Embedding Transfer with Label Relaxation for Improved Metric Learning

Sungyeon Kim[1] Dongwon Kim[1] Minsu Cho[1,2] Suha Kwak[1,2]
Dept. of CSE, POSTECH[1] Graduate School of AI, POSTECH[2]
{sungyeon.kim, kdwon, mscho, suha.kwak}@postech.ac.kr

Abstract

This paper presents a novel method for embedding transfer, a task of transferring knowledge of a learned embedding model to another. Our method exploits pairwise similarities between samples in the source embedding space as the knowledge, and transfers them through a loss used for learning target embedding models. To this end, we design a new loss called relaxed contrastive loss, which employs the pairwise similarities as relaxed labels for inter-sample relations. Our loss provides a rich supervisory signal beyond class equivalence, enables more important pairs to contribute more to training, and imposes no restriction on manifolds of target embedding spaces. Experiments on metric learning benchmarks demonstrate that our method largely improves performance, or reduces sizes and output dimensions of target models effectively. We further show that it can be also used to enhance quality of self-supervised representation and performance of classification models. In all the experiments, our method clearly outperforms existing embedding transfer techniques.

1. Introduction

Deep metric learning aims to learn an embedding space where samples of the same class are grouped tightly together. Such an embedding space has played important roles in many tasks including image retrieval [19, 20, 29, 40, 41], few-shot learning [35, 39, 42], zero-shot learning [3, 58], and self-supervised representation learning [4, 14, 44]. In these tasks, the performance and efficiency of models rely heavily on the quality and dimension of their learned embedding spaces. To obtain high-quality and compact embedding spaces, previous methods have proposed new metric learning losses [19, 29, 40, 41, 46, 52], advanced sampling strategies [13, 22, 47, 49], regularization techniques [18, 28], or ensemble models [21, 30, 31].

For the same purpose, we study transferring knowledge of a learned embedding model (source) to another (target), which we call *embedding transfer*. This task can be considered as a variant of *knowledge distillation* [16] that focuses on metric learning instead of classification. The knowledge captured by the source embedding model can provide rich information beyond class labels such as intra-class variations and degrees of semantic affinity between samples. Given a proper way to transfer the knowledge, embedding transfer enables us to improve the performance of target embedding models or compress them, as knowledge distillation does for classification models [11, 16, 36, 50, 55].

Existing methods for embedding transfer extract knowledge from a source embedding space in forms of probability distributions of samples [33], their geometric relations [32, 53], or the rank of their similarities [6]. The knowledge is then transferred by forcing target models to approximate those extracted patterns directly in their embedding spaces. Although these methods shed light on the effective yet less explored approach to enhancing the performance of metric learning, there is still large room for further improvement. In particular, they fail to utilize detailed inter-sample relations in the source embedding space [6, 33] or blindly accept the transferred knowledge without considering the importance of samples [32, 53].

This paper presents a new embedding transfer method that overcomes the above limitations. Our method defines the knowledge as pairwise similarities between samples in a source embedding space. Pairwise similarities are useful to characterize an embedding space in detail, thus have been widely used for learning embedding spaces [12, 37, 40, 46] and identifying underlying manifolds of data [9, 43]. Also, they capture detailed inter-sample relations, which are missing in probability distributions [33] and the rank of similarities [6] used as knowledge in previous work.

To transfer the knowledge effectively, we propose a new loss, called *relaxed contrastive loss*, that is used for learning target embedding models with the knowledge in the form of pairwise similarities. Our loss utilizes the pairwise similarities as *relaxed labels* of inter-sample relations, unlike conventional metric learning losses that rely on binary labels indicating class equivalence between samples (*i.e.*, 1 if two samples are of the same class and 0 otherwise) as supervision. By replacing the binary labels with the pairwise similarities, our loss can provide rich supervision beyond

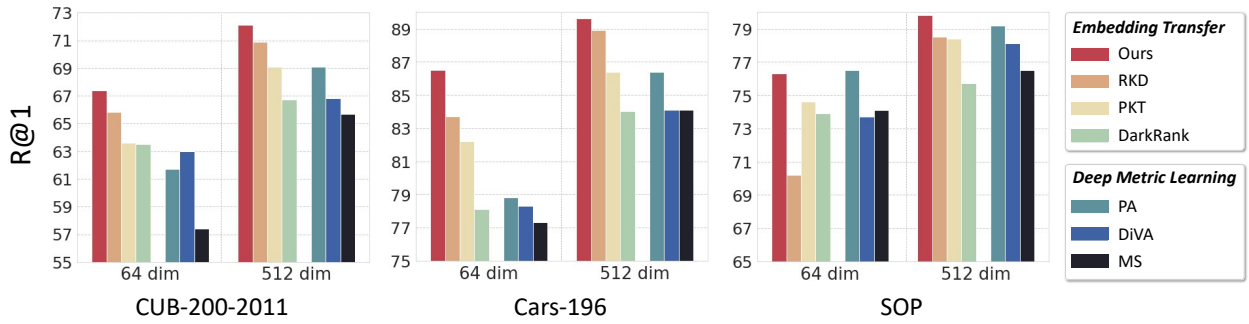

Figure 1. Accuracy in Recall@1 on the three standard benchmarks for deep metric learning. All embedding transfer methods adopt Proxy-Anchor (PA) [19] with 512 dimension as the source model. Our method achieves the state of the art when embedding dimension is 512, and is as competitive as recent metric learning models even with a substantially smaller embedding dimension. In all experiments, it is superior to other embedding transfer techniques. More results can be found in Table 1 and 2.

what the binary labels offer, such as the degree of similarity and hardness of a pair of training samples.

