Deep Metric Learning with Tuplet Margin Loss

Baosheng Yu and Dacheng Tao
UBTECH Sydney AI Centre, School of Computer Science,
Faculty of Engineering, The University of Sydney, Darlington, NSW 2008, Australia
{baosheng.yu, dacheng.tao}@sydney.edu.au

Abstract

Deep metric learning, in which the loss function plays a key role, has proven to be extremely useful in visual recognition tasks. However, existing deep metric learning loss functions such as contrastive loss and triplet loss usually rely on delicately selected samples (pairs or triplets) for fast convergence. In this paper, we propose a new deep metric learning loss function, tuplet margin loss, using randomly selected samples from each mini-batch. Specifically, the proposed tuplet margin loss implicitly up-weights hard samples and down-weights easy samples, while a slack margin in angular space is introduced to mitigate the problem of overfitting on the hardest sample. Furthermore, we address the problem of intra-pair variation by disentangling class-specific information to improve the generalizability of tuplet margin loss. Experimental results on three widely used deep metric learning datasets, CARS196, CUB200-2011, and Stanford Online Products, demonstrate significant improvements over existing deep metric learning methods.

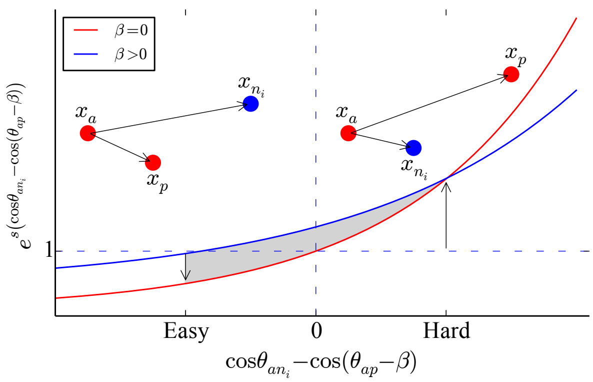
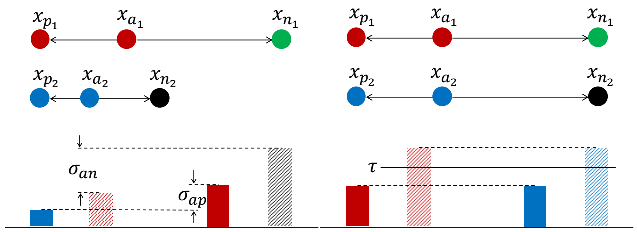


Figure 1: An illustration of tuplet margin loss function. Given a tuplet $(x_a, x_p, x_{n_1}, \dots, x_{n_{k-1}})$, tuplet margin loss exponentially up-weights hard triplets and down-weights easy triplets within the tuplet. Specifically, the loss of each triplet (x_a, x_p, x_{n_i}) is defined by the scale factor $s > 1$ and the violate margin $\cos\theta_{an_i} - \cos\theta_{ap}$. A slack margin $\beta > 0$ is used to mitigate the problem of overfitting on the hardest triplets by paying more attention to "moderately hard triplets" (the shaded area). See more details in Section 3.3.

1. Introduction

Deep metric learning focuses on learning a deep feature embedding consistent with semantic similarity, *i.e.*, a small intra-class variation and a large inter-class variation [38, 35]. It has been proven that deep metric learning methods are extremely valuable in visual recognition tasks such as one-shot learning [5, 29], image retrieval [9, 18], person re-identification [39, 12], and face recognition [27, 23]. With the growing scale of training data, *i.e.*, both the number of samples and classes, deep metric learning loss function has attracted more and more attention in large-scale visual recognition tasks [23, 17].

Deep metric learning loss function can be divided into two main groups: (1) classification-based loss functions, *e.g.*, large-margin softmax loss [14] and center loss [36]; and (2) distance-based loss functions, *e.g.*, contrastive loss [2, 27] and triplet loss [24, 23]. However, existing loss functions usually suffer from several inherent drawbacks. Specifically, classification-based loss functions usually use a classification layer or a reference point for each class [36], in which both the computation and the requirements on device memory increase linearly with the number of classes [8]. Recently, several methods such as dynamic class selection [43] and distributed parallel acceleration [4] have been developed to relieve the computation and memory bottlenecks in classification-based loss functions, while the discussion of approximation algorithms for massive classification is beyond the scope of this paper. Regardless of the heavy classification layer or massive reference points, distance-based loss functions directly optimize the margin between intra- and inter-class distances, and are independent with the number of classes [23]. However, existing distance-based loss functions, *e.g.*, triplet loss,



(a) (b)

Figure 2: An illustration of intra-pair variation. The height of each bar indicates pairwise distance, *i.e.*, color-fill bar for positive pair and pattern-fill bar for negative pair. In both (a) and (b), there is a clear margin between positive and negative pairs, *i.e.*, $\forall i = 1, 2$, we have $d(x_{a_i}, x_{p_i}) < d(x_{a_i}, x_{n_i})$. However, in (a), the distance metric on each specific type of pairs (positive or negative pairs) varies among different classes, *i.e.*, the distribution of pairwise distance is class-dependent on training set. Comparing with the class-independent distribution shown in (b), the intra-pair variation increases the risk of failing to find a proper threshold τ to separate all positive and negative pairs and degrades the performance for visual recognition such as verification task. A class-independent distance metric can be learned by minimizing the intra-pair variances, *i.e.*, σ_{ap} for positive pairs and σ_{an} for negative pairs. See more details in Section 3.4.

usually suffer from the problem of slow convergence, and rely heavily on mining informative samples for fast convergence [23, 8], raising a number of severe sampling problems: (1) the number of possible triplets grows cubically with the number of training samples [8]; (2) mining informative triplets tends to be difficult, *e.g.*, both randomly selected triplets and the hardest triplets lead to bad local minima [23, 6]; and (3) the training stability benefits from large mini-batches, in which cross-device synchronization is a non-trivial engineering task [23, 17].

