arXiv:1707.01220v1 [cs.CV] 5 Jul 2017

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,
Institute of Automation, Chinese Academy of Sciences(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, natural language processing, etc. 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.* [1] 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 on the pedestrian re-identification task. The results are quite encouraging. Our DarkRank 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.

1 Introduction

Metric learning is the basis for many computer vision tasks, including face verification[2, 3] and pedestrian re-identification[4, 5]. In recent years, end-to-end deep metric learning method which jointly learns feature representation and metric has achieved great success[6, 7, 2]. A key factor for the success of these deep metric learning methods is the powerful network architectures[8, 9, 10]. 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[11, 12], model quantization[13, 14] and knowledge transfer[15, 16, 1]. 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

Submitted to 31st Conference on Neural Information Processing Systems (NIPS 2017).

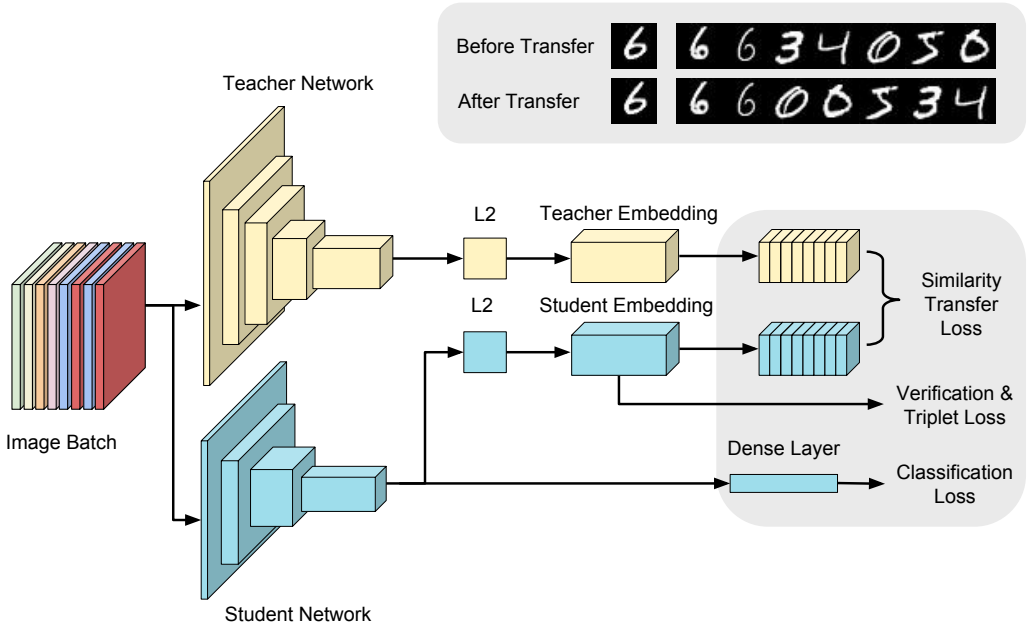


Figure 1: The network architecture of our DarkRank method.

trains a smaller student network guided by a larger and more powerful teacher. Among these methods, knowledge transfer based methods are the most practical. Because compared with other methods that mostly need tailor made hardwares or implementations, they can archive considerable acceleration without bells and whistles.

Knowledge Distill (KD)[1] and its variants[15, 16] 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 Sec. 4.1, and depict our method in Fig. 1.

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

- We exploit 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 listwise learning to rank methods[17, 18] to solve it.
- We test our proposed method on two large scale person re-identification datasets. Our method can significantly improve the performance of student network. When further combined with KD, Our methods achieve three times wall time speedup with minor performance degradation.

2 Related works

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

2.1 Deep Metric Learning

Different from most traditional metric learning methods that focus on learning a Mahalanobis distance in Euclidean space[19, 20] or high dimensional kernel space[21], 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 discriminate inter-class embeddings and reduce the intra-class variance. Classification loss and its variants[22, 23] can learn robust features that help to discriminate samples from different classes. However, for the 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 point in the label space, thus ignores the intra-class variance. Verification loss[24] is a popular alternative because it directly encodes both the similarity ans dissimilarity supervisions. The weakness of verification loss is that it can not guarantee a margin between inter-class samples. Triplet loss and its variants[25, 26, 27] overcome this disadvantage by imposing an order on embedding triplets other than pairs. But its good performance requires a careful design of sampling and training procedure[2, 26]. Other related work include center loss [23] which maintains a shifting center for each class to reduce the intra-class variance by simultaneously drawing the center and the sample towards each other.