Specifically, the proposed loss pushes apart or pulls together a pair of samples in a target embedding space following the principle of the original contrastive loss [12], but the semantic similarity of the pair given by the knowledge controls the strength of pushing and pulling. Also, we reveal that the loss lets more important pairs contribute more to learning the target embedding model, thus resolves the limitation of previous methods that treat samples equally during transfer [32, 53]. In addition to the use of relaxed relation labels, we further modify the loss so that it does not impose any restriction on the manifold of target embedding space, unlike conventional losses that enforce target embedding spaces ℓ_2 normalized. This modification enables to utilize given embedding dimensions more effectively and provides extra performance improvement.

The efficacy of the proposed method is first demonstrated on public benchmark datasets for deep metric learning. Our method substantially improves image retrieval performance when the target model has the same architecture as its source counterpart, and greatly reduces the size and embedding dimension of the target model with a negligible performance drop when the target model is smaller than the source model, as shown in Fig. 1. We also show that our method enhances the quality of self-supervised representation through self embedding transfer and the performance of classification models in the knowledge distillation setting. In all the experiments, our method outperforms existing embedding transfer techniques [6, 32, 33].

2. Related Work

Deep metric learning. Deep metric learning is an approach to learning embedding spaces using class labels. Previous work in this field has developed loss functions for modeling inter-sample relations based on class labels and reflecting them on the learned embedding spaces. Contrastive loss [7, 12] pulls a pair of samples together if their class labels are the same and pushes them away otherwise. Triplet loss [37, 45] takes a triplet of anchor, positive, and negative as input, and makes the anchor-positive distance smaller than the anchor-negative distance. The idea of pushing and pulling a pair is extended to consider higher order relations in recently proposed losses [40, 41, 46]. Meanwhile, self-supervised representation learning has been greatly advanced by leveraging pairwise relations between data as in deep metric learning. For example, MoCo [5, 14] and SimCLR [4] pull embedding vectors of the same image closer and push those of different images away. Since these approaches to learning embedding spaces demand binary relations, *i.e.*, the equality of classes or identities, they cannot be used directly for transferring knowledge of an embedding space that is not binary.

Knowledge distillation. Knowledge distillation means a technique that transfers knowledge of a source model to a target model; embedding transfer can be regarded as its variant focusing on metric learning. A seminal work by Hinton *et al.* [16] achieves this goal by encouraging the target model to imitate class logits of the source model, and has been extended to transfer various types of knowledge of the source model [1, 36, 50, 55]. Knowledge distillation has been employed for various purposes including model compression [16, 36, 50, 55], cross-modality learning [44], and network regularization [54] as well as performance improvement [11]. In terms of target task, however, it has been applied mostly to classification; only a few methods introduced in the next paragraph study transferring knowledge of embedding spaces, *i.e.*, embedding transfer.

Embedding transfer. Early approaches in this area extract and transfer the rank of similarities between samples [6] and probability distributions of their similarities [33] in the source embedding spaces. Unfortunately, these methods have trouble capturing elaborate relations between samples. On the other hand, recently proposed methods utilize geo-

metric relations between samples like distances and angles as the knowledge to take fine details of the source embedding space into account [32, 53]. However, they let the target model blindly accept the knowledge without considering the relative importance of samples, leading to less effective embedding transfer. Our method overcomes the aforementioned limitations: It makes use of rich pairwise similarities between samples as the knowledge, and can take relative importance of samples into account when transferring the knowledge.

3. Proposed Method

This section first introduces the problem formulation of embedding transfer, then reviews briefly the original contrastive loss [12] and describes the derivation of the relaxed contrastive loss in detail. It also discusses the effect of label relation with empirical evidences.

3.1. Problem Formulation of Embedding Transfer

Embedding transfer is the task of transferring knowledge from a source embedding model s to a target embedding model t. Let $f^s : \mathcal{X} \to \mathcal{Z}^s$ and $f^t : \mathcal{X} \to \mathcal{Z}^t$ denote the two embedding models, which are mapping functions from the same data space $\mathcal{X}$ to their own embedding spaces. The goal of embedding transfer is to transfer knowledge captured in $\mathcal{Z}^s$ to $\mathcal{Z}^t$ for various purposes like performance enhancement, embedding dimension reduction, and embedding model compression.

3.2. Revisiting Original Contrastive Loss

Contrastive loss [12] is one of the most representative losses for learning semantic embedding by leveraging pairwise relations of samples. Let $f_i := f(x_i)$ be the embedding vector of input data x_i produced by the embedding network f, and $d_{ij} := ||f_i - f_j||_2$ denote the Euclidean distance between embedding vectors f_i and f_j. The contrastive loss is then formulated as

$$\mathcal{L}(X) = \underbrace{\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{n} y_{ij} d_{ij}^2}_{\textit{attracting}} + \underbrace{\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{n} (1 - y_{ij}) \left[\delta - d_{ij}\right]_+^2}_{\textit{repelling}}, \tag{1}$$

where X is a batch of embedding vectors, n is the number of samples in the batch, δ is a margin, and $[\cdot]_+$ denotes the hinge function. Also, y_{ij} is the binary label indicating the class equivalence between the pair of samples (i, j): $y_{ij} = 1$ if the pair is of the same class (*i.e.*, positive pair), and 0 otherwise (*i.e.*, negative pair). Note that all embedding vectors are l_2 normalized to prevent the margin from becoming trivial. This loss consists of two constituents, an attracting term and a repelling term. In the embedding space, the attracting term forces positive pairs to be closer, and the repelling term encourages to push negative pairs apart beyond the margin.