Recently, significant improvements on distance-based loss function have been achieved by using the notation of tuplet, which generalizes a triplet with multiple negative examples to form a better approximation of inter-class distance [18, 25, 17]. Although delivering impressive performance improvements, tuplet-based loss functions further exacerbate the sampling problems, because the computational complexity exponentially increases with the number of negative examples. Therefore, here we develop a new tuplet-based loss function, tuplet margin loss, using a set of *randomly* sampled tuplets. Unlike previous distance-based loss functions [18, 25], in which informative samples are explicitly selected by sampling heuristics, we address informative samples from the view of loss function, while using only a set of randomly sampled tuplets. Inspired by the focal loss for object detection [13], we exponentially up-weight hard triplets and down-weight easy triplets within each tuplet (see an example in Figure 1). However, the exponential weighting scheme usually tends to form a relatively large margin between the intra- and inter-class distances by overfitting the hardest triplet in each tuplet. To solve this problem, we introduce a slack margin in angular space to pay more attention to "moderately hard triplets" rather than "the hardest triplets" [23]. An intuitive example of the slack margin for changing the weighting scheme is shown in Figure 1.

Distance-based loss functions, including the proposed tuplet margin loss, focus on optimizing the margin between intra- and inter-class distances by penalizing the margin between positive and negative pairs with the same anchor point, while leaving a risk of learning a class-dependent distance metric from the training set. A class-dependent distance metric indicates that the distance distribution of positive pairs (or negative pairs) varies among different classes, which has not been well-addressed in previous work [1]. We refer to the variation within each type of pairs (positive or negative pairs) as the intra-pair variation, and argue that the intra-pair variation degrades the generalizability of deep metric learning model. An intuitive failure case induced by the intra-pair variation is shown in Figure 2. To solve this problem, we disentangle the intra-pair variation from intra/inter-class variation and try to learn a class-independent distance metric by minimizing the intra-pair variances in both positive and negative pairs.

In this paper, our main contribution is a new tuplet-based deep metric learning loss function: (1) we propose a tuplet margin loss by using a set of randomly selected samples, which is computationally more efficient than explicitly mining informative samples; (2) we introduce a slack margin to address the problem of overfitting on the hardest sample; and (3) we address the problem of intra-pair variation to further improve the generalizability of deep metric learning model. Specifically, with the proposed tuplet margin loss, we achieve the state-of-the-art results on three widely used deep metric learning datasets, *i.e.*, CARS196 [11], CUB200-2011 [30], and Stanford Online Products [18].

2. Related Work

Classification-based Loss Function. Feature embeddings learned by the classification loss generalize well to a variety of visual recognition tasks [21, 40]. Inspired by this, center loss [36] and large-margin softmax loss [14] have been proposed to further improve the discriminability of classification-based loss function. Specifically, center loss minimizes the distance between each example and its class center, forming a class-dependent constraint. Large-margin softmax loss has since been significantly improved by both

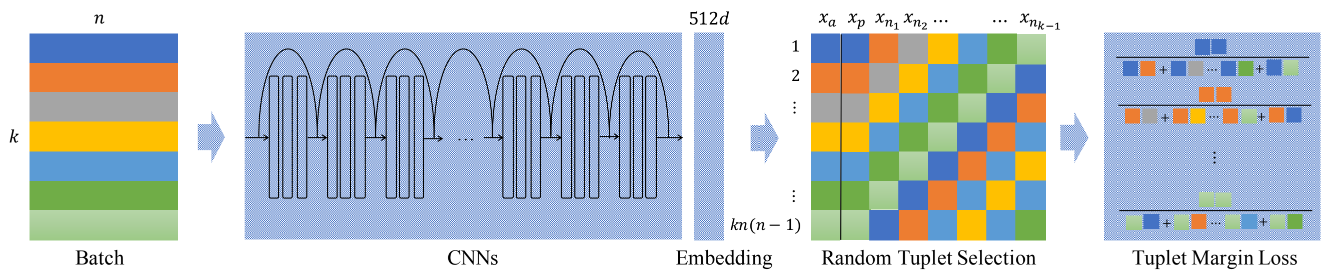


Figure 3: An illustration of the tuplet-based deep metric learning framework. We first randomly sample a mini-batch of training data, which contains kn training samples from k different classes, *i.e.*, n samples per class. The deep neural network is used to learn a fixed-dimensional feature embedding, e.g, 512d. We then construct a set of tuplets using all $kn(n-1)$ positive pairs within the mini-batch and negative examples are randomly sampled from each of the other $k-1$ classes. Finally, the loss function is evaluated on the tuplets constructed from each mini-batch.

feature normalization [22, 32, 16, 45] and weight normalization [15, 16]. Recently, several different types of margin, such as additive cosine margin [33, 31] and additive angular margin [4] have been explored to further improve the large-margin softmax loss performance.

Distance-based Loss Function. Due to the scalability for a large number of classes, distance-based loss functions, especially the triplet loss, have attracted considerable attention in many visual recognition tasks such as image retrieval [9, 18], person re-identification [39, 12], and face recognition [27, 23]. However, triplet loss usually suffers from slow convergence, so the triplet selection method has become central to improving the performance of triplet loss [23]. Inspired by this, several improvements to triplet selection have been proposed: (1) novel triplet selection methods, *e.g.*, batch-hard triplets [8], and distance-weighted sampling [37]; (2) correcting selection bias by learning an invariant representation [41]; and (3) generating hard triplets via adversarial networks [44].

Recently, deep metric learning loss functions have been further improved by exploring new pairwise structures [18, 28, 25, 17, 26]. Specifically, both lifted structured loss [18] and N-pair loss [25] share similar motivation by making full use of each mini-batch or exploring negative examples from multiple different classes to give a better approximation for inter-class distance. The proposed tuplet margin loss falls within the same category with [18] and [25], *i.e.*, tuplet-based loss functions.

Ensemble Deep Metric Learning. Besides improved loss functions, improvements have also been achieved by exploring ensemble methods in deep metric learning, such as boosting [19, 20], cascades [42], hierarchical structures [6], and attention-based ensembles [10]. Specifically, these ensemble methods usually are complementary with different loss functions [10] and might be used to further improve the performance of the proposed tuplet margin loss.

3. Method

In this section, we first introduce tuplet-basd loss function for deep metric learning. We then formulate the proposed tuplet margin loss as well as the random tuplet selection method. Lastly, we introduce the problem of intra-pair variation and the proposed intra-pair variance minimization method.

3.1. Tuplet-based Deep Metric Learning

Let $x \in X$ denote the data and $y \in Y$ denote its label, deep metric learning aims to learn a discriminative feature embedding $f(x)$ with a small intra-class distance and a large inter-class distance, *i.e.*,

$$\|f(x_a) - f(x_p)\|_2^2 < \|f(x_a) - f(x_n)\|_2^2, \qquad (1)$$

where x_a, x_p share the same label and x_n has a different label. This constraint is known as the triplet constraint and we usually refer to (x_a, x_p, x_n) as a triplet, in which x_a is called anchor example, x_p is the positive example, and x_n is the negative example. The notation of tuplet generalizes the triplet to explore multiple negative examples [18, 25]. In this paper, we use the definition of tuplet similar to [25] as follows:

$$t = (x_a, x_p, x_{n_1}, \dots, x_{n_{k-1}}), \qquad (2)$$

where k is the number of classes in each mini-batch and all negative examples $x_{n_i}, i = 1, \dots, k-1$ come from different classes. Specifically, the relationship between the tuplet and triplet can be described as follows: each tuplet $(x_a, x_p, x_{n_1}, \dots, x_{n_{k-1}})$ contains $k-1$ triplets, sharing the same positive pair (x_a, x_p), *i.e.*,

$$(x_a, x_p, x_{n_i}), \; \forall\, i = 1, \dots, k-1. \qquad (3)$$

The triplet constraint then can be generalized to the tuplet as follows: $\forall\, i = 1, 2, \dots, k-1$,

$$\|f(x_a) - f(x_p)\|_2^2 < \|f(x_a) - f(x_{n_i})\|_2^2. \qquad (4)$$

Similar to [25], a typical tuplet-based loss function can be defined as follows:

$$\mathcal{L}_{tuplet} = \log\left(1 + \sum_{i=1}^{k-1} e^{d(x_a,x_p)-d\left(x_a,x_{n_i}\right)}\right), \tag{5}$$

where $d(\cdot,\cdot)$ is the distance function, *i.e.*,

$$d(x_1, x_2) = \|f(x_1) - f(x_2)\|_2^2. \tag{6}$$

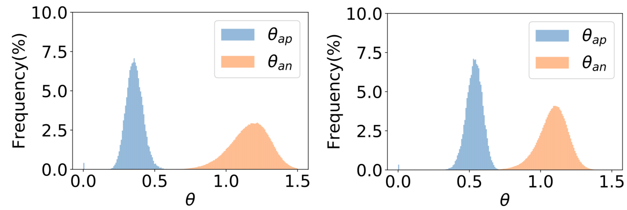
3.2. Random Tuplet Selection

Mining informative samples is not only a delicate problem for the convergence of distance-based loss functions, but also computation intensive in practice, especially for tuplet-based loss functions. Previous work puts a lot of efforts on more effective and efficient sampling methods. Specifically, the distance-weighted sampling method [37] aims to perform an unbiased sampling towards all distances, while the hard negative class mining method [25] tries to reduce the computation complexity by keeping only 2 samples for positive class.

Unlike previous work, we address the sampling problems from the perspective of loss function itself. Therefore, we use randomly sampled tuplets in this paper and we introduce the random tuplet selection method as follows. The proposed random tuplet selection method works in an online manner, *i.e.*, tuplets are sampled from each mini-batch. Specifically, each mini-batch contains kn randomly sampled training examples from k classes with n samples per class. We then collect all positive pairs within the mini-batch, *i.e.*, $kn(n-1)$ positive pairs in total. For each positive pair (x_a, x_p), we randomly sample one negative example from each of the other $k-1$ classes to form a tuplet, $(x_a, x_p, x_{n_1}, \ldots, x_{n_{k-1}})$. Finally, we obtain $kn(n-1)$ tuplets from each mini-batch.

3.3. Tuplet Margin Loss

Considering that both the norm $\|f(x)\|_2$ and the direction of feature embedding $f(x)$ have influence on the margin between positive and negative pairs, tuplet-based loss function usually minimizes the L2-norm of feature embedding $\|f(x)\|_2$ to remove the influence of feature norm, *i.e.*, the classification decision is only related to the direction of feature embedding [25]. However, we find that the margin between positive and negative pairs is upper bounded by the norm of feature embedding. Furthermore, it has been observed that the loss function tends to be minimized by increasing the norm of the feature embedding for easy samples [22]. Inspired by this, we argue that the feature norm also changes the weighting scheme on easy samples and hard samples. Therefore, we disentangle the norm and the direction of feature embedding by (1) preserving the direction of feature embedding $f(x)$ using L2-normalization,



(a) $\beta = 0$ (b) $\beta = 0.1$

Figure 4: An illustration of the slack margin for the tuplet-based loss function. A relative large margin between intra-class and inter-class distance is usually achieved by overfitting on the hardest triplets in training data. The slack margin can be directly used to change the distribution of positive and negative pairs in the embedding space.

i.e., $\|f(x)\|_2^2 = 1$, and (2) introducing a scale factor $s \geq 0$ to control the norm of feature embedding. We then reformulate the tuplet-based loss function as follows:

$$\mathcal{L}_{tuplet} = \log\left(1 + \sum_{i=1}^{k-1} e^{s\left(\cos\theta_{an_i} - \cos\theta_{ap}\right)}\right), \tag{7}$$

where θ_{ap} is the angle between $f(x_a)$ and $f(x_p)$, and θ_{an_i} is the angle between $f(x_a)$ and $f(x_{n_i})$.

An intuitive explanation of the scale factor s is the radius of the hyper-sphere where the feature embeddings are located [33, 31, 4]. That is, the scale factor s has severe influence on the convergence due to a lower bound on the difference between cosine similarities, *i.e.*, $\forall\, \theta_1,\ \theta_2$,

$$e^{\cos\theta_1 - \cos\theta_2} \geq 1/e^2 \gg 0. \tag{8}$$

Considering that deep metric learning models are mainly optimized by using stochastic gradient descent (SGD) method, we also consider the influence of the scale factor s from the view of gradient. Specifically, given w as the model parameter and the tuplet-based loss function defined in (7), we then have

$$\begin{aligned}\frac{\partial \mathcal{L}_{tuplet}}{\partial w} &= \frac{s}{1 + \sum_{j=1}^{k-1} e^{s\alpha_j}} \sum_{i=1}^{k-1}\left(\mathbf{e}^{s\alpha_i}\frac{\partial \alpha_i}{\partial w}\right) \\ &\propto \sum_{i=1}^{k-1}\left(\mathbf{e}^{s\alpha_i}\frac{\partial \alpha_i}{\partial w}\right),\end{aligned} \tag{9}$$

where α_i is the violate margin, *i.e.*,

$$\alpha_i = \cos\theta_{an_i} - \cos\theta_{ap},\ \forall\, i = 1, \ldots, k-1. \tag{10}$$

As we can see from (9) and (10), the gradient with respect to the hard triplet ($\alpha_i > 0$) will be exponentially up-weighted, while the gradient with respect to the easy triplet ($\alpha_i > 0$) will be exponentially down-weighted. That is, the scale factor s can be used to *implicitly* explore hard triplets from randomly sampled tuplets for fast convergence.

The tuplet-based loss function exponentially up-weights the gradients of hard triplets according to their violate margins, with the hardest triplet counting for much more than the other triplets. As a result, the tuplet-based loss function usually forms a relatively large margin between the intra- and inter-class distances by overfitting the hardest triplet, *i.e.*,

$$\theta_{an_i} \gg \theta_{ap}, \forall\, i = 1, \ldots, k-1. \tag{11}$$

To address the above problem, we introduce a slack margin $\beta \geq 0$ to form a relaxation of (11) as follows:

$$\theta_{an_i} \gg \theta_{ap} - \beta,\ \forall i = 1, \ldots, k-1. \tag{12}$$

The proposed tuplet margin loss then can be derived by applying the slack margin β into the tuplet-based loss function as follows:

$$\mathcal{L}_{tuplet} = \log\left(1 + \sum_{i=1}^{k-1} e^{s\left(\cos\theta_{an_i} - \ \cos(\theta_{ap} - \beta)\right)}\right). \tag{13}$$

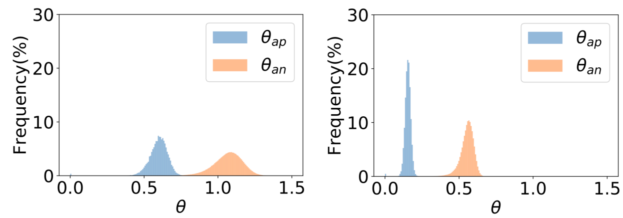
We refer to this new loss function as the tuplet margin loss. An illustration of the influence of the proposed slack margin is shown in Figure 4. Specifically, the proposed slack margin not only changes the distribution of pairwise distance in positive and negative pairs but also forces the loss to pay more attention to "moderately hard triplets". Therefore, the proposed slack margin improves the performance of the tuplet-based loss function by reducing the risk of overfitting on the hardest triplets.

3.4. Intra-pair Variation

Distance-based loss functions, including the proposed tuplet margin loss, optimize the margin between intra- and inter-class distances. However, a clear margin between positive and negative pairs sharing the same anchor example does not always indicate a good generalization [1]. An intuitive example is shown in Figure 2 and we attribute the poor generalization to the class-dependent distance metric. Specifically, given two triplets $(x_{a_1}, x_{p_1}, x_{n_1})$ and $(x_{a_2}, x_{p_2}, x_{n_2})$, in which x_{a_1} and x_{a_2} are from different classes, a small intra-class distance and a large inter-class distance is usually described by the triplet constraint, *i.e.*,

$$d(x_{a_i}, x_{p_i}) < d(x_{a_i}, x_{n_i}),\ \forall i = 1, 2. \tag{14}$$

From (14), we see that the triplet constraint is dependent upon the class of the anchor example x_{a_i}, while it takes the risk of an unbalanced distance metric among different classes, *e.g.*, $d(x_{a_1}, x_{p_1}) > d(x_{a_2}, x_{n_2})$. Specifically, if two random variables D_1 and D_2 denote the pairwise distance of positive (or negative) pairs from two different classes, the difference $P(D_1)$ and $P(D_2)$ then indicates the class-dependent information. As a result, we argue that the class-dependent information learned from the training set degrades the generalizability of the deep metric learning model to unseen test data.



(a) $\lambda = 0$ (b) $\lambda = 0.5$

Figure 5: An illustration of the intra-pair variation minimization. By minimizing the intra-pair variation, all positive pairs (or negative pairs) have more consistent and compact distribution regardless of the class information. Furthermore, the intra-pair variation minimization can be seen as a regularization for the distance-based loss function.

We formulate the above class-dependent distributions as follows. Let D_i denote the pairwise distance of positive (or negative) pairs, in which all anchor examples are from the class i. Considering that each mini-batch is randomly sampled from k classes with n examples per class, the distribution of pairwise distance on all positive pairs (or negative pairs) can be formulated as the averaged mixture of $P(D_i)$, *i.e.*,

$$P(D) = \frac{1}{k}\sum_{i=1}^{k} P(D_i).$$

Theorem 1. *Given a set of independent distributions $P(D_i)$, $i = 1, \ldots, k$, and their averaged mixture $P(D)$, we then have the variance of D as follows:*

$$\sigma^2 = \frac{1}{k}\sum_{i=1}^{k}\sigma_i^2 + \frac{1}{k^2}\sum_{i<j}(\mu_i - \mu_j)^2,$$

where u_i and σ_i^2 denote the mean and variance of D_i, respectively.

Proof. In Appendix. □

From Theorem 1, we know that both the variance in each class σ_i^2 and the difference between different classes $|\mu_i - \mu_j|$ can be well-captured by their averaged mixture, *i.e.*, the variance of all positive (or negative) pairs σ^2. Inspired by this, we reduce the influence of the class-dependent information from the training data, *e.g.*, bias and

β	R@1	R@2	R@4	R@8	R@16
0	89.4	93.9	96.3	97.8	98.8
0.05	90.9	95.0	97.0	98.1	98.9
0.10	**91.5**	**95.4**	**97.3**	**98.5**	**99.2**
0.15	89.0	94.4	96.9	**98.5**	**99.2**
0.20	85.2	92.0	95.5	97.7	99.0

(a) Comparison of different β

λ	R@1	R@2	R@4	R@8	R@16
0	91.5	95.4	97.3	98.5	99.2
0.3	93.5	96.6	97.9	98.8	**99.4**
0.5	**93.7**	**96.7**	**98.1**	**98.9**	99.3
1.0	93.6	96.4	98.0	98.8	99.3
1.5	92.6	96.0	97.5	98.5	99.1

(b) Comparison of different λ

Table 1: Effectiveness of the slack margin and the intra-pair variation minimization. In (a), we perform experiments on different β by using the same $\lambda = 0$. In (b), we use $\beta = 0.1$ and perform experiments on different λ.

noise, by minimizing the variance σ^2 within each type of pairs. We refer to the variation in each type of pairs as the intra-pair variation and minimize the intra-pair variation of all positive pairs as follows:

$$\mathcal{L}_{pos} = \mathbb{E}[(1-\epsilon)\mu_{ap} - \cos\theta_{ap}]_+^2, \tag{15}$$

where $[\cdot]_+ = \max(0,\cdot)$, $\mu_{ap} = \mathbb{E}\left[\cos\theta_{ap}\right]$ is the mean cosine similarity of all positive pairs, and a small positive scalar $\epsilon = 0.01$ is used for convergence. Similarly, we define the loss function for all negative pairs as

$$\mathcal{L}_{neg} = \mathbb{E}[\cos\theta_{an} - (1+\epsilon)\mu_{an}]_+^2. \tag{16}$$

An illustration of the intra-pair variation minimization can be found in Figure 5. Finally, we learn the deep feature embedding by jointly minimizing the tuplet margin loss and the intra-pair loss as follows:

$$\mathcal{L} = \mathcal{L}_{tuplet} + \lambda\left(\mathcal{L}_{pos} + \mathcal{L}_{neg}\right), \tag{17}$$

where $\lambda > 0$ forms a trade-off between two loss functions.

4. Implementation

We implement the proposed method using Pytorch [1]. For training, all images are resized to 224×224, and we crop images when bounding boxes are available. We horizontally flip all training images randomly with the probability 0.5 for data augmentation. We use ResNet-50 [7] as the backbone network in most of our experiments, while we demonstrate the scalability of the proposed method to a larger model using ResNet-101. All our models are initialized from the weights pretrained on ImageNet [3]. Unless mentioned, we use the feature dimension of 512 and a batch-size of 256 (*i.e.*, $k = 32$ and $n = 8$). We use SGD with a momentum of 0.9 and a weight decay of 0.0001. The learning rate starts from 0.01 and is divided by 10 for every 30 epochs. We train our models for maximum 100 epochs and report the performance at the best epoch.

[1] https://pytorch.org

5. Experiments

We evaluate the proposed method on three popular image retrieval datasets, *i.e.*, CARS196 [11], CUB200-2011 [30], and Stanford Online Products [18]. We use the same evaluation metric, Recall@K metric, and the same train/test protocol with [18]:

- CARS196 [11] contains 16,185 images of 196 different car models and is divided into two parts: all 8054 images from the first 98 classes are used for training, while the remaining 8131 images are used for testing.
- CUB200-2011 [30] contains 11,788 images of 200 different bird species. All 5864 images from the first 100 classes are used for training and the remaining 5924 images are used for testing.
- Stanford Online Products [18] contains 120,053 images of 22,634 different products. All 59,551 images from the first 11,318 classes are used for training and 60,502 images from the remaining 11,316 classes are used for testing.

5.1. Effectiveness of Tuplet Margin Loss

To demonstrate the effectiveness of the proposed tuplet margin loss, especially the slack margin and the intra-pair variation minimization, we conduct a number of experiments for different β and λ on the cropped version of CARS196 dataset. We use ResNet-50 as the backbone network and fix other hyper-parameters to: $s = 64$, $k = 32$ and $n = 8$. Experimental results are shown in Table 1. Specifically, in Table 1(a), we see that the proposed tuplet margin loss greatly improves the performance of the tuplet-based loss function by using a proper slack margin, $\beta = 0.1$. In Table 1(b), with the proposed intra-pair variation minimization method, the performance of the tuplet margin loss is further improved by a clear margin, *e.g.*, R@1 from 91.5% to 93.7%.

5.2. Comparison with Current State-of-the-Art

We compare the proposed tuplet margin loss with recent state-of-the-art methods such as Angular [34], HDC

Method	CARS196					CUB200-2011				
	R@1	R@2	R@4	R@8	R@16	R@1	R@2	R@4	R@8	R@16
N-pairs [25]	71.1	79.7	86.5	91.6	-	51.0	63.3	74.3	83.2	-
Angular [34]	71.4	81.4	87.5	92.1	-	54.7	66.3	76.0	83.9	-
Proxy-NCA [17]	73.2	82.4	86.4	87.8	-	49.2	61.9	67.9	72.4	-
HDC [42]	73.7	83.2	89.5	93.8	96.7	53.6	65.7	77.0	85.6	91.5
Margin [37]	79.6	86.5	91.9	95.1	97.3	**63.6**	**74.4**	**83.1**	**90.0**	**94.2**
BIER [19]	78.0	85.8	91.1	95.1	97.3	55.3	67.2	76.9	85.1	91.7
A-BIER [20]	82.0	89.0	93.2	96.1	97.8	57.5	68.7	78.3	86.2	91.9
ABE [10]	85.2	90.5	94.0	96.1	-	60.6	71.5	79.8	87.4	-
TML (**ours**)	**86.3**	**92.3**	**95.4**	**97.3**	**98.7**	62.5	73.9	83.0	89.4	**94.2**
Method	CARS196(cropped)					CUB200-2011(cropped)				
	R@1	R@2	R@4	R@8	R@16	R@1	R@2	R@4	R@8	R@16
HDC [42]	83.8	89.8	93.6	96.2	97.8	60.7	72.4	81.9	89.2	93.7
Margin [37]	86.9	92.7	95.6	97.6	98.7	63.9	75.3	84.4	90.6	94.8
BIER [19]	87.2	92.2	95.3	97.4	98.5	63.7	74.0	82.5	89.3	93.8
A-BIER [20]	90.3	94.1	96.8	97.9	98.9	65.5	75.8	83.9	90.2	94.2
ABE [10]	93.0	95.9	97.5	98.5	-	70.6	79.8	86.9	92.2	-
TML (**ours**)	**93.7**	**96.7**	**98.1**	**98.9**	**99.2**	**73.7**	**83.0**	**89.7**	**93.6**	**96.4**

Table 2: Results on CARS196 and CUB200-2011.

Method	R@1	R@10	R@100	R@1000
Lifted [18]	62.1	79.8	91.3	97.4
Histogram [28]	63.9	81.7	92.2	97.7
N-pairs [25]	67.7	83.8	93.0	97.8
HDC [42]	69.5	84.4	92.8	97.7
Angular [34]	70.9	85.0	93.5	98.0
Margin [37]	72.7	86.2	93.8	98.0
Proxy-NCA [17]	73.7	-	-	-
BIER [19]	72.7	86.5	94.0	98.0
A-BIER [20]	74.2	86.9	94.0	97.8
ABE [10]	76.3	88.4	94.8	98.2
TML (**ours**)	**78.0**	**91.2**	**96.7**	**99.0**

Table 3: Results on Stanford Online Products. Specifically, we randomly sample 4 images per class for each mini-batch due to limited images for some classes. To obtain the proper number of tuplets, each mini-batch contains examples sampled from 96 classes.

[42], Margin [37], and Proxy-NCA [17]. Specifically, for fair comparison on CARS196 and CUB200-2011, we report both the performance with and without using tight bounding boxes. For experiments in Table 2 and Table 3, we use ResNet-50 as the backbone network and fix other hyper-parameters to: $s = 64$, $\beta = 0.1$, and $\lambda = 0.5$. Unless mentioned, we use $k = 32$ and $n = 8$ for each mini-batch. We see that the proposed tuplet margin loss (TML) significantly outperforms all other methods, including several ensemble-based methods, BIER [19], A-BIER [20], and ABE [10]. Furthermore, as a typical deep metric learning loss function, the proposed tuplet margin loss might be further improved by these ensemble-based frameworks.

5.3. Ablation Study

We perform several ablation studies on the cropped version of CARS196 dataset, to better understand important hyper-parameters in tuplet margin loss. We use the ResNet-50 as backbone network and fix other parameters to: $\beta = 0.1$ and $\lambda = 0.5$. Experimental results on the scale factor s and the feature embedding dimension are shown in Table 4 and Table 5. Specifically, a larger scale factor s makes it easier for the model to fit all training data, while increasing the risk of overfitting. In Table 4, we find that $s = 64$ is a good trade-off in our experiments, which is consistent with the experience in the classification-based loss functions [22, 33, 31, 4]. In Table 5, we see that the proposed tuplet margin loss also works well with a smaller feature dimension, *e.g.*, 128, which is computationally more efficient in practice.

To further demonstrate the influence of different batch-sizes and backbone networks for the proposed tuplet margin loss, we perform a number of experiments on the cropped version of CARS196. For fair comparison, we fix the following hyper-parameters, $s = 64$, $\beta = 0.1$, and $\lambda = 0.5$. In Table 6, we see that tuplet-margin loss achieves better performance with a more powerful backbone network. In

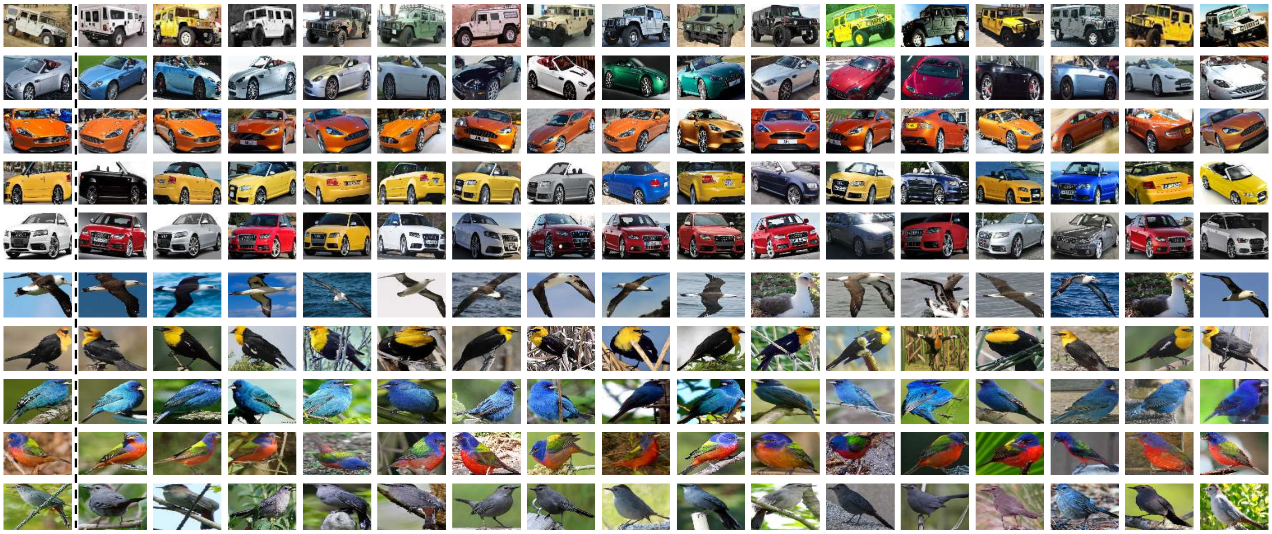


Figure 6: Retrieval results on CARS196 and CUB200-2011. The first column refers to query images.

s	R@1	R@2	R@4	R@8	R@16
1	69.1	79.9	87.5	92.9	96.3
8	77.6	84.6	89.7	93.5	95.9
16	86.3	91.3	94.2	96.2	97.6
32	91.2	94.5	96.5	97.9	98.7
64	**93.7**	**96.7**	**98.1**	**98.9**	99.3
128	92.1	96.1	97.9	98.8	**99.4**

Table 4: Comparison of different scale factors. We use $k = 32$, $n = 8$, and the feature dimension 512.

Dim	R@1	R@2	R@4	R@8	R@16
128	92.3	95.8	97.5	98.5	99.1
256	93.1	96.2	97.6	98.6	99.1
512	**93.7**	**96.7**	**98.1**	**98.9**	**99.3**
1024	93.5	96.4	97.8	98.8	99.1

Table 5: Comparison of different feature dimensions. We use $k = 32$, $n = 8$, and the scale factor $s = 64$.

Table 7, we find that the proposed tuplet margin loss is not very sensitive to different batch sizes, while the best performance is achieved by a small batch-size, which is similar to the loss function in classification task.

6. Conclusion

In this paper, we propose a new tuplet-based loss function, tuplet margin loss, for deep metric learning. We introduce a slack margin to mitigate the problem of overfitting on the hardest sample and address the problem of intra-pair variation to further improve the generalizability of tuplet margin loss. Specifically, the proposed tuplet margin loss uses randomly sampled data and is not very sensitive to different batch sizes, making it interesting to examine its scalability in large-scale distributed training setting and we leave it for future study.

Backbone	R@1	R@2	R@4	R@8	R@16
ResNet-50	93.7	**96.7**	98.1	**98.9**	**99.3**
ResNet-101	**94.3**	**96.7**	**98.2**	**98.9**	**99.3**

Table 6: Comparison of different backbone networks. We use $k = 32$ and $n = 8$.

k	n	R@1	R@2	R@4	R@8	R@16
32	4	93.2	96.3	97.8	98.7	**99.3**
32	8	**93.7**	**96.7**	**98.1**	**98.9**	**99.3**
64	4	92.2	96.2	97.7	98.6	99.2
64	8	92.3	95.8	97.5	98.5	99.1

Table 7: Comparison of different batch-sizes. We use ResNet-50 as the backbone network.

7. Acknowledgement

Baosheng Yu and Dacheng Tao were partially supported by Australian Research Council Projects FL-170100117 and DP-180103424.

References

[1] Weihua Chen, Xiaotang Chen, Jianguo Zhang, and Kaiqi Huang. Beyond triplet loss: a deep quadruplet network for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 403–412, 2017. 2, 5

[2] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 539–546, 2005. 1

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 248–255. Ieee, 2009. 6

[4] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4690–4699, 2019. 1, 3, 4, 7

[5] Li Fei-Fei, Rob Fergus, and Pietro Perona. One-shot learning of object categories. *IEEE transactions on pattern analysis and machine intelligence*, 28(4):594–611, 2006. 1

[6] Weifeng Ge, Weilin Huang, Dengke Dong, and Matthew R. Scott. Deep metric learning with hierarchical triplet loss. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 2, 3

[7] 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 (CVPR)*, pages 770–778, 2016. 6

[8] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*, 2017. 1, 2, 3

[9] Junshi Huang, Rogerio S Feris, Qiang Chen, and Shuicheng Yan. Cross-domain image retrieval with a dual attribute-aware ranking network. In *Proceedings of the IEEE international conference on computer vision (ICCV)*, pages 1062–1070, 2015. 1, 3

[10] Wonsik Kim, Bhavya Goyal, Kunal Chawla, Jungmin Lee, and Keunjoo Kwon. Attention-based ensemble for deep metric learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 3, 7

[11] 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. 2, 6

[12] Wei Li, Rui Zhao, Tong Xiao, and Xiaogang Wang. Deepreid: Deep filter pairing neural network for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 152–159, 2014. 1, 3

[13] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision (ICCV)*, pages 2980–2988, 2017. 2

[14] Weiyang Liu, Yandong Wen, Zhiding Yu, and Meng Yang. Large-margin softmax loss for convolutional neural networks. In *International Conference on Machine Learning*, pages 507–516, 2016. 1, 2

[15] Weiyang Liu, Yan-Ming Zhang, Xingguo Li, Zhiding Yu, Bo Dai, Tuo Zhao, and Le Song. Deep hyperspherical learning. In *Advances in Neural Information Processing Systems*, pages 3950–3960, 2017. 3

[16] Yu Liu, Hongyang Li, and Xiaogang Wang. Rethinking feature discrimination and polymerization for large-scale recognition. *arXiv preprint arXiv:1710.00870*, 2017. 3

[17] Yair Movshovitz-Attias, Alexander Toshev, Thomas K Leung, Sergey Ioffe, and Saurabh Singh. No fuss distance metric learning using proxies. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 360–368, 2017. 1, 2, 3, 7

[18] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4004–4012, 2016. 1, 2, 3, 6, 7

[19] Michael Opitz, Georg Waltner, Horst Possegger, and Horst Bischof. Bier-boosting independent embeddings robustly. In *IProceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. 3, 7

[20] 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*, 2018. 3, 7

[21] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, et al. Deep face recognition. In *British Machine Vision Conference (BMVC)*, 2015. 2

[22] Rajeev Ranjan, Carlos D Castillo, and Rama Chellappa. L2-constrained softmax loss for discriminative face verification. *arXiv preprint arXiv:1703.09507*, 2017. 3, 4, 7

[23] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 815–823, 2015. 1, 2, 3

[24] Matthew Schultz and Thorsten Joachims. Learning a distance metric from relative comparisons. In *Advances in neural information processing systems*, pages 41–48, 2004. 1

[25] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *Advances in Neural Information Processing Systems*, pages 1857–1865, 2016. 2, 3, 4, 7

[26] Hyun Oh Song, Stefanie Jegelka, Vivek Rathod, and Kevin Murphy. Deep metric learning via facility location. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, volume 8, 2017. 3

[27] Yi Sun, Yuheng Chen, Xiaogang Wang, and Xiaoou Tang. Deep learning face representation by joint identification-verification. In *Advances in neural information processing systems*, pages 1988–1996, 2014. 1, 3

[28] Evgeniya Ustinova and Victor Lempitsky. Learning deep embeddings with histogram loss. In *Advances in Neural In-*

formation Processing Systems, pages 4170–4178, 2016. 3, 7

[29] Oriol Vinyals, Charles Blundell, Tim Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, pages 3630–3638, 2016. 1

[30] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011. 2, 6

[31] Feng Wang, Jian Cheng, Weiyang Liu, and Haijun Liu. Additive margin softmax for face verification. *IEEE Signal Processing Letters*, 25(7):926–930, 2018. 3, 4, 7

[32] Feng Wang, Xiang Xiang, Jian Cheng, and Alan Loddon Yuille. Normface: l 2 hypersphere embedding for face verification. In *Proceedings of ACM International Conference on Multimedia*, pages 1041–1049. ACM, 2017. 3

[33] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. Cosface: Large margin cosine loss for deep face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5265–5274, 2018. 3, 4, 7

[34] Jian Wang, Feng Zhou, Shilei Wen, Xiao Liu, and Yuanqing Lin. Deep metric learning with angular loss. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2612–2620, 2017. 6, 7

[35] Kilian Q Weinberger and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