Correlation Congruence for Knowledge Distillation

Baoyun Peng[1*†], Xiao Jin[2*], Jiaheng Liu[3], Dongsheng Li[1‡]
Yichao Wu[2], Yu Liu[4], Shunfeng Zhou[2], Zhaoning Zhang[1]
[1] Natioan University of Defense Technology, [2] Sensetime Group Limited,
[3] Beihang University, [4] The Chinese University of Hong Kong.
{pengbaoyun13,dsli}@nudt.edu.cnom {jinxiaocuhk,zzningxp}@gmail.com
{zhoushunfeng,wuyichao}@sensetime.com liujiaheng@buaa.edu.cn yuliu@ee.cuhk.edu.hk

Abstract

Most teacher-student frameworks based on knowledge distillation (KD) depend on a strong congruent constraint on instance level. However, they usually ignore the correlation between multiple instances, which is also valuable for knowledge transfer. In this work, we propose a new framework named correlation congruence for knowledge distillation (CCKD), which transfers not only the instance-level information but also the correlation between instances. Furthermore, a generalized kernel method based on Taylor series expansion is proposed to better capture the correlation between instances. Empirical experiments and ablation studies on image classification tasks (including CIFAR-100, ImageNet-1K) and metric learning tasks (including ReID and Face Recognition) show that the proposed CCKD substantially outperforms the original KD and other SOTA KD-based methods. The CCKD can be easily deployed in the majority of the teacher-student framework such as KD and hint-based learning methods.

1. Introduction

Over the past few decades, various deep neural network (DNN) models have achieved state-of-the-art performance in many vision tasks [28, 29, 8]. Generally, networks with many parameters and computations perform superior to those with fewer parameters and computations when trained on the same dataset. Nevertheless, it's difficult to deploy such large networks on resource-limited embedded systems. Along with the increasing demands for low-cost networks running on embedded systems, there is an urgency for getting a minor network with less computation and memory consumptions, while narrowing the gap of performance between a minor network and a large network.

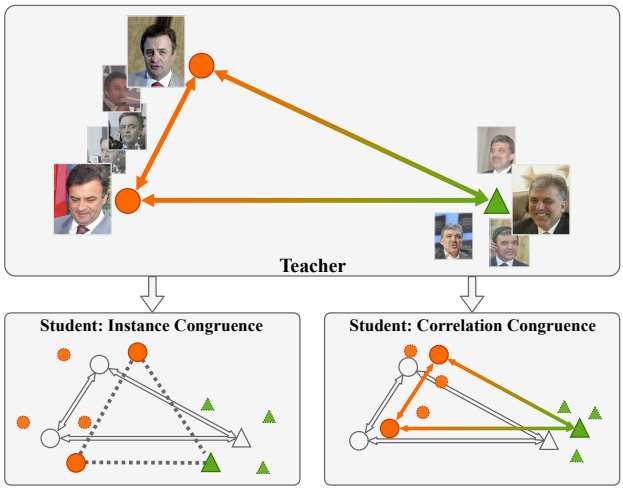


Figure 1: The difference between instance congruence and correlation congruence. When focusing on only instance congruence, the correlation between instances of the student may be much different from the teacher's, and the cohesiveness of intra-class would be worse. CCKD solve the problem by adding a correlation congruence when transferring knowledge.

Several techniques have been proposed to address this issue, e.g. parameter pruning and sharing [11, 24], compact convolutional filters [38, 16], low-rank factorization [18, 6] and knowledge distillation [15]. Among these approaches, knowledge distillation has been proved to be an effective way to promote the performance of a small network by mimicking the behavior of a high-capacity network. It works by adding a strong congruent constraint on outputs of teacher and student for each input instance to encourage the student to mimic teacher's behavior, e.g., minimizing the KullbackLeibler divergence of predictions [15] or minimizing the euclidean distance of feature representations [21] between teacher and student.

*Equal contribution.
†This work was done while Baoyun Peng was an intern at SenseTime.
‡Corresponding author.

However, it's hard for the student to learn a mapping function identical to the teacher due to the gap (network capacity) between teacher and student. By focusing on only instance congruence, the student would learn a much more different instances correlation from the teacher, as shown in Figure 1. Usually, the embedding space of a teacher possesses the characteristic that intra-class instances cohere together while inter-class instances separate from each other. But its counterpart of student model trained by instance congruence would lack such desired characteristic.

We claim that beyond instance congruence, the correlation between instances is also valuable knowledge for promoting the performance of the student. Based on this philosophy, we propose a new distillation framework called Correlation Congruence Knowledge Distillation (CCKD) which focus on not only instance congruence, but also correlation congruence. CCKD aims to transfer the correlation knowledge between instances to the student, as shown in Figure 1, and can be easily implemented and trained with mini-batch. The only requirement for CCKD is that the dimension for both the teacher and student should be the same. To cope with the mismatch of feature representations of teacher and student network, we apply a fully-connected layer with the same dimension for both teacher and student network. We conduct various experiments on four representative tasks and different networks to validate the effectiveness of the proposed approach.

Our contributions in this paper are summarized as follows:

1. We propose a new distillation framework named correlation congruence knowledge distillation (CCKD), which focuses on not only instance congruence but also correlation congruence.

2. We introduce a general kernel-based method to capture the correlation between instances in a mini-batch better. We have evaluated and analyzed the impact of different correlation metrics on different tasks.

3. We explore different sampler strategies for mini-batch training to further improve the correlation knowledge transfer.