The gradient of the loss with respect to d_{ij} is given by

$$\frac{\partial \mathcal{L}(X)}{\partial d_{ij}} = \begin{cases} \frac{2}{n} d_{ij}, & \text{if } y_{ij} = 1, \\ \frac{2}{n}(d_{ij} - \delta), & \text{else if } y_{ij} = 0 \text{ and } d_{ij} < \delta, \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

As shown in Eq. (2), the magnitude of the gradient increases as the distance of a positive pair increases or the distance of a negative pair decreases. When the distance of a negative pair is larger than the margin δ, the gradient becomes 0.

3.3. Relaxed Contrastive Loss

The basic idea of the relaxed contrastive loss is to pull or push a pair of samples in the target embedding space according to their semantic similarity captured in the source embedding space as knowledge. Unlike the original contrastive loss, it relaxes the binary labels indicating class equivalence relations using pairwise similarities given in the transferred knowledge.

This idea can be implemented simply by replacing y_{ij} in Eq. (1) with the semantic similarity of x_i and x_j in the source embedding space. The loss then becomes a linear combination of the attracting and repelling terms in which their weights (*i.e.*, relaxation of y_{ij}) are proportional to the semantic similarities. Specifically, it is formulated as

$$\mathcal{L}(X) = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}^s {d_{ij}^t}^2 + \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{n} (1 - w_{ij}^s) \left[\delta - d_{ij}^t\right]_+^2, \tag{3}$$

where w_{ij}^s is the weight derived from the semantic similarities in the source embedding space, $f_i^t := f^t(x_i) \in \mathbb{R}^d$ indicates the embedding vector of input x_i produced by f^t, and d_{ij}^t is Euclidean distance between target embedding vectors f_i^t and f_j^t. For computing the weight terms, we employ a Gaussian kernel based on the Euclidean distance as follows:

$$w_{ij}^s = \exp\left(-\frac{||f_i^s - f_j^s||_2^2}{\sigma}\right) \in [0, 1], \tag{4}$$

where σ is kernel bandwidth $f_i^s := f^s(x_i)$ indicates the embedding vector of input x_i given by the f^s, and $||\cdot||_2$ denotes l_2 norm of vector.

Eq. (3) shows that the strength of pulling or pushing embedding vectors is now controlled by the weights in the new loss function. In the target embedding space, a pair of samples that the source embedding model regards more similar attract each other more strongly while those considered more dissimilar are pushed more heavily out of the margin

δ. This behavior of the loss can be explained through its gradient, which is given by

$$\frac{\partial \mathcal{L}(X)}{\partial d_{ij}^t} = \begin{cases} \frac{2}{n}\{d_{ij}^t - \delta(1 - w_{ij}^s)\}, & \text{if } d_{ij}^t < \delta, \\ \frac{2}{n} w_{ij}^s d_{ij}^t, & \text{otherwise.} \end{cases} \tag{5}$$

Unlike the original one, the aspect of our loss gradient depends on the transferred knowledge w_{ij}^s, thus the force of pushing a pair (i, j) apart and that of pulling them together are determined by both d_{ij}^t and w_{ij}^s. In the ideal case, d_{ij}^t will converge to $\delta(1 - w_{ij}^s)$, which is the semantic dissimilarity scaled by δ, and where the two forces are balanced.

This aspect of gradient also differentiates our method from the previous arts that imitate the knowledge through regression losses [32, 53]. As illustrated in Fig. 2, the proposed loss rarely cares about a pair (i, j) when its distance is large in both the source and target spaces, *i.e.*, $w_{ij}^s \approx 0$ and $d_{ij}^t > \delta$, as its loss gradient is close to 0. This behavior can be interpreted as that our loss disregards less important pairs to focus on more important ones. Recall that what we expect from a learned embedding space is that *nearby* samples are semantically similar in the space; if the distance of a semantically dissimilar pair is sufficiently large, it does not impair such a quality of the embedding space and can be regarded as less important consequently. On the other hand, the previous methods using regression losses handle samples equivalently without considering their importance [32, 53], leading to suboptimal results.

The loss in Eq. (3) takes advantage of the rich semantic information of the source embedding space in a flexible and effective manner, but it still has a problem to be resolved: It imposes a restriction on the manifold of the target space since it demands l_2 normalization of the embedding vectors to prevent the divergence of their magnitudes and to keep the margin non-trivial, as in the original contrastive loss. To resolve this issue, we replace the pairwise distances of the loss in Eq. (3) with their *relative* versions, then the final form of the relaxed contrastive loss is given by

$$\begin{aligned} \mathcal{L}(X) = & \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}^s \left(\frac{d_{ij}^t}{\mu_i}\right)^2 \\ & + \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{n} (1 - w_{ij}^s)\left[\delta - \frac{d_{ij}^t}{\mu_i}\right]_+^2, \\ & \text{where } \mu_i = \frac{1}{n}\sum_{k=1}^{n} d_{ij}^t. \end{aligned} \tag{6}$$

The relative distance between f_i^t and f_j^t is their pairwise distance divided by μ_i, the average distance of all pairs associated with f_i^t in the batch. Since scales of pairwise distances are roughly canceled in their relative versions, the above loss can alleviate the aforementioned normalization issue. Thus, although the source embedding space is limited to the surface of unit hypersphere due to the l_2 normalization, the target embedding model can exploit the entire space of $\mathbb{R}^d$ with no restriction on its manifold. We found empirically that this advantage helps improve performance of target embedding models and reduce their embedding dimensions effectively.

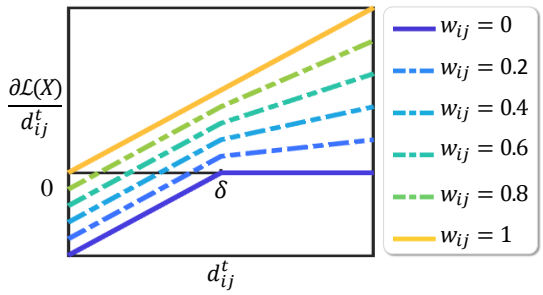

Figure 2. Gradients of relaxed contrastive loss versus pairwise distance given different weights.

