.5**	86.6	76.3	90.3
KD + SoftRank	68.2	**86.7**	**76.4**	**91.4**
Teacher	74.3	89.8	81.2	93.7

Table 1: mAP(%) and Rank-1 accuracy(%) on Market1501 of various methods. We use average pooling of features in multi-query test.

From Table. **??**, we can see that directly matching the distances between teacher and student model only has marginal improvement over the original student model. We owe the reason to that the student model struggles to match the exact distances as teacher's due to its limited capacity. As for our method, both soft and hard variants make significant improvements over the original model. They could get similar satisfactory results. As discussed in the formulation, the hard variant has great computational advantage over the soft one in training, thus it is more preferable for the practitioners. Moreover, in synergy with KD, the performance of the student model can be further improved. This complementary results demonstrate that our method indeed transfers the inter-instance knowledge in the teacher network which is ignored by KD.

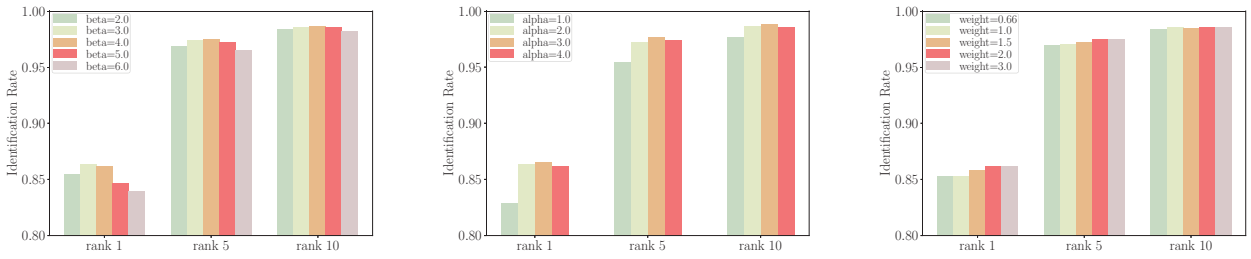
Method	Rank 1	Rank 5	Rank 10
Student	82.6	95.2	97.4
Direct Match	82.6	95.6	97.7
HardRank	86.0	**97.5**	**98.8**
SoftRank	**86.2**	**97.5**	98.6
KD	87.8	97.5	98.7
KD + HardRank	88.6	**98.2**	**99.0**
KD + SoftRank	**88.7**	98.0	**99.0**
Teacher	89.7	98.4	99.2

Table 2: Rank-1,5,10 accuracy(%) of various methods on CUHK03.

On CUHK03 dataset, we can observe similar trends as on Market1501, except that the model performance on CUHK03 is much higher, which makes the performance improvement less significant.

Ablation Analysis

In this section, we conduct ablation analysis on the hyper-parameters for our proposed soft DarkRank method, and discuss how they affect the ReID performance.



(a) contrast in score function β (b) scale factor in score function α (c) transfer loss weight λ

Figure 3: The effect of different parameters on the performance of CUHK03 validation set. Here we report Rank-1, 5, 10 results.

Contrast β Since the rank information only reveals the relative distance between the query and each candidate, it does not provide much details of the absolute distance in the metric space. If the distances of candidates and the query are close, the associated probabilities for the permutations are also close, which makes it hard to distinguish from a good ranking to a bad ranking. So we introduce the contrast parameter β to sharpen the differences of the scores. We test different values of β on CUHK03 validation set, and find 3.0 is where the model performance peaks. Figure **??** shows the details.

Scaling factor α While constraining embeddings on the unit hyper-sphere is the standard setting for metric learning methods in person ReID, a recent work(**?**) shows that small embedding norm may hurt the representation power of embeddings. We compensate this by introducing a scaling factor α and test different values on the CUHK03 validation set. Figure **??** shows the influences on performance of different scaling factors. We choose $\alpha = 3.0$ where the model performance peaks.

Loss weight λ During the training process, it is important to balance the transfer loss and the original training loss. We set the loss weight of our transfer loss to 2.0 according to the results in Fig. **??**. Note that it also reveals that the performance of our model is quite stable in a large range of λ.

Transfer without Identity

Method	Single Query		Multiple Query	
	mAP	Rank 1	mAP	Rank 1
FitNet	64.0	83.4	72.4	88.6
FitNet + DarkRank	67.3	85.3	74.9	90.3

Table 3: mAP(%) and Rank-1 accuracy(%) on Market1501 of FitNet. We use average pooling features in multi-query test.

Supervised learning has achieved great success in computer vision, but the majority of collected data remains unlabeled. In tasks like self-supervised learning(**?**), class level supervision is not available. The supervision signal purely comes from pairwise similarity. Knowledge transfer methods like KD are hard to fit in these cases. As an advantage, our method utilize instance level supervision, and thus is available for both supervised and unsupervised tasks. Another well-known instance level method is FitNet(**?**), which directly matches the embeddings of student and teacher with L2 loss. We compare the transfer performance of FitNet with and without our DarkRank. As shown in Table. **??**, FitNet achieves similar performance as our method alone. And combined with our method, a significant improvement is achieved. This result further proves that our method utilizes a different kind of information complimenting existing intra-instance methods.

Image Retrieval and Clustering Results

Method	F_1	NMI	Recall@1
Student	0.153	0.461	0.311
DarkRank	0.168	0.483	0.340
Teacher	0.172	0.484	0.367

Table 4: F_1, NMI, Recall@1 of DarkRank on CUB-200-2011.

The goal of image clustering is to group images into categories according to their visual similarity. And image retrieval is about finding the most similar images in a gallery for a given query image. These tasks rely heavily on the embeddings learnt by model, since the similarity of a image pair is generally calculated based on the Euclidean or Mahalanobis distance between their embeddings. The metrics we adopted for image clustering are F_1 and NMI. F_1 is the harmonic mean of precision and recall. $F_1 = 2PR/(P + R)$. The Normalized Mutual Information(NMI) reflects the correspondence between candidate clustering Ω and ground-truth clustering $\mathbb{C}$ of the same dataset. NMI $= 2I(\Omega, \mathbb{C})/(H(\Omega) + H(\mathbb{C}))$, here $I(\cdot)$ and $H(\cdot)$ are mutual information and entropy, respectively. NMI ranges from 0 to 1, where higher value indicates better correspondence. We choose Recall@1, which is the percentage of

returned images belongs to the same category as the query image, as the metric for image retrieval task. The networks and hyper-parameters are as stated in implementation details section. We present the image retrieval and clustering results on CUB-200-2011 in Table. **??**. The results show our method achieves significant margin in all F_1, NMI, Recall@1 metrics. This again shows our method is generally applicable to various kinds of metric learning tasks.

Speedup

Model	NIN-BN	Inception-BN
Number of parameters	7.6M	10.3M
Images / Second	526	178
Speedup	2.96	1.00
Rank-1 on CUHK03	0.887	0.897
Rank-1 on Market1501	0.867	0.898

Table 5: Complexity and performance comparisons of the student network and teacher network.

We summarize the complexity and the performance of the teacher and the student network in Table. **??**. The speed is tested on Pascal Titan X with MXNet (**?**). We don't further optimize the implementation for testing. Note that, as the first work that studies knowledge transfer in deep metric learning model, we choose two off-the-shelf network architectures rather than deliberately designing them. Even though, we still achieve a 3X wall time acceleration with minor performance loss. We believe we can further benefit from the latest network design philosophy (**?**; **?**), and achieve even better speedup.

Conclusion

In this paper, we have proposed a new type of knowledge – cross sample similarities for model compression and acceleration. To fully utilize the knowledge, we have modified the classical listwise rank loss to bridge teacher networks and student networks. Through our knowledge transfer, the student model can significantly improve its performance on various metric learning tasks. Moreover, by combining with other transfer methods which exploit the intra-instance knowledge, the performance gap between teachers and students can be further narrowed. Particularly, without deliberately tuning the network architecture, our method achieves about three times wall clock speedup with minor performance loss with off-the-shelf networks. We believe our preliminary work provides a new possibility for knowledge transfer based model acceleration. In the future, we would like to exploit the use of cross sample similarities in more general applications beyond deep metric learning.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (No. 61773375, No. 61375036, No. 61602481, No. 61702510), and in part by the Microsoft Collaborative Research Project.

References

Bai, S.; Bai, X.; and Tian, Q. 2017. Scalable person re-identification on supervised smoothed manifold. In *CVPR*.

Bromley, J.; Bentz, J. W.; Bottou, L.; Guyon, I.; LeCun, Y.; Moore, C.; Säckinger, E.; and Shah, R. 1993. Signature verification using a Siamese time delay neural network. *International Journal of Pattern Recognition and Artificial Intelligence*.

Bucila, C.; Caruana, R.; and Niculescu-Mizil, A. 2006. Model compression: Making big, slow models practical. In *KDD*.

Burges, C. J. C.; Ragno, R.; and Le, Q. V. 2006. Learning to rank with nonsmooth cost functions. In *NIPS*.

Cao, Z.; Qin, T.; Liu, T.-Y.; Tsai, M.-F.; and Li, H. 2007. Learning to rank: from pairwise approach to listwise approach. In *ICML*.

Chen, T.; Li, M.; Li, Y.; Lin, M.; Wang, N.; Wang, M.; Xiao, T.; Xu, B.; Zhang, C.; and Zhang, Z. 2016. MXNet: A flexible and efficient machine learning library for heterogeneous distributed systems. In *NIPS Workshop*.

Chen, J.; Zhang, Z.; and Wang, Y. 2015. Relevance metric learning for person re-identification by exploiting listwise similarities. *IEEE Transactions on Image Processing*.

Cheng, D.; Gong, Y.; Zhou, S.; Wang, J.; and Zheng, N. 2016. Person re-identification by multi-channel parts-based CNN with improved triplet loss function. In *CVPR*.

Cossock, D., and Zhang, T. 2006. Subset ranking using regression. In *International Conference on Computational Learning Theory*.

Czarnecki, W. M.; Osindero, S.; Jaderberg, M.; Swirszcz, G.; and Pascanu, R. 2017. Sobolev training for neural networks. In *NIPS*.

Han, S.; Pool, J.; Tran, J.; and Dally, W. 2015. Learning both weights and connections for efficient neural network. In *NIPS*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*.

Herbrich, R.; Graepel, T.; and Obermayer, K. 1998. Large margin rank boundaries for ordinal regression. In *NIPS Workshop*.

Hermans, A.; Beyer, L.; and Leibe, B. 2017. In defense of the triplet loss for person re-identification. In *arXiv:1703.07737*.

Hinton, G. E.; Vinyals, O.; and Dean, J. 2014. Distilling the knowledge in a neural network. In *NIPS Workshop*.

Huang, Z., and Wang, N. 2017. Like what you like: Knowledge distill via neuron selectivity transfer. In *arXiv:1707.01219*.

Huang, G.; Liu, Z.; Weinberger, K. Q.; and van der Maaten, L. 2017. Densely connected convolutional networks. In *CVPR*.

Hubara, I.; Courbariaux, M.; Soudry, D.; El-Yaniv, R.; and Bengio, Y. 2016. Binarized neural networks. In *NIPS*.

Kwok, J. T., and Tsang, I. W. 2003. Learning with idealized kernels. In *ICML*.

LeCun, Y.; Denker, J. S.; and Solla, S. A. 1989. Optimal brain damage. In *NIPS*.

Li, W.; Zhao, R.; Xiao, T.; and Wang, X. 2014. DeepReID: Deep filter pairing neural network for person re-identification. In *CVPR*.

Lin, M.; Chen, Q.; and Yan, S. 2014. Network in network. In *ICLR*.

Liu, J.; Zha, Z.-J.; Tian, Q. I.; Liu, D.; Yao, T.; Ling, Q.; and Mei, T. 2016a. Multi-scale triplet CNN for person re-identification. In *ACM MM*.

Liu, W.; Wen, Y.; Yu, Z.; and Yang, M. 2016b. Large-margin softmax loss for convolutional neural networks. In *ICML*.

Qian, Q.; Jin, R.; Zhu, S.; and Lin, Y. 2015. Fine-grained visual categorization via multi-stage metric learning. In *CVPR*.

Ranjan, R.; Castillo, C. D.; and Chellappa, R. 2017. L2-constrained softmax loss for discriminative face verification. In *arXiv:1704.00438*.

Rastegari, M.; Ordonez, V.; Redmon, J.; and Farhadi, A. 2016. XNOR-Net: Imagenet classification using binary convolutional neural networks. In *ECCV*.

Romero, A.; Ballas, N.; Kahou, S. E.; Chassang, A.; Gatta, C.; and Bengio, Y. 2015. FitNets: Hints for thin deep nets. In *ICLR*.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; Berg, A. C.; and Fei-Fei, L. 2015. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*.

Schroff, F.; Kalenichenko, D.; and Philbin, J. 2015. FaceNet: A unified embedding for face recognition and clustering. In *CVPR*.

Shashua, A., and Levin, A. 2003. Ranking with large margin principle: Two approaches. In *NIPS*.

Song, H. O.; Xiang, Y.; Jegelka, S.; and Savarese, S. 2016. Deep metric learning via lifted structured feature embedding. In *CVPR*.

Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S. E.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; and Rabinovich, A. 2015. Going deeper with convolutions. In *CVPR*.

Taigman, Y.; Yang, M.; Ranzato, M.; and Wolf, L. 2014. DeepFace: Closing the gap to human-level performance in face verification. In *CVPR*.

Wang, X., and Gupta, A. 2015. Unsupervised learning of visual representations using videos. In *ICCV*.

Wang, F.; Zuo, W.; Lin, L.; Zhang, D.; and Zhang, L. 2016. Joint learning of single-image and cross-image representations for person re-identification. In *CVPR*.

Weinberger, K. Q.; Blitzer, J.; and Saul, L. 2006. Distance metric learning for large margin nearest neighbor classification. In *NIPS*.

Wen, Y.; Zhang, K.; Li, Z.; and Qiao, Y. 2016. A discriminative feature learning approach for deep face recognition. In *ECCV*.

Xia, F.; Liu, T.-Y.; Wang, J.; Zhang, W.; and Li, H. 2008. Listwise approach to learning to rank: theory and algorithm. In *ICML*.

Xie, S.; Girshick, R.; Dollr, P.; Tu, Z.; and He, K. 2017. Aggregated residual transformations for deep neural networks. In *CVPR*.

Xing, E. P.; Jordan, M. I.; Russell, S. J.; and Ng, A. Y. 2003. Distance metric learning with application to clustering with side-information. In *NIPS*.

Yue, Y.; Finley, T.; Radlinski, F.; and Joachims, T. 2007. A support vector method for optimizing average precision. In *ACM SIGIR*.

Zagoruyko, S., and Komodakis, N. 2017. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *ICLR*.

Zheng, L.; Shen, L.; Tian, L.; Wang, S.; Wang, J.; and Tian, Q. 2015. Scalable person re-identification: A benchmark. In *ICCV*.