Extensive empirical experiments and ablation studies show the effectiveness of the proposed method in different tasks (CIFAR-100, ImageNet-1K, person re-identification, and face recognition) to improve distillation performance.

2. Related Work

Since this paper focuses on training a small but high-performance network based on knowledge distillation, we discuss related works in model compression and acceleration, knowledge distillation in this section. In both areas, various approaches have been proposed over the past few years. We summarize them as follows.

Model Compression and Acceleration. Model compression and acceleration aim to create a network with few computation and parameters cost, meanwhile maintaining high performance. A straightway is to design light-weight but powerful network since the original convolution network has many redundant parameters. For example, depthwise separable convolution is used to replacing standard convolution in [16]. Pointwise group convolution and channel shuffle are proposed to reduce the burden of computation while maintaining high accuracy in [38]. Another way is network pruning, which boosts the speed of inference by pruning the neurons or filters with low importance based on certain criteria [11, 24]. In [18, 6], weights were decomposed through low-rank decomposition to save memory cost. Quantization seeks to use low-precision bits to store model's weights or activation outputs [10, 17, 34].

Knowledge Distillation. Transferring knowledge from a cumbersome network to a small network is a classical problem, and it has drawn much attention in recent years. In [15], Hinton *et al*. propose knowledge distillation (KD), in which the student network was trained by the soft output of an ensemble of teacher networks. Comparing to the one-hot label, the output from a well-performed teacher network contains more information about the fine-grained structure among data, consequently helping the student achieve better performance. Since then, there have been works exploring variants of knowledge distillation. In [2], Ba and Caruana show that the performance of a shallower and wider network trained by KD can approximate to deeper ones. Romero *et al*. [25] propose to transfer the knowledge using not only final outputs but also intermediate ones, and add a regressor on intermediate layers to match different size of teacher's and student's outputs. In [37], the authors propose an attention-based method to match the activation-based and gradient-based spatial attention maps. In [36], the flow of solution procedure (FSP), which is generated by computing the Gram matrix of features across layers, was used for knowledge transfer. To improve the robustness of the student, Sau and Balasubramanian [27] perturbe the logits of a teacher as a regularization.

Different from above offline training methods, several works adopt collaboratively training strategy. Deep mutual learning [39] conducts distillation collaboratively for peer student models by learning from each other. Anil *et al*. [1] further extend this idea by online distillation of multiples networks. In their work, networks are trained in parallel, and the knowledge is shared by using distillation loss to accelerate the training process.

Besides, several works utilize the adversarial method for modeling knowledge transfer between teacher and student [35, 13, 14]. In [35], they adopt generative adversarial net-

works combined with distillation to better aligning the distributions between teacher and student in embedding space. Byeongho *et al*. [14] adopt the adversarial method to discover adversarial samples supporting decision boundary.

In this paper, beyond instance knowledge, we take the correlation in embedded space between instances as valuable knowledge to transfer correlation among instances in the embedded space between for knowledge distillation.

3. CCKD

3.1. Background and Notations

We refer a well-performed teacher network with parameters $\boldsymbol{W}_t$ as T and a new student network with parameters $\boldsymbol{W}_s$ as S like in [15, 37, 36, 1, 25]. The input dataset of the network is noted as $\chi = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n\}$, and the corresponding ground truth is noted as $\mathcal{Y} = \{\boldsymbol{y}_1, \boldsymbol{y}_2, ..., \boldsymbol{y}_n\}$, n represents the number of samples in dataset. Since deep network can be viewed as a mapping function stacked by multiple non-linear layers, we note $\phi_t(\boldsymbol{x}; \boldsymbol{W}_t)$ and $\phi_s(\boldsymbol{x}; \boldsymbol{W}_s)$ as the mapping functions of teacher and student, $\boldsymbol{x}$ represents the input data. $\boldsymbol{f}_s$ and $\boldsymbol{f}_t$ represent the feature representations of teacher and student. The logits of teacher and student are noted as $\boldsymbol{z}_t = \phi(\boldsymbol{x}; \boldsymbol{W}_s)$ and $\boldsymbol{z}_s = \phi(\boldsymbol{x}; \boldsymbol{W}_t)$. $\boldsymbol{p}_t = softmax(\boldsymbol{z}_t)$ and $\boldsymbol{p}_s = softmax(\boldsymbol{z}_s)$ represent the final prediction probilities of teacher and student.

3.2. Knowledge Distillation

Overparameterized networks have shown powerful optimization properties to learn the desired mapping function from data [7], of which the output reflects fine-grained structure one-hot labels might ignore. Based on this insight, knowledge distillation was first proposed in [3] for model compression, then Hinton *et al*. [15] popularized it. The idea of knowledge distillation is to let the student mimic the teacher's behavior by adding a strong congruent constraint on predictions [3, 15, 25] using KL divergence

$$L_{\text{KD}} = \frac{1}{n}\sum_{i=1}^{n} \tau^2 KL(\boldsymbol{p}_s^{\tau}, \boldsymbol{p}_t^{\tau}), \tag{1}$$

where τ is a relaxation hyperparameter (referred as temperature in [15]) to soften the output of teacher network, $\boldsymbol{p}^{\tau} = softmax(\frac{\boldsymbol{z}}{\tau})$. In several works [30, 21] the KL divergence is replaced by euclidean distance,

$$L_{\text{mimic}} = \frac{1}{n}\sum_{i=1}^{n} \|\boldsymbol{f}_s - \boldsymbol{f}_t\|_2^2. \tag{2}$$

Regardless of congruent constraint on final predictions [15], feature representations [30] or activations of hidden layer [25], these methods only focus on instance congruence while ignore the correlation between instances. Due to the gap (in capacity) between teacher and student, it's hard for a light-weight student to learn an identical mapping function from a cumbersome teacher by instance congruence. We argue that the correlation between instances is also vital for classification since it directly reflects how the teacher model the structure of different instances in embedded feature space.