3.4. Discussion on Label Relaxation

Label relaxation allows our loss to exploit rich information such as degree of similarity between samples, within-class variation, and between-class affinity, all of which cannot be offered by the binary inter-sample relations. To demonstrate this property empirically, we in Fig. 3 enumerate image pairs with top-5 and bottom-5 normalized weights, *i.e.*, w_{ij}^s of Eq. (4). As shown in the figure, pairs exhibiting more similar poses or backgrounds have higher weights even in the same class while those of different classes showing large appearance variations are assigned low weights.

Label relaxation thus improves generalization of target models by providing such rich and diverse supervisory signals, in contrast to the binary labels which only allow the model to learn to discriminate different classes and lead to degraded performance on unseen classes consequently. This is demonstrated by evaluating two embedding models trained by the relaxed contrastive loss and its unrelaxed version using y_{ij} instead of w_{ij}^s in Eq. (6), respectively. Fig. 4 compares the performance of the two models on the training and test splits of the Cars-196 dataset [23]. As shown in the figure, relaxed contrastive loss helps the model generalize well to unseen test data while the model trained by the unrelaxed version is quickly overfitted to training data.

Our label relaxation method is independent of loss functions, thus can be integrated with other metric learning losses based on pairwise relations of samples. Relaxed contrastive loss is yet chosen as our loss due to its simplicity, interpretability, and superior performance. We have also applied the same method to Multi-Similarity (MS) loss [46], and observed that relaxed MS loss achieves comparable per-

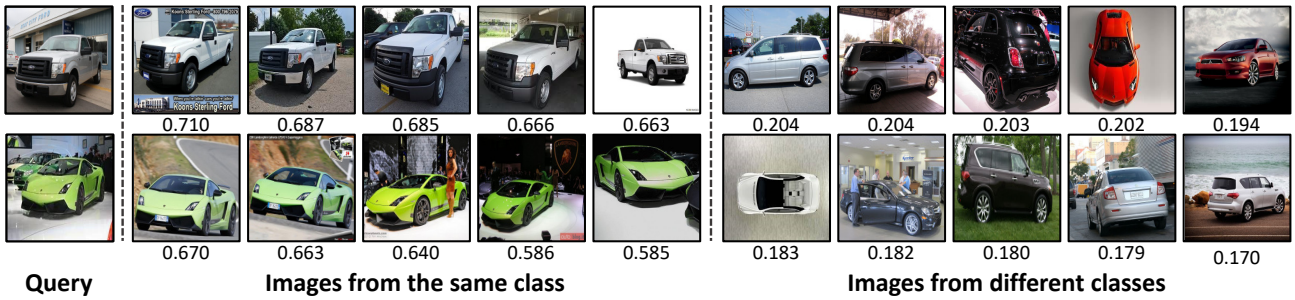

Figure 3. Image pairs sorted by their normalized weights of Eq. (4) on the Cars-196 dataset.

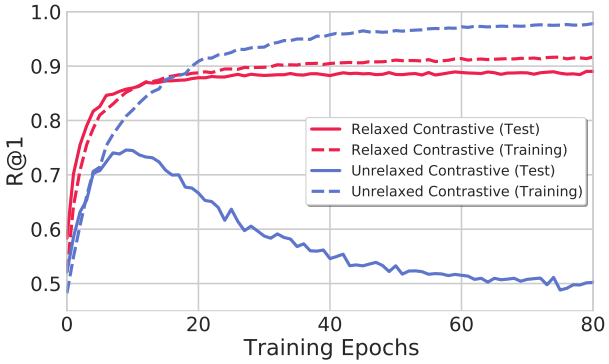

Figure 4. Accuracy in Recall@1 versus epochs on the Cars-196 dataset [23]. The dotted and solid lines represent training and test accuracy, respectively.

formance but demands more hyper-parameters and careful tuning of them. More analysis and comparisons are given in the supplementary material.

4. Experiments

This section demonstrates the effectiveness of embedding transfer by our method in three different tasks, deep metric learning, self-supervised representation learning, and knowledge distillation for classification.

4.1. Deep Metric Learning

On standard benchmarks for metric learning, we evaluate and compare target embedding models trained by embedding transfer methods, RKD [32], PKT [33], and DarkRank [6] as well as ours. These methods train target models solely by embedding transfer losses; no other supervision is required for the target model. In addition, three knowledge distillation techniques, FitNet [36], Attention [55], and CRD [44] are also evaluated on the same datasets to examine their effectiveness for embedding transfer.[1] In this case, knowledge distillation losses are coupled with a metric learning loss since they extract knowledge from intermediate layers of the source model and are not directly aware of its embedding space consequently.

The experiments are conducted in the following three settings by varying the type of target model. *(i) Self-transfer for performance improvement*: Transfer to a model with the same architecture and embedding dimension. *(ii) Dimensionality reduction*: Transfer to the same architecture with a lower embedding dimension. *(iii) Model compression*: Transfer to a smaller network with a lower embedding dimension.

4.1.1 Setup

Datasets and evaluation. Target models are evaluated in terms of image retrieval performance on the CUB-200-2011 [48], Cars-196 [23] and SOP datasets [41]. Each dataset is split into training and test sets following the standard setting presented in [41]. As a performance measure, we adopt Recall@K that counts how many queries have at least one correct sample among their K nearest neighbors in learned embedding spaces.