2.2 Knowledge Transfer for Model Acceleration and Compression

In [28], 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[1]. The insight comes from that the softened probabilities output by classifier encodes 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, thus improve the generalization ability of them. Subsequent works FitNets[16] and Attention Transfer[15] tried to exploit other diverse knowledges in intermediate feature maps of CNNs to improve the performance. In this paper, we explore a unique type of knowledge inside deep metric learning model – cross sample similarities to train a better student network.

2.3 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 [29, 30] directly optimize the relevance label or similarity score between query and each candidate; while pairwise approaches compare the relative relevance or similarity of two candidates. Representative works of pairwise ranking include Ranking SVM [31] and Lambda Rank [32]. Listwise methods either directly optimize the ranking evaluation metric or maximize the likelihood of the ground-truth rank. SVM MAP [33], ListNet [17] and ListMLE [18] 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 query into student models.

3 Background

In this section, we review ListNet and ListMLE which are classical listwise learning to rank methods introduced by Cao *et al.* [17] and Xia *et al.* [18] 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 possible rank permutation based on the relevance or similarity score between each candidate $\mathbf{x}$ and query $\mathbf{q}$.

We use π to denote a permutation of the indexes of samples in a list. 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 rank or permutation π is given as:

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

where $S(\mathbf{x})$ is a score function based on the distance between $\mathbf{x}$ and the query $\mathbf{q}$, and π denotes a permutation of a list of length n. 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[18]. 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:

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

4 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.

4.1 Motivation

We depict our framework in Fig. 1 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.

4.2 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 embedding 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. 1. 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 the n is not too

large. Whereas, hard transfer only consider the most possible ranking labeled by the teacher. As demonstrated in the experiments, hard transfer is a good approximation to 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:

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

For the gradient of $\mathbf{x}$ with respect to S_i, 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 [22], verification loss [24] and triplet loss [2] and the proposed DarkRank loss which can either be its soft or hard variant.

5 Experiments

In this section, we test the performance of our DarkRank method on two large-scale person re-identification datasets, and compare it with several baselines and closely related works. Next, We conduct ablation analysis on the influence of the hyperparameters in our method.

5.1 Datasets

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

CUHK03. CUHK03[34] contains 13164 images from 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[34]. We report precision of rank 1, 5, 10, and 20 on the first split.

Market1501. Market1501[35] contains 32668 images of 1501 identities. These images are collected from six different cameras. We follow the training and evaluation protocol in [35], and report mean Average Precision (mAP) and precision of rank 1 in both single query and multiple query settings.

5.2 Implementation Details

We choose the Inception-BN[10] model as our teacher network and the NIN-BN[36] model as our student network. Both networks are pre-trained on the ImageNet LSVRC image classification dataset[37]. We first remove the fully connected layer specific to the pre-trained task, and then globally average pool the features. Lastly, it is connected to a fully connected layer followed a L2 normalization layer to output the final embedded features. 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 1 shows the architecture of our network.

We set the margin in large margin softmax loss to 3, and 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 as our optimizer. We set the learning rate to 0.01 for the Inception-BN model and 5×10^{-4} for the NIN-BN model, and set the weight decay to 10^{-4}. We train the model for 100 epochs, and we shrink the learning rate by a factor of 0.1 at the 50 and 75 epochs. The mini-batch size is set to 8.

We resize all input images to 256×128 and randomly crop them to 224×112. We also randomly flip the images during the training. We first construct all possible cross view positive image pairs, and randomly shuffle them at the start of each epoch. We implement our method in MXNet [38], and use the provided pre-trained models.