3.3. Correlation Congruence

In this section, we present correlation congruence knowledge distillation (CCKD) in detail. Different from previous methods, CCKD considers not only the instance level congruence but also correlation congruence between instances. Figure 2 shows the overview of CCKD. CCKD consists of two part: instance congruence (KL divergence on predictions of teacher and student) and correlation congruence.

Let $\boldsymbol{F}_t$ and $\boldsymbol{F}_s$ represent the set of feature representations of teacher and student respectively,

$$\begin{aligned} \boldsymbol{F}_t &= matrix(\boldsymbol{f}_1^t, \boldsymbol{f}_2^t, ..., \boldsymbol{f}_n^t), \\ \boldsymbol{F}_s &= matrix(\boldsymbol{f}_1^s, \boldsymbol{f}_2^s, ..., \boldsymbol{f}_n^s). \end{aligned} \tag{3}$$

The feature $\boldsymbol{f}$ can be seen as a point in the embedded feature space. Without loss of generality, a mapping function is introduced as follow:

$$\psi : \boldsymbol{F} \rightarrow \boldsymbol{C} \in \mathbb{R}^{n \times n}. \tag{4}$$

where $\boldsymbol{C}$ is a correlation matrix. Each element in $\boldsymbol{C}$ represents the correlation between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ in embedding space, which is defined as

$$\boldsymbol{C}_{ij} = \varphi(\boldsymbol{f}_i, \boldsymbol{f}_j), \quad \boldsymbol{C}_{ij} \in \mathbb{R} \tag{5}$$

The function φ can be any correlation metric, and we will introduce three metrics for capturing the correlation between instances in the next section. Then, the correlation congruence can be formulated as follow:

$$\begin{aligned} L_{CC} &= \frac{1}{n^2}\|\psi(\boldsymbol{F}_t) - \psi(\boldsymbol{F}_s)\|_2^2 \\ &= \frac{1}{n^2}\sum_{i,j}(\varphi(\boldsymbol{f}_i^s, \boldsymbol{f}_j^s) - \varphi(\boldsymbol{f}_i^t, \boldsymbol{f}_j^t))^2. \end{aligned} \tag{6}$$

Then, the optimization goal of CCKD is to minimize the following loss function:

$$L_{CCKD} = \alpha L_{\text{CE}} + (1-\alpha) L_{\text{KD}} + \beta L_{\text{CC}}, \tag{7}$$

where L_{CE} is the cross-entropy loss, α and β are two hyperparamemters for balancing correlation congruence and instance congruence.

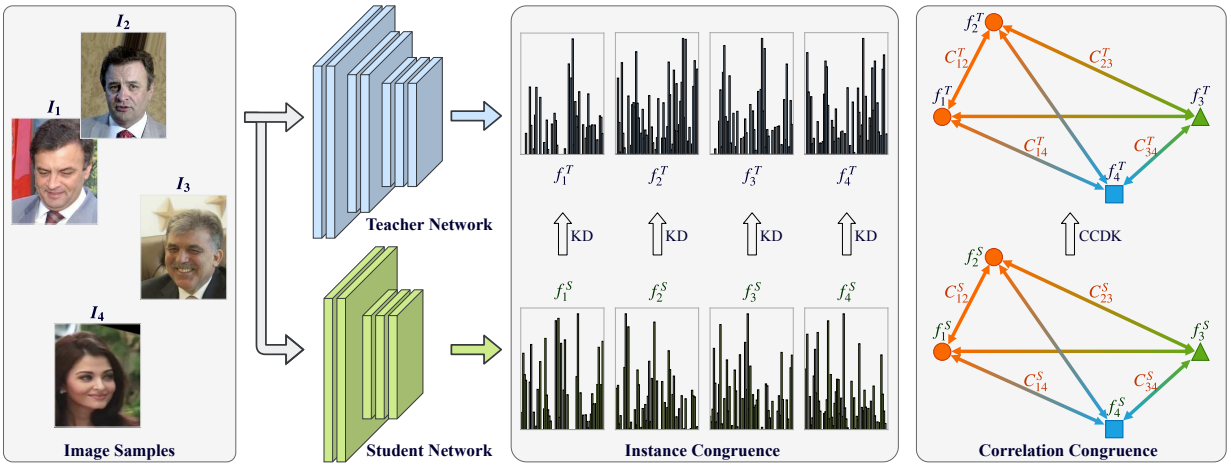


Figure 2: The overall framework of correlation congruence for knowledge distillation (T: teacher; S: teacher; f_i^T: teacher's output of i_{th} sample; f_i^S: student's output of i_{th} sample; C_i: correlation between i_{th} and j_{th} sample). CCKD aims to not only instance congruence but also correlation congruence between multiple instances.

3.4. Generalized kernel-based correlation

Capturing the complex correlations between instances is not easy due to a very high dimension in the embedded space [31]. In this section, we introduce kernel trick to capture the high order correlation between instances in the feature space.