Source and target embedding networks. For the *self-transfer* and *dimensionality reduction* experiments, we employ BatchNorm Inception [17] with 512 output dimension as the source model. Target models for the two settings basically have the same architecture as the source model, but for *dimensionality reduction*, the output dimension is reduced to 64. On the other hand, in the *model compression* experiment, we adopt ResNet50 [15] with 512 output dimension as the source model and ResNet18 [15] with 128 output dimension as the target model. In all the three settings, the source models are trained by proxy-anchor loss [19] with l_2 normalization of embedding vectors, while the target models are pre-trained for the ImageNet classification task [10] and have no l_2 normalization applied.

Implementation details. We train all the models using the AdamW optimizer [26] with the cosine learning decay [25] and initial learning rate of 10^{-4}. They are learned for 90 epochs in the CUB-200-2011 and Cars-196 datasets, and 150 epochs on the SOP dataset. Training images are randomly flipped horizontally and cropped to 224×224, and test images are center-cropped after being resized to 256×256. Further, we generate two different views of each

[1] Knowledge distillation methods relying on classification logits cannot be applied to our task where source model has no classification layer.

	Recall@K		CUB-200-2011			Cars-196			SOP		
			1	2	4	1	2	4	1	10	100
(a)	*Source*: PA [19]	BN^{512}	69.1	78.9	86.1	86.4	91.9	95.0	79.2	90.7	96.2
	FitNet [36]	BN^{512}	69.9	79.5	86.2	87.6	92.2	95.6	78.7	90.4	96.1
	Attention [55]	BN^{512}	66.3	76.2	84.5	84.7	90.6	94.2	78.2	90.4	96.2
	CRD [44]	BN^{512}	67.7	78.1	85.7	85.3	91.1	94.8	78.1	90.2	95.8
	DarkRank [6]	BN^{512}	66.7	76.5	84.8	84.0	90.0	93.8	75.7	88.3	95.3
	PKT [33]	BN^{512}	69.1	78.8	86.4	86.4	91.6	94.9	78.4	90.2	96.0
	RKD [32]	BN^{512}	70.9	80.8	87.5	88.9	93.5	96.4	78.5	90.2	96.0
	Ours	BN^{512}	**72.1**	**81.3**	**87.6**	**89.6**	**94.0**	**96.5**	**79.8**	**91.1**	**96.3**
(b)	*Source*: PA [19]	BN^{512}	69.1	78.9	86.1	86.4	91.9	95.0	79.2	90.7	96.2
	FitNet [36]	BN^{64}	62.3	73.8	83.0	81.2	87.7	92.5	**76.6**	**89.3**	**95.4**
	Attention [55]	BN^{64}	58.3	69.4	79.1	79.2	86.7	91.8	76.3	89.2	95.4
	CRD [44]	BN^{64}	60.9	72.7	81.7	79.2	87.2	92.1	75.5	88.3	95.3
	DarkRank [6]	BN^{64}	63.5	74.3	83.1	78.1	85.9	91.1	73.9	87.5	94.8
	PKT [33]	BN^{64}	63.6	75.8	84.0	82.2	88.7	93.5	74.6	87.3	94.2
	RKD [32]	BN^{64}	65.8	76.7	85.0	83.7	89.9	94.1	70.2	83.8	92.1
	Ours	BN^{64}	**67.4**	**78.0**	**85.9**	**86.5**	**92.3**	**95.3**	76.3	88.6	94.8
(c)	*Source*: PA [19]	$R50^{512}$	69.9	79.6	88.6	87.7	92.7	95.5	80.5	91.8	98.8
	FitNet [36]	$R18^{128}$	61.0	72.2	81.1	78.5	86.0	91.4	76.7	89.4	95.5
	Attention [55]	$R18^{128}$	61.0	71.7	81.5	78.6	85.9	91.0	76.4	89.3	95.5
	CRD [44]	$R18^{128}$	62.8	73.8	83.2	80.6	87.9	92.5	76.2	88.9	95.3
	DarkRank [6]	$R18^{128}$	61.2	72.5	82.0	75.3	83.6	89.4	72.7	86.7	94.5
	PKT [33]	$R18^{128}$	65.0	75.6	84.8	81.6	88.8	93.4	76.9	89.2	95.5
	RKD [32]	$R18^{128}$	65.8	76.3	84.8	84.2	90.4	94.3	75.7	88.4	95.1
	Ours	$R18^{128}$	**66.6**	**78.1**	**85.9**	**86.0**	**91.6**	**95.3**	**78.4**	**90.4**	**96.1**

Table 1. Image retrieval performance of embedding transfer and knowledge distillation methods in the three different settings: (a) Self-transfer, (b) dimensionality reduction, and (c) model compression. Embedding networks of the methods are denoted by abbreviations: BN–Inception with BatchNorm, R50–ResNet50, R18–ResNet18. Superscripts indicate embedding dimensions of the networks.

image in a batch by the random augmentations; details and effects of this augmentation strategy are described in the supplementary material. We set both δ and σ in our loss to 1 for all the experiments. For knowledge distillation, Proxy-Anchor loss [19] is coupled with distillation losses using the same weight.

4.1.2 Results

The proposed method is compared to the embedding transfer and knowledge distillation methods in terms of performance of target embedding models on the three benchmark datasets in Table 1. Its records are also compared with those of state-of-the-art metric learning methods on the same datasets in Table 2.

In the *self-transfer* setting (Table 1(a)), the proposed method notably improves retrieval performance and clearly surpasses the state of the art on all the datasets without bells and whistles (Table 2); the effect of embedding transfer by our method is qualitatively demonstrated in Fig. 5. On the other hand, the performance of existing embedding transfer methods is inferior to that of the source model on the SOP dataset. The proposed method demonstrates more interesting results in the *dimensionality reduction* setting (Table 1(b)): It outperforms recent metric learning methods, MS and DiVA, whose embedding dimension is 8 times higher (Table 2). This result enables significant speedup of image retrieval systems at the cost of a tiny performance drop. Finally, in the *model compression* setting (Table 1(c)), our method achieves impressive performance even with a substantially smaller network and a lower embedding dimension; the performance drop by the compression is marginal and its accuracy is as competitive as MS with a heavier network and a larger embedding dimension.