5.3 Models

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

Knowledge Distill (KD): Since the classification loss is included in our model, we also test the knowledge distill with softened softmax target. According to [1], we set the temperature T to 4 and the loss weight to 16 for softmax knowledge distill method.

FitNet: This method uses l_2 loss to directly match the student's embeddings with the teacher's.

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

$$L_{\text{match}}(\mathbf{X}^s, \mathbf{X}^t) = \sum_{i=2}^{n} \left(\|\mathbf{x}_i^s - \mathbf{q}^s\|_2 - \|\mathbf{x}_i^t - \mathbf{q}^t\|_2 \right)^2 \tag{7}$$

5.4 Results

We present the results of CUHK03 and Market1501 in Table. 1 and Table. 2, respectively.

Table 1: Precision(%) of various methods on CUHK03.

Method	Rank 1	Rank 5	Rank 10	Rank 20
Student	82.6	95.2	97.4	98.8
Direct Match	82.6	95.6	97.7	98.9
FitNet	**86.4**	**97.5**	98.6	**99.2**
Hard DarkRank	86.0	**97.5**	**98.8**	**99.2**
Soft DarkRank	86.2	**97.5**	98.6	**99.2**
KD	87.8	97.5	98.7	99.4
KD + FitNet	88.3	97.9	98.9	99.3
KD + Hard DarkRank	88.6	**98.2**	**99.0**	**99.4**
KD + Soft DarkRank	**88.7**	98.0	**99.0**	**99.4**
Teacher	89.7	98.4	99.2	99.4

Table 2: mAP(%) and Precision(%) on Market1501 of various methods. We use average pooling features in multi-query test.

	Single Query		Multiple Query	
Method	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
FitNet	**64.0**	83.4	**72.4**	88.6
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 + FitNet	67.4	86.0	75.5	90.7
KD + Hard DarkRank	**68.5**	86.6	76.3	90.3
KD + Soft DarkRank	68.2	**86.7**	**76.4**	**91.4**
Teacher	74.3	89.8	81.2	93.7

From Table 1, 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

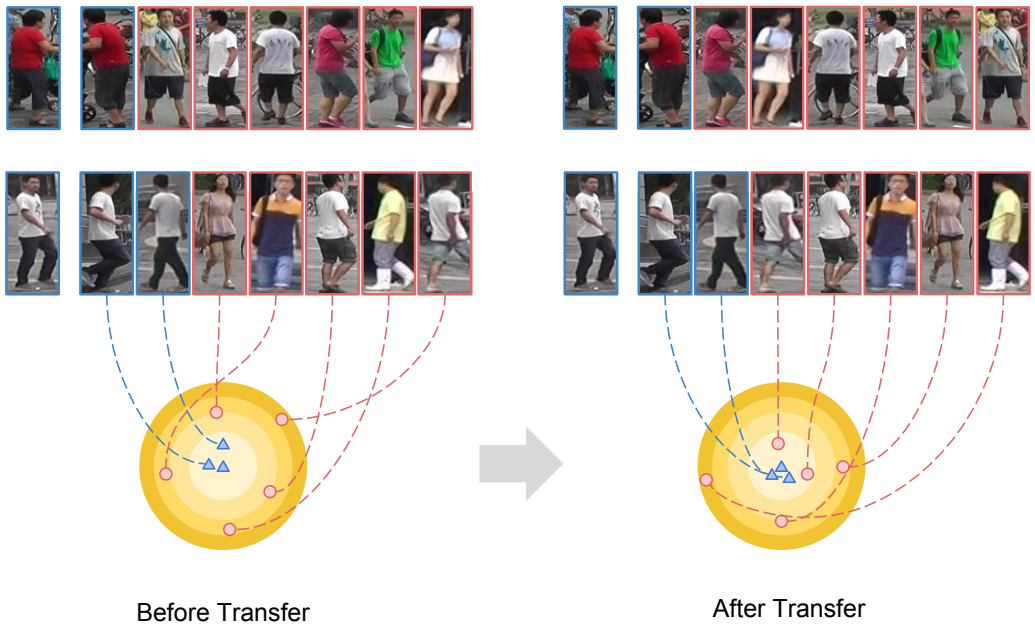


Figure 2: Selected visual results before and after our DarkRank transfer on Market1501.