Let $\boldsymbol{x}, \boldsymbol{y} \in \Omega$ represent two instances in feature space, and we introduce different mapping functions $k : \Omega \times \Omega \mapsto R$ as correlation metric, including:

1. naive MMD: $k(\boldsymbol{x}, \boldsymbol{y}) = \left|\frac{1}{n}\sum_i \boldsymbol{x}_i - \frac{1}{n}\sum_i \boldsymbol{y}_i\right|$;
2. Bilinear Pool: $k(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}^\top \cdot \boldsymbol{y}$;
3. Gaussian RBF: $k(\boldsymbol{x}, \boldsymbol{y}) = exp(-\frac{\|\boldsymbol{x}-\boldsymbol{y}\|_2^2}{2\delta^2})$;

MMD can reflect the distance between mean embeddings. Bilinear Pooling [22] can be seen as a naive 2_{th} order function, of which the correlation between two instances is computed by element-wise dot product. Gaussian RBF is a common kernel function whose value depends only on the euclidean distance from the origin space.

Comparing to naive MMD and Bilinear Pool, Gaussian RBF is more flexible and powerful in capturing the complex non-linear relationship between instances. Based on Gaussian RBF, the correlation mapping function ϕ can be computed by a kernel function $K : F \times F \in \mathbb{R}^{n\times n}$, where each element can be computed as

$$[k(\boldsymbol{F}, \boldsymbol{F})]_{ij} \approx \sum_{p=0}^{P} \alpha_p (\boldsymbol{F}_{i\cdot} \cdot \boldsymbol{F}_{j\cdot}^\top)^P. \tag{8}$$

which can be approximated by P-order Taylor series. Once specifying the kernel function, then the coefficient α_p is also confirmed. Each element $[k(\boldsymbol{F}, \boldsymbol{F})]_{ij}$ encodes the pairwise correlations between i_{th} and j_{th} features in $\boldsymbol{F}$. We take Gaussian RBF kernel function as an example, then

$$\begin{aligned}[k(\boldsymbol{F}, \boldsymbol{F})]_{ij} &= exp(-\gamma \left\|\boldsymbol{F}_i - \boldsymbol{F}_i\right\|^2) \\ &\approx \sum_{p=0}^{P} exp(-2\gamma)\frac{(2\gamma)^p}{p!}(\boldsymbol{F}_{i\cdot} \cdot \boldsymbol{F}_{j\cdot}^\top)^p.\end{aligned} \tag{9}$$

where γ is a tunable parameter.

3.5. Strategy for Mini-batch Sampler

Since the correlation between instances is computed in a mini-batch, a proper sampler is important for balancing the intra-class and inter-class correlation congruence. A simple strategy is uniformly at random sampler (UR-sampler), which would lead to such a situation that all examples come from different classes when the class number is large. Although it is an unbiased estimation for truth gradient of instance congruence, UR-sampler would result in a high biased estimate for the gradient of intra-class correlation.

To balance the intra-class and inter-class correlation congruence, we propose two strategies for mini-batch sampler: class-uniform random sampler (CUR-sampler) and superclass-uniorm random sampler (SUR-sampler). CUR-sampler samples by class and random selects fixed k number of examples for each sampled class (e.g. each batch consists of 6 class, and each class contains $k = 8$ examples, forming a 48 batch size). SUR-sampler is similar to CUR-sampler but different in that it samples examples by the superclass, a more soft form of the true class generated by clustering. To get the superclass of training examples, we first extract the feature using the teacher model, then use

the K-means to cluster. The superclass of example is defined as the cluster it belongs. Comparing to CUR-sampler, SUR-sampler is more flexible and tolerant for imbalance label since the superclass inflects the coarse structure of instances in embedded space.

3.6. Complexity analysis and implementation details

To cope with the mini-batch training, we compute the correlation in a mini-batch. Formula 9 involves the computation of a large pairwise matrix $b \times b$ (b is the batch size), and each element is approximated by p-order Taylor-series with p times dot product computation between two d dimension vectors. The total computation complexity is $O(pbd^2)$ in a mini-batch, and the extra space consumption is $O(b^2 + d^2)$ for storing the correlation matrix. Compared to huge parameters and computation for training a deep neural network, the time and computation consumption for correlation congruence can be ignored. Besides, since the correlation congruent constraint is added on embedding space, it only requires that the feature dimension of the student network is the same with the teacher. To cope with the mismatch dimension between teacher and student, a fully-connected layer with fixed-length dimension is added for both teacher and student network, which has minor influence on other methods in this paper.

4. Experiments

We evaluate CCKD on multiple tasks, including image classification tasks (CIFAR-100 and ImageNet-1K) and metric learning tasks (including MSMT17 dataset ReID and MegaFace for face recognition), and compare it with closely related works. Extensive experiments and analysis are conducted to delve into the correlation congruence knowledge distillation.

4.1. Experimental Settings

Network Architecture and Implementation Details Given the steady performance and efficiency computation, ResNet [12] and MobileNet [26] network are chosen in this work.

In the main experiments, we set the order $P = 2$, and compute Equation 9 in a mini-batch. For the networks in CIFAR-100 and ImageNet-1K, we add a fully-connected layer with 128-d output to form a sharing embedding space for teacher and student. The hyper-parameter α is set to zero, and correlation congruence scale $beta$ is set to 0.003, $\gamma = 0.4$. CUR-sampler is used for all the main experiments with $k = 4$.