We found that the knowledge distillation methods tend to underperform, especially on the CUB-200-2011 and Cars-196 datasets. In particular, their performance depends heavily on the coupled metric learning loss since they cannot directly transfer knowledge of the source embedding space. In contrast, our method is superior to them in most experiments with no additional loss nor memory buffer [44].

4.1.3 Ablation Study

We conduct ablation study in the *self-transfer* setting to examine the contribution of each component of the relaxed contrastive loss. The results are summarized in Table 3.

We first validate the effect of label relaxation by replacing indicator y_{ij} in Eq. (1) with semantic similarity w_{ij}^s obtained from source embedding model. The result

Recall@K		CUB-200-2011			Cars-196			SOP		
		1	2	4	1	2	4	1	10	100
MS [46]	BN^{512}	65.7	77.0	86.3	84.1	90.4	94.0	78.2	90.5	96.0
SoftTriple [34]	BN^{512}	65.4	76.4	84.5	84.5	90.7	94.5	78.3	90.3	95.9
DiVA [27]	BN^{512}	66.8	77.7	-	84.1	90.7	-	78.1	90.6	-
PA [19]	BN^{512}	69.1	78.9	86.1	86.4	91.9	95.0	79.2	90.7	96.2
Ours	BN^{512}	**72.1**	**81.3**	**87.6**	**89.6**	**94.0**	**96.5**	**79.8**	**91.1**	**96.3**
MS [46]	BN^{64}	57.4	69.8	80.0	77.3	85.3	90.5	74.1	87.8	94.7
SoftTriple [34]	BN^{64}	60.1	71.9	81.2	78.6	86.6	91.8	76.3	**89.1**	**95.3**
DiVA [27]	BN^{64}	63.0	74.5	83.3	78.3	86.6	91.2	73.7	87.5	94.8
PA [19]	BN^{64}	61.7	73.0	81.8	78.8	87.0	92.2	**76.5**	89.0	95.1
Ours	BN^{64}	**67.4**	**78.0**	**85.9**	**86.5**	**92.3**	**95.3**	76.3	88.6	94.8
PA [19]	$R18^{128}$	61.8	72.9	82.1	78.7	86.5	91.7	76.2	89.1	95.2
Ours	$R18^{128}$	**66.6**	**78.1**	**85.9**	**86.0**	**91.6**	**95.3**	**78.4**	**90.4**	**96.1**

Table 2. Image retrieval performance of the proposed method and the state-of-the-art metric learning models. Embedding networks of the methods are fixed by Inception with BatchNorm (BN) for fair comparisons, and superscripts indicate embedding dimensions.

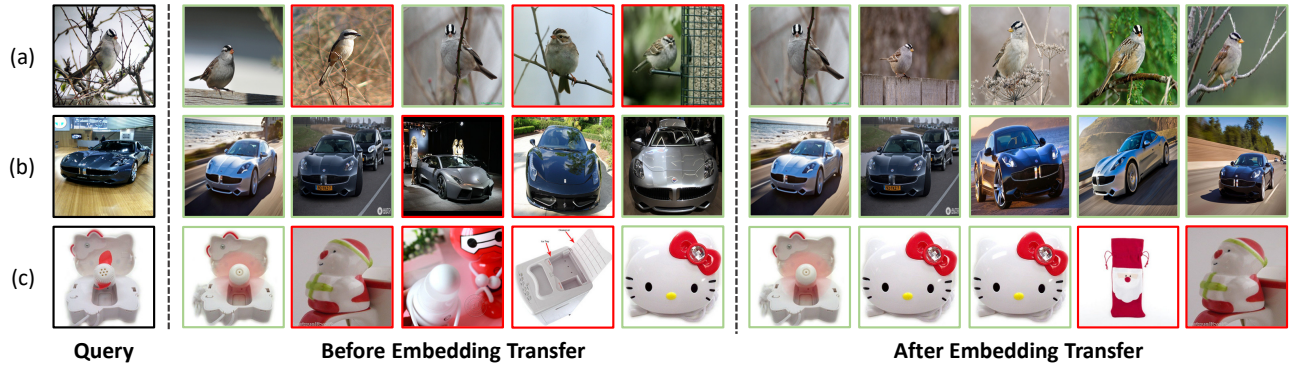

Figure 5. Top 5 image retrievals of the state of the art [19] before and after the proposed method is applied. (a) CUB-2020-2011. (b) Cars-196. (c) SOP. Images with green boundary are success cases and those with red boundary are false positives. More qualitative results can be found in the supplementary material.

Methods	Recall@1	
	CUB	Cars
Original contrastive loss	65.3	80.5
+ Label relaxation	70.4	88.2
+ Relative distance	72.1	89.6

Table 3. Ablation study of the components of our loss on the CUB-200-2011 (CUB) and Cars-196 (Cars) datasets.

suggests that label relaxation significantly improves performance by exploiting the rich semantic relations between samples. Also, we investigate the effect of using the relative distance, *i.e.*, removing l_2 normalization to be free from the restriction on the embedding manifold. The result shows that adopting the relative distance further improves the performance as it allows to fully exploit the entire embedding dimensions with no restriction; a similar observation has been reported in [32].

4.2. Self-supervised Representation Learning

Knowledge distillation has been known to improve the performance of classification models by self-transfer [11], but is not available for self-supervised representation learning due to the absence of class labels. We argue that embedding transfer can play this role for networks trained in a self-supervised manner since it distills and transfers knowledge without relying on class labels.