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 Sec. 4.2, 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 a different kind of knowledge in the teacher network which is ignored by KD. Note that there are also some cases that the identity is not available, thus makes KD not applicable, such as tasks like extreme classification [39]. In those cases, our method is still feasible for knowledge transfer.

In Market1501 dataset, we can observe similar trends as in CUHK03, except that Market1501 is much harder, which makes the performance improvement even more significant.

An interesting observation is that though FitNet outperforms our DarkRank in standalone transfer, but when combined with KD, its results are significantly worse than ours. The reason behind this is that both FitNet and KD are trying to transfer the knowledge within a single sample. In contrast, our method aims at transferring cross sample similarities that represent a different type of knowledge which all single sample based methods fail to attend to.

Some visual results before and after our DarkRank transfer are shown in Fig. 2. Different border colors of images denote the their relations to the query image. From the figure, we can see that with the help of teacher's knowledge, the student model learns a finer metric that can capture similarities in images like clothing.

5.5 Ablation Analysis

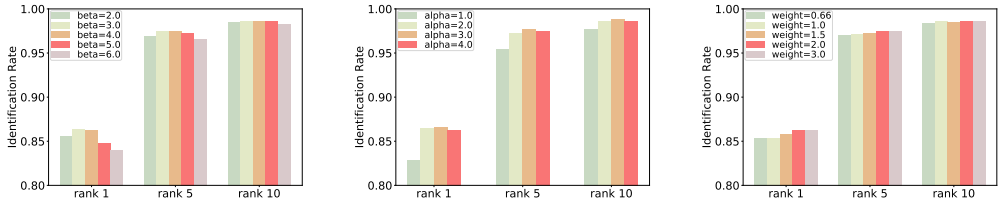
In this section, we conduct ablation analysis on the hyperparameters in our proposed soft DeepRank method, and discuss how them affect the final performance.

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 between each candidate and query are too 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 3(a) shows the details.

Scaling factor α. While constraining the embedding to live on the unit hypersphere is a standard setting for metric learning methods in person re-identification filed, a recent work[40] shows that small α can hurt the representing ability of embeddings, and suggests to make α a tunable parameter. We test different α on the CUHK03 validation set. Figure 3(b) shows the influences on performance of different scaling factors. We set $\alpha = 3.0$ where the model performance peaked.

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. 3(c). Note that it also reveals that the performance of our model is quite stable in a large range of λ.



(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. We report rank 1, 5, 10 results, respectively.

5.6 Speedup

Table 3: Complexity and performance comparisons of the student network and teacher network.

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	88.7	89.7
Rank 1 on Market1501	86.7	89.8

We summarize the complexity and the performance of the teacher and student network in Table. 3. The speed is tested on Pascal Titan X with MXNet [38]. We don't further optimize the implementation for testing. Note that, as the first work that studies knowledge tranfer 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 [9, 41], and achieve even better speedup.

6 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 two largest pedestrian re-identification datasets. Moreover, by combining with KD which exploits the knowledge from individual sample level, 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.

References

[1] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *NIPS Workshop*, 2014.

[2] Florian Schroff, Dmitry Kalenichenko, and James Philbin. FaceNet: A unified embedding for face recognition and clustering. *CVPR*, 2015.

[3] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. DeepFace: Closing the gap to human-level performance in face verification. *CVPR*, 2014.

[4] Faqiang Wang, Wangmeng Zuo, Liang Lin, David Zhang, and Lei Zhang. Joint learning of single-image and cross-image representations for person re-identification. *CVPR*, 2016.

[5] Jiaxin Chen, Zhaoxiang Zhang, and Yunhong Wang. Relevance metric learning for person re-identification by exploiting listwise similarities. *IEEE Transactions on Image Processing*, 24:4741–4755, 2015.

[6] Qi Qian, Rong Jin, Shenghuo Zhu, and Yuanqing Lin. Fine-grained visual categorization via multi-stage metric learning. *CVPR*, 2015.

[7] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding. *CVPR*, 2016.

[8] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. *CVPR*, 2017.

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

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

[11] Yann LeCun, John S. Denker, and Sara A. Solla. Optimal brain damage. *NIPS*, 1989.

[12] Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. *NIPS*, 2015.

[13] Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Binarized neural networks. *NIPS*, 2016.

[14] Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi. XNOR-Net: Imagenet classification using binary convolutional neural networks. *ECCV*, 2016.

[15] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *ICLR*, 2017.

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

[17] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. Learning to rank: from pairwise approach to listwise approach. *ICML*, 2007.

[18] Fen Xia, Tie-Yan Liu, Jue Wang, Wensheng Zhang, and Hang Li. Listwise approach to learning to rank: theory and algorithm. *ICML*, 2008.

[19] Eric P. Xing, Michael I. Jordan, Stuart J Russell, and Andrew Y. Ng. Distance metric learning with application to clustering with side-information. *NIPS*, 2003.

[20] James T. Kwok and Ivor W. Tsang. Learning with idealized kernels. *ICML*, 2003.

[21] Kilian Q Weinberger, John Blitzer, and Lawrence Saul. Distance metric learning for large margin nearest neighbor classification. *NIPS*, 2006.

[22] Weiyang Liu, Yandong Wen, Zhiding Yu, and Meng Yang. Large-margin softmax loss for convolutional neural networks. *ICML*, 2016.

[23] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. A discriminative feature learning approach for deep face recognition. *ECCV*, 2016.

[24] Jane Bromley, James W. Bentz, Léon Bottou, Isabelle Guyon, Yann LeCun, Cliff Moore, Eduard Säckinger, and Roopak Shah. Signature verification using a "Siamese" time delay neural network. *International Journal of Pattern Recognition and Artificial Intelligence*, 7:669–688, 1993.

[25] De Cheng, Yihong Gong, Sanping Zhou, Jinjun Wang, and Nanning Zheng. Person re-identification by multi-channel parts-based CNN with improved triplet loss function. *CVPR*, 2016.

[26] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*, 2017.

[27] Jiawei Liu, Zheng-Jun Zha, Q. I. Tian, Dong Liu, Ting Yao, Qiang Ling, and Tao Mei. Multi-scale triplet CNN for person re-identification. *ACM MM*, 2016.

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

[29] David Cossock and Tong Zhang. Subset ranking using regression. *International Conference on Computational Learning Theory*, 2006.

[30] Amnon Shashua and Anat Levin. Ranking with large margin principle: Two approaches. *NIPS*, 2003.

[31] Ralf Herbrich, Thore Graepel, and Klaus Obermayer. Large margin rank boundaries for ordinal regression. *NIPS Workshop*, 1998.

[32] Christopher J. C. Burges, Robert Ragno, and Quoc V. Le. Learning to rank with nonsmooth cost functions. *NIPS*, 2006.

[33] Yisong Yue, Thomas Finley, Filip Radlinski, and Thorsten Joachims. A support vector method for optimizing average precision. *ACM SIGIR*, 2007.

[34] Wei Li, Rui Zhao, Tong Xiao, and Xiaogang Wang. DeepReID: Deep filter pairing neural network for person re-identification. *CVPR*, 2014.

[35] Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable person re-identification: A benchmark. *ICCV*, 2015.

[36] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *ICLR*, 2014.

[37] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[38] Tianqi Chen, Mu Li, Yutian Li, Min Lin, Naiyan Wang, Minjie Wang, Tianjun Xiao, Bing Xu, Chiyuan Zhang, and Zheng Zhang. MXNet: A flexible and efficient machine learning library for heterogeneous distributed systems. *NIPS Workshop*, 2016.

[39] Anna Choromanska, Alekh Agarwal, and John Langford. Extreme multi-class classification. *NIPS Workshop*, 2015.

[40] Rajeev Ranjan, Carlos D. Castillo, and Rama Chellappa. L2-constrained softmax loss for discriminative face verification. *arXiv preprint arXiv:1704.00438*, 2017.

[41] Gao Huang, Zhuang Liu, Kilian Q Weinberger, and Laurens van der Maaten. Densely connected convolutional networks. *CVPR*, 2017.