On CIFar-100, ImageNet-1K and MSMT17, Original Knowledge distillation (KD) [15] and cross-entropy (CE) are chosen as the baselines. For face recognition, ArcFace loss [5] and $L2$-mimic loss [21, 23] are adopt. We compare CCKD with several state-of-the-art distillation related methods, including attention transfer (AT) [37], deel mutual learning (DML) [39] and conditional adversarial network (Adv) [35]. For attention transfer, we add it for last two blocks as suggested in [37]. For adversarial training, the discriminator consists of FC(128×64) + BN + ReLU + FC (64×2) + Sigmoid activation layers, and we adopt BinaryCrossEntropy loss to train it. All the networks and training procedures are implemented in PyTorch.

4.2. Classification Results on CIFAR-100

CIFAR-100 [20] consists of colored natural images with 32×32 size. There are 100 classes in CIFAR-100, each class contains 500 images in the train set and 100 images in the test set. We use the standard data augmentation scheme (flip/padding/random crop) that is widely used for this dataset, and normalize the input images using the channel means and standard deviations. We set the weight decay of student network to $1e-4$, batch size to 64, and use stochastic gradient descent with momentum. The starting learning rate is set as 0.1 and divided by ten at 80, 120, 160 epochs, totally 200 epochs. Top-1 and top-5 accuracy are adopted as a performance metric.

Table 1: Validation accuracy results on CIFAR-100. ResNet-110 is as teacher network, ResNet-20 and ResNet-14 as student networks. We keep the same training configuration for all the methods for fair comparasion.

method	resnet-20		resnet-14	
	top-1	top-5	top-1	top-5
CE	68.4	91.3	66.4	90.3
KD	70.8	92.4	68.3	90.7
DML	71.2	92.5	69.1	91.2
AT	71.0	92.4	68.6	91.1
Adv	70.5	92.1	68.1	90.6
CCKD	**72.4**	**92.9**	**70.2**	**92.0**

Table 1 summarizes the results of CIFAR-100. CCKD gets a 72.4% and 70.2% of top-1 accuracy for ResNet-20 and ResNet-14, and substantially surpasses the CE by 4.0% and 3.8%, 1.6% and 1.9% over KD. For the online distillation DML [39], we train target network (ResNet-14 and ResNet-20) collaboratively with ResNet-110, and evaluate performance of target network. Comparing to other SOTA methods, CCKD still significantly All the four distillation related methods surpass the original CE over 2%, which verifies the effectiveness of teacher-student methods.

4.3. Results on ImageNet-1K

ImageNet-1K [4] consists 1.28M training images and 50K testing images in total. We adopt the ResNet-50 [12]

as the teacher network, MobileNetV2 with 0.5 width multiplier as the student network. The data augmentation scheme for training images is the same as [12], and apply a center-crop at test time. All the images are normalized using the channel means and standard deviations. We set the weight decay of student network to $1e-4$, batch size to 1,024 (training on 16 TiTAN X, each with 64 batch size), and use stochastic gradient descent with momentum. The starting learning rate is set as 0.4, then divided by ten at 50, 80, 120 epochs, totally 150 epochs.

Table 2: Validation accuracy results on ImageNet 1K. The teacher network is ResNet-50, student network is MobileNetV2 with 0.5 width multiplier. We keep the same configuration for CE and other four student networks.

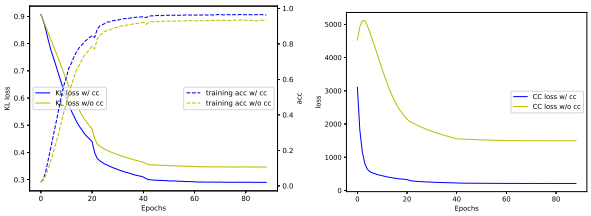
method	top-1 accuracy	top-5 accuracy
teacher	75.5	92.7
CE	64.2	85.4
KD	66.7	87.3
DML	65.3	86.1
Adv	66.8	87.3
AT	65.4	86.1
CCKD	**67.7**	**87.7**

For fair comparison, we keep the same configuration for all the methods. Table 2 summarizes the results on ImageNet 1K. CCKD gets a 67.7% Top-1 accuracy, which surpasses the cross-entropy by promoting 3.3. Compare with original KD[15], CCKD surpasses by 1.0 in top-1 accuracy. AT and DML perform worse than original KD. To our best knowledge, we have not found any works that successfully verify the effectiveness of KD on ImageNet-1K dataset. It has been reported in work [37] that KD struggles to work when the architecture and depth of student network are different from the teacher. But we found that by removing the dropout layer and using a proper temperature (T in [4,8]), the KD can surpass the student over 2.0%.

4.4. Person Re-Identification on MSMT17

Comparing to closed set classification, open set classification is more dependent on a good metric learning and more realistic scenario. We apply the proposed method to two open-set classification: person re-identification (ReID) and face recognition.

For ReID, we evaluate proposed method on MSMT17 [33]. It contains 180 hours of videos captured by 12 outdoor cameras, three indoor cameras under different seasons and time. There are 126,441 bounding boxes of 4,101 identities annotated. All the bounding boxes are split to the train set (32621 bounding boxes, 1041 identities), query set (11659 bounding boxes, 3060 identities), and gallery set (82161 bounding boxes). There is no intersection of identities between train set and query & gallery set. We train



(a) KL loss and training acc (b) CC loss

Figure 3: The curve of training loss and validation accuracy.

the networks on the train set and perform identification on the query and gallery set. Rank-1&5 and mean accuracy precision (mAP) are adopted as the performance metric.

ResNet-50 is used as the teacher network and ResNet-18 as student network. The dimension of the feature representation is set to 256. We set the weight decay to $5e-4$, batch size to 40, and use stochastic gradient descent with momentum. The learning rate is set as 0.0003, then divided by ten at 45, 60 epochs, totally 90 epochs.

Table 3: Validation accuracy results on MSMT17. The teacher network is ResNet-50, student network is Resnet-18.

method	pretrained?	rank-1	rank-5	mAP
teacher	yes	66.4	79	34.3
CE	no	32.4	49.0	14.2
DML-1	no	34.5	51.5	16.5
DML-2	yes	50.2	66.4	25.3
KD	no	56.8	72.3	28.3
AT	no	57.6	72.5	28.7
Adv	no	56.0	71.6	27.8
CCKD	no	**59.7**	**74.1**	**30.7**

Table 3 summarizes the results of MSMT17 with CCKD, as well as the comparison against other SOTA methods. For a fair comparison, all the distillation based methods (except DML) are trained without ImageNet-1K pretraining. For DML, both the results with/without ImageNet-1K pretraining are represented. It can be seen that the performance of the CCKD significantly surpasses KD and other SOTA KD-based methods, and promote the original KD by 3.1% for rank-1 accuracy and 2.4% for mAP. Without the guidance of the teacher, the student trained by cross-entropy only achieves 14.2% mAP, which is much lower than 28.3% of KD.

Figure 3 shows the training loss and accuracy of ResNet-18. It can be observed that although KL divergence loss after convergence is almost the same, the correlation congruence loss for CCKD is much lower than original KD, consequently results in higher performance.

4.5. Face recognition results on Megaface

Similar to ReID, face recognition is a classical metric learning problem. Learning a discriminative embedded

space is the key to get a powerful recognition model. Usually, thousands of identities (class) are required for training a well-performed recognition model. Empirical evidence shows that mimicking the feature layer with hint-based L2 loss can bring great improvement for small network [21, 23]. In this experiment, instead of using KD loss, we adopt the $L2$-mimic loss. MS-Celeb-1M [9] and IMDB-Face [32] are used as training datasets.

We choose MegaFace [19], a very popular benchmark, as the test set to evaluate the proposed method. MegaFace aims at the evaluation of face recognition algorithms at million-scale of distractors (people who are not in the testing set). We adopt the 1:N identification protocol in Megaface to evaluate the different methods. Rank-1 identification rate at a different number of distractors is used as an evaluation metric. We set weight decay to 5e-4, batch size to 1024, and use stochastic gradient descent with momentum. The learning rate is set as 0.1 and divided by ten at 50, 80, 100 epochs, 120 epochs in total. ResNet-50 is used as teacher network and MobileNetV2 with 0.5 width multiplier as student network.

Table 4: Results on Megaface. The teacher network is ResNet-50 trained on MsCeleb-1M [9] and IMDb-face [32] using ArcFace [5]. The student network is MobileNetV2 with a width multiplier=0.5. We keep the same training configuration for mimic, mimic with Adv and CCKD.

method	Rank-1 Identification rate at different distractors					
	ds=10^1	ds=10^2	ds=10^3	ds=10^4	ds=10^5	ds=10^6
teacher	99.76	99.66	99.58	99.49	99.23	98.15
student	99.20	96.37	91.49	84.45	75.60	65.91
mimic	99.63	98.73	97.25	94.39	89.60	83.01
mimic+Adv	99.64	98.80	97.43	94.81	90.52	84.13
CCKD	**99.66**	**99.07**	**97.93**	**95.76**	**91.99**	**86.29**

Table 4 shows the results on megaface. It can be observed that ArcFace loss, which is trained by only using pure one-hot labels, achieves 65.91% Rank-1 identification rate with 1M distractors. When guided by the teacher using $L2$-mimic loss, the student network can achieve 83.01%, promoting by 18.1%. This result shows that even a much small network can get a substantial improvement in performance when designing a proper target and optimization goal. By adding the constraints on correlations among instance, CCKD achieves 86.29% Rank-1 identification rate with 1M distractors, which surpasses the mimicking by 3.28% and 2.16% promotion over Adv [35].

4.6. Ablation Studies

Correlation Metrics. To explore the impact of different correlation metrics on CCKD, we evaluate three popular metrics, namely max mean discrepancy (MMD), Bilinear Pool, and Gaussian RBF. We approximate the Gaussian RBF by using 2-order Taylor series. MMD reflects the difference between instance pairs in mean embeddings. Bilinear Pool evaluates the similarity of instances pair, and we adopt identity matrix as the linear matrix. When the features are normalized to unit length, it is equal to the cosine similarity. Gaussian RBF is a common kernel function whose value depends only on the euclidean distance from the original space.

Table 5: Results on MSMT17 with different correlation methods, including MMD, Bilinear Pool and Gaussian RBF. The Gaussian RBF achieves the best result.

correlation metric	rank-1	rank-5	mAP
MMD	58.9	73.6	29.4
Bilinear	59.2	73.8	30.2
Gaussian RBF	59.6	74.0	30.4

Table 5 shows the results of MSMT17 with different correlation metrics. Gaussian RBF achieves the better performance comparing to MMD and Bilinear Pool, while MMD performs worst. So in the main experiments, we use the Gaussian RBF approximated by 2-order Taylor series. All three correlation matrics greatly surpass the original KD, which proves the effectiveness of correlation in knowledge distillation.

Order of Taylor series. To exploit the high order of correlations between instances, we expand the Gaussian RBF by Tarloy series to 1, 2, 3 -order respectively.

Table 6: Results on MSMT17 with different order ($p = 1, 2, 3$) Taylor series (mean of 3 runs).

Expand order	rank-1	rank-5	mAP
p=1	59.2	73.7	30.1
p=2	59.6	74.0	30.4
p=3	60.1	74.2	30.6

Table 6 summarizes the results on MSMT17 with approximated Gaussian RBF at different orders. It can be observed that 3-order is better than $1, 2$-order and 1-order performs worst. Generally speaking, expanding Gaussian RBF to high order can capture more complex correlations, and consequently achieves higher performance in knowledge distillation.

Impact of β. To exploit the impact of hyper-parameter β, we have tried different β. Table 7 shows the results under the different values of β, from which we can observe that CCKD is consistently superior to KD.

Impact of Different Sampler Strategies. To explore a proper sampler strategy, we evaluate the impacts of different sampler strategies including uniform random sampler (UR-sampler), class-uniform random sampler (CUR-sampler) and superclass-uniform random sampler (SUR-

Table 7: Results on MSMT17 under different β. (mean of 3 runs.)

β (10^{-3})	0 (KD)	1	2	3	4	5	10
rank1	56.8	58.7	58.9	59.4	59.8	59.5	59.1
rank5	72.3	73.9	74.1	74.4	74.9	74.7	74.4
mAP	28.3	30.3	30.6	30.8	31.4	31.3	30.9

sampler) on MSMT17 dataset. For SUR-sampler, the k-means is adopted, and the number of clusters is set to 1000 to generate superclass. For a fair comparison, the batch size is set to 40 for all three strategies, and we set different $k = 1, 2, 4, 8, 20$ both for CUR-sampler and SUR-sampler.

Table 8 summarizes the results. It can be observed that the sampler strategy have a great impact on performance. Both SUR-sampler and CUR-sampler are sensitive to the value of k, which plays a role in balancing the intra-class and inter-class correlation congruence. When given fixed batch size, a larger k means a smaller number of classes in a mini-batch. Both CUR-sampler and SUR-sampler become worse when $k = 8$ or above. A possible explanation is that a small number of classes in a mini-batch would result in a high bias estimation for the true gradient, while the SUR-sampler performs better than CUR-sampler in such bad cases. By selecting proper k (e.g., 2 or 4 in our experiments), Both CUR-sampler and SUR-sampler performs better than UR-sampler.

Table 8: Results on MSMT17 with different batch sampler strategies. The teacher network is ResNet-50 and the student network is ResNet-18.

sampler	rank-1	rank-5	mAP
UR-sampler	57.2	72.3	28.6
CUR-sampler(k=1)	57.4	72.4	28.8
CUR-sampler(k=2)	58.9	73.6	29.4
CUR-sampler(k=4)	**59.7**	74.1	30.2
CUR-sampler(k=8)	55.7	71.8	29.1
CUR-sampler(k=20)	24.7	40.9	10.7
SUR-sampler(k=1)	56.2	72.2	29.4
SUR-sampler(k=2)	58.3	73.9	29.9
SUR-sampler(k=4)	59.6	**75.0**	**31.1**
SUR-sampler(k=8)	56.2	72.2	29.4
SUR-sampler(k=20)	30.1	47.7	13.7

4.7. Analyze

To delving into essence beyond results, we perform analysis based on visualization. We count the cosine similarities of intra-class instances and inter-class instances on MSMT17 since it is a common metric for open-set recognition. Figure 4 shows the heatmaps of cosine similarities. The top row shows intra-class instances, and the bottom row shows inter-class instances from two different identities. Each cell relates to the cosine similarity between the corresponding instance pair.

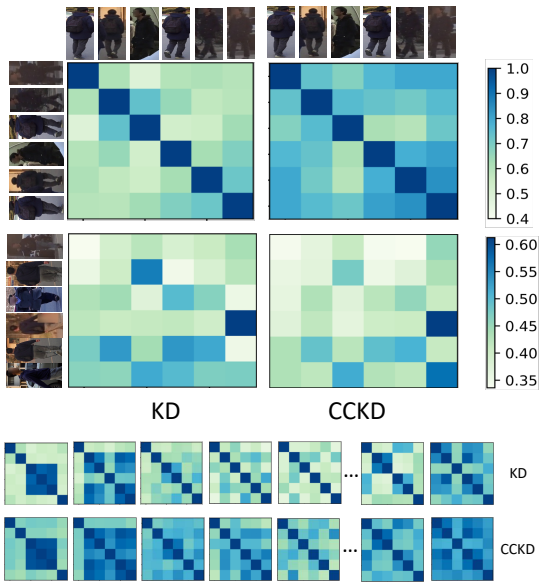


Figure 4: The heatmaps of cosine similarities between instances pairs. The top row shows intra-class similarities and the middle row shows inter-class similarities between two identities. More intra-class heatmap are showed in bottom two rows. (best viewed in color)

It can be observed that cosine similarity between intra-class instances of CCKD is larger than KD overall, which means a more cohesion of intra-class instances in embedding space, although there is not much difference between CCKD and KD in inter-class cosine similarity. It seems that CCKD can help the student to learn a more discriminative embedding space. While CCKD by considering the correlation congruence between instances, consequently getting better performance.

5. Conclusions

In this paper, we propose a new distillation framework named correlation congruence knowledge distillation (CCKD), which considers not only instance information but also correlation information between instances when transferring knowledge. To better capture correlation, a generalized method based on the Taylor series expansion of kernel function is proposed. To further improve the CCKD, two new mini-batch sampler strategies are proposed. Extensive experiments on four representative tasks show that the proposed approach can significantly promote the performance of the student network.

Acknowledgement

This work is sponsored in part by the National Key R&D Program of China under Grant (No. 2018YFB0204300) and the National Natural Science Foundation of China under Grant (No. 61872376).

References

[1] Rohan Anil, Gabriel Pereyra, Alexandre Passos, Robert Ormandi, George E Dahl, and Geoffrey E Hinton. Large scale distributed neural network training through online distillation. *arXiv preprint arXiv:1804.03235*, 2018.

[2] Jimmy Ba and Rich Caruana. Do deep nets really need to be deep? In *Advances in neural information processing systems*, pages 2654–2662, 2014.

[3] Cristian Buciluǎ, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, pages 535–541, New York, NY, USA, 2006. ACM.

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. Ieee, 2009.

[5] Jiankang Deng, Jia Guo, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. 2018.

[6] Emily L Denton, Wojciech Zaremba, Joan Bruna, Yann LeCun, and Rob Fergus. Exploiting linear structure within convolutional networks for efficient evaluation. In *Advances in neural information processing systems*, pages 1269–1277, 2014.

[7] Simon S Du, Xiyu Zhai, Barnabas Poczos, and Aarti Singh. Gradient descent provably optimizes over-parameterized neural networks. *arXiv preprint arXiv:1810.02054*, 2018.

[8] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[9] Yandong Guo, Lei Zhang, Yuxiao Hu, Xiaodong He, and Jianfeng Gao. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. pages 87–102, 2016.

[10] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015.

[11] Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. In *Advances in neural information processing systems*, pages 1135–1143, 2015.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[13] Byeongho Heo, Minsik Lee, Sangdoo Yun, and Jin Young Choi. Improving knowledge distillation with supporting adversarial samples. *arXiv preprint arXiv:1805.05532*, 2018.

[14] Byeongho Heo, Minsik Lee, Sangdoo Yun, and Jin Young Choi. Knowledge distillation with adversarial samples supporting decisionboundary. *arXiv preprint arXiv:1805.05532*, 2018.

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

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

[17] Itay Hubara, Matthieu Courbariaux, Daniel Soudry, El Yaniv Ran, and Yoshua Bengio. Quantized neural networks: Training neural networks with low precision weights and activations. *Journal of Machine Learning Research*, 18, 2016.

[18] Max Jaderberg, Andrea Vedaldi, and Andrew Zisserman. Speeding up convolutional neural networks with low rank expansions. *arXiv preprint arXiv:1405.3866*, 2014.

[19] Ira Kemelmachershlizerman, Steven M. Seitz, Daniel Miller, and Evan Brossard. The megaface benchmark: 1 million faces for recognition at scale. In *Computer Vision and Pattern Recognition*, pages 4873–4882, 2016.

[20] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

[21] Quanquan Li, Shengying Jin, and Junjie Yan. Mimicking very efficient network for object detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7341–7349. IEEE, 2017.

[22] Tsung-Yu Lin, Aruni RoyChowdhury, and Subhransu Maji. Bilinear cnn models for fine-grained visual recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1449–1457, 2015.

[23] Ping Luo, Zhenyao Zhu, Ziwei Liu, Xiaogang Wang, Xiaoou Tang, et al. Face model compression by distilling knowledge from neurons. In *AAAI*, pages 3560–3566, 2016.

[24] Pavlo Molchanov, Stephen Tyree, Tero Karras, Timo Aila, and Jan Kautz. Pruning convolutional neural networks for resource efficient inference. 2016.

[25] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2014.

[26] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4510–4520, 2018.

[27] Bharat Bhusan Sau and Vineeth N Balasubramanian. Deep model compression: Distilling knowledge from noisy teachers. *arXiv preprint arXiv:1610.09650*, 2016.

[28] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[29] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[30] Gregor Urban, Krzysztof J. Geras, Samira Ebrahimi Kahou, Ozlem Aslan, Shengjie Wang, Rich Caruana, Abdelrahman Mohamed, Matthai Philipose, and Matt Richardson. Do deep convolutional nets really need to be deep and convolutional? *Nature*, 521, 2016.

[31] Greg Ver Steeg and Aram Galstyan. Discovering structure in high-dimensional data through correlation explanation. In *Advances in Neural Information Processing Systems*, pages 577–585, 2014.

[32] Fei Wang, Liren Chen, Cheng Li, Shiyao Huang, Yanjie Chen, Chen Qian, and Chen Change Loy. The devil of face recognition is in the noise. *arXiv preprint arXiv:1807.11649*, 2018.

[33] Longhui Wei, Shiliang Zhang, Wen Gao, and Qi Tian. Person transfer gan to bridge domain gap for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 79–88, 2018.

[34] Jiaxiang Wu, Leng Cong, Yuhang Wang, Qinghao Hu, and Jian Cheng. Quantized convolutional neural networks for mobile devices. In *Computer Vision and Pattern Recognition*, pages 4820–4828, 2016.

[35] Zheng Xu, Yen-Chang Hsu, and Jiawei Huang. Training shallow and thin networks for acceleration via knowledge distillation with conditional adversarial networks. 2018.

[36] Junho Yim, Donggyu Joo, Jihoon Bae, and Junmo Kim. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, 2017.

[37] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. 2016.

[38] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. 2017.

[39] Ying Zhang, Tao Xiang, Timothy M Hospedales, and Huchuan Lu. Deep mutual learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4320–4328, 2018.