This section examines a potential of embedding transfer methods in this context, by learning representations using the embedding transfer methods and the knowledge extracted from existing self-supervised networks. Our method is compared with RKD [32] and PKT [33], but DarkRank [6] is excluded since its complexity, proportional to the number of sample permutations, is excessively large in the self-supervised learning setting. We adopt a network trained by SimCLR [4], the state of the art in self-supervised representation learning, as the source embedding model.

4.2.1 Setup

Datasets and evaluation. Self-supervised models and those enhanced by embedding transfer are evaluated on the CIFAR-10 [24] and STL-10 [8] datasets. In the STL-10 dataset, both the labeled training set and unlabeled set are

Dataset	CIFAR-10	STL-10
Before embedding transfer	93.4	89.2
PKT [33]	65.3	71.6
RKD [32]	93.6	79.8
Ours	**93.9**	**89.6**

Table 4. Performance of linear classifiers trained on representations obtained by embedding transfer techniques incorporated with self-supervised learning frameworks.

used for training, and the rest are kept for testing. The performance of the models is measured by the linear evaluation protocol [2, 56, 57], in which a linear classifier on top of a frozen self-supervised network is trained and evaluated.

Source models and their training. We reimplement SimCLR [4] framework to train source embedding models. Following the original framework, ResNet50 is employed as the base network of the source models and a Multi-Layer Perceptron (MLP) head is appended to its last pooling layer. On the CIFAR-10 dataset, the source models are trained for 1K epochs while following the details (*e.g.*, augmentation, learning rate, and temperature) described in [4]. On the STL-10 dataset, we adopt the same configuration except that Gaussian blur is additionally employed for data augmentation.

Target models and their training. For training of target models, the MLP on top of the source models are removed and their embedding vectors are l_2 normalized. Target models have the same architecture as their source counterpart where the MLP head is removed, but with no l_2 normalization. Details of training target models are the same as those for the corresponding source models. All target models are trained using the LARS optimizer [51] with initial learning rate of 4.0 and weight decay of 10^{-6}. We warm up the learning rate linearly during the first 10 epochs and apply the cosine decay [25] after that. Regarding hyperparameters, both δ and σ in our loss are set to 1.

4.2.2 Results

The performance of embedding transfer methods in the self-supervised learning task is summarized in Table 4. The proposed method improves the quality of the learned representations on both of the two datasets. Moreover, it clearly outperforms the existing embedding transfer techniques when incorporated with SimCLR. In contrast, other embedding transfer methods are often inferior to the source model, and especially PKT shows unstable performance in every experiment. This is because of their limitations: As the batch size increases, the probability distributions considered by PKT becomes nearly uniform, and the computational burden of RKD grows significantly due to its angle calculation. Our method enhances the performance of the existing self-supervised models without such difficulties.

Source Target	ResNet56 (72.34) ResNet20 (69.06)	VGG13 (74.64) VGG8 (70.36)
HKD [16]	70.66	72.98
RKD [32] + HKD	71.18	72.97
CRD [44] + HKD	71.63	**74.29**
Ours + HKD	**71.95**	73.82

Table 5. Test accuracy of target models on the CIFAR100 dataset.

4.3. Image Classification

Finally, we demonstrate that the proposed method can be used also for enhancing classifiers as a knowledge distillation technique. Following the convention in this task, its efficacy is validated in the model compression setting on the CIFAR-100 [24] dataset with two source–target combinations: ResNet56–ResNet20 [15] and VGG13–VGG8 [38]. Our method is compared with RKD [16] and CRD [44] as well as HKD [16]; all methods including ours are combined with HKD and use the cross-entropy loss additionally. In detail, our relaxed contrastive loss utilizes the outputs from the last pooling layer of the source and target models, and the embedding vectors of the source are l_2 normalized. We directly follow [44] for other training details, and both δ and σ in our loss are set to 1.

As shown in Table 5, our method is comparable to or outperforming the state of the art [44]. This result indicates that our method is universal and can be applied to tasks other than metric learning.

5. Conclusion

We have presented a novel method to distill and transfer knowledge of a learned embedding model effectively. Our loss utilizes rich pairwise relations between samples in the source embedding space as the knowledge through relaxed relation labels, and effectively transfers the knowledge by focusing more on sample pairs important for learning target embedding models. As a result, our method has achieved impressive performance over the state of the art on metric learning benchmarks and demonstrated that it can reduce the size and embedding dimension of an embedding model significantly with a negligible performance drop. Moreover, we have shown that our method can enhance the quality of self-supervised representation and performance of classification models.

Acknowledgement: This work was supported by the NRF grant, the IITP grant, and R&D program for Advanced Integrated-intelligence for IDentification, funded by Ministry of Science and ICT, Korea (No.2019-0-01906 Artificial Intelligence Graduate School Program–POSTECH, NRF-2021R1A2C3012728–30%, NRF-2018R1A5A1060031–20%, NRF-2018M3E3A1057306–30%, IITP-2020-0-00842–20%).

References

[1] Sungsoo Ahn, Shell Xu Hu, Andreas Damianou, Neil D. Lawrence, and Zhenwen Dai. Variational information distillation for knowledge transfer. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[2] Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. In *Proc. Neural Information Processing Systems (NeurIPS)*, 2019.

[3] Maxime Bucher, Stéphane Herbin, and Frédéric Jurie. Improving semantic embedding consistency by metric learning for zero-shot classification. In *Proc. European Conference on Computer Vision (ECCV)*, 2016.

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proc. International Conference on Machine Learning (ICML)*, 2020.

[5] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[6] Yuntao Chen, Naiyan Wang, and Zhaoxiang Zhang. Darkrank: Accelerating deep metric learning via cross sample similarities transfer. In *Proc. AAAI Conference on Artificial Intelligence (AAAI)*, 2017.

[7] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[8] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proc. International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2011.

[9] Michael A. A. Cox and Trevor F. Cox. *Multidimensional Scaling*, pages 315–347. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: a large-scale hierarchical image database. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[11] Tommaso Furlanello, Zachary Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born again neural networks. In *Proc. International Conference on Machine Learning (ICML)*, 2018.

[12] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[13] Ben Harwood, Vijay Kumar B G, Gustavo Carneiro, Ian Reid, and Tom Drummond. Smart mining for deep metric learning. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2017.

[14] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

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

[17] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proc. International Conference on Machine Learning (ICML)*, 2015.

[18] Pierre Jacob, David Picard, Aymeric Histace, and Edouard Klein. Metric learning with horde: High-order regularizer for deep embeddings. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2019.

[19] Sungyeon Kim, Dongwon Kim, Minsu Cho, and Suha Kwak. Proxy anchor loss for deep metric learning. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[20] Sungyeon Kim, Minkyo Seo, Ivan Laptev, Minsu Cho, and Suha Kwak. Deep metric learning beyond binary supervision. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[21] Wonsik Kim, Bhavya Goyal, Kunal Chawla, Jungmin Lee, and Keunjoo Kwon. Attention-based ensemble for deep metric learning. In *Proc. European Conference on Computer Vision (ECCV)*, 2018.

[22] Byungsoo Ko and Geonmo Gu. Embedding expansion: Augmentation in embedding space for deep metric learning. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[23] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 554–561, 2013.

[24] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009.

[25] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

[26] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *Proc. International Conference on Learning Representations (ICLR)*, 2019.

[27] Timo Milbich, Karsten Roth, Homanga Bharadhwaj, Samarth Sinha, Yoshua Bengio, Björn Ommer, and Joseph Paul Cohen. Diva: Diverse visual feature aggregation fordeep metric learning. In *Proc. European Conference on Computer Vision (ECCV)*, 2020.

[28] Deen Dayal Mohan, Nishant Sankaran, Dennis Fedorishin, Srirangaraj Setlur, and Venu Govindaraju. Moving in the right direction: A regularization for deep metric learning. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[29] Yair Movshovitz-Attias, Alexander Toshev, Thomas K Leung, Sergey Ioffe, and Saurabh Singh. No fuss distance metric learning using proxies. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2017.

[30] M. Opitz, G. Waltner, H. Possegger, and H. Bischof. Bier — boosting independent embeddings robustly. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2017.

[31] Michael Opitz, Georg Waltner, Horst Possegger, and Horst Bischof. Deep metric learning with bier: Boosting independent embeddings robustly. *IEEE Transactions on Pattern*

Analysis and Machine Intelligence (TPAMI), 2018.

[32] Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[33] Nikolaos Passalis and Anastasios Tefas. Learning deep representations with probabilistic knowledge transfer. In *Proc. European Conference on Computer Vision (ECCV)*, 2018.

[34] Qi Qian, Lei Shang, Baigui Sun, Juhua Hu, Hao Li, and Rong Jin. Softtriple loss: Deep metric learning without triplet sampling. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2019.

[35] Limeng Qiao, Yemin Shi, Jia Li, Yaowei Wang, Tiejun Huang, and Yonghong Tian. Transductive episodic-wise adaptive metric for few-shot learning. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2019.

[36] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. In *Proc. International Conference on Learning Representations (ICLR)*, 2014.

[37] Florian Schroff, Dmitry Kalenichenko, and James Philbin. FaceNet: A unified embedding for face recognition and clustering. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[38] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proc. International Conference on Learning Representations (ICLR)*, 2015.

[39] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Proc. Neural Information Processing Systems (NeurIPS)*, 2017.

[40] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *Proc. Neural Information Processing Systems (NeurIPS)*, 2016.

[41] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[42] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[43] Joshua B. Tenenbaum, Vin de Silva, and John C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[44] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation. In *Proc. International Conference on Learning Representations (ICLR)*, 2019.

[45] Jiang Wang, Yang Song, T. Leung, C. Rosenberg, Jingbin Wang, J. Philbin, Bo Chen, and Ying Wu. Learning fine-grained image similarity with deep ranking. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[46] Xun Wang, Xintong Han, Weilin Huang, Dengke Dong, and Matthew R Scott. Multi-similarity loss with general pair weighting for deep metric learning. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[47] Xun Wang, Haozhi Zhang, Weilin Huang, and Matthew R Scott. Cross-batch memory for embedding learning. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6388–6397, 2020.

[48] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010.

[49] Chao-Yuan Wu, R. Manmatha, Alexander J. Smola, and Philipp Krahenbuhl. Sampling matters in deep embedding learning. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2017.

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

[51] Yang You, Igor Gitman, and Boris Ginsburg. Large batch training of convolutional networks. *arXiv preprint arXiv:1708.03888*, 2017.

[52] Baosheng Yu and Dacheng Tao. Deep metric learning with tuplet margin loss. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2019.

[53] Lu Yu, Vacit Oguz Yazici, Xialei Liu, Joost van de Weijer, Yongmei Cheng, and Arnau Ramisa. Learning metrics from teachers: Compact networks for image embedding. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[54] Sukmin Yun, Jongjin Park, Kimin Lee, and Jinwoo Shin. Regularizing class-wise predictions via self-knowledge distillation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[55] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *Proc. International Conference on Learning Representations (ICLR)*, 2016.

[56] Xiaohua Zhai, Avital Oliver, Alexander Kolesnikov, and Lucas Beyer. S4l: Self-supervised semi-supervised learning. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2019.

[57] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *Proc. European Conference on Computer Vision (ECCV)*, 2016.

[58] Ziming Zhang and Venkatesh Saligrama. Zero-shot learning via joint latent similarity embedding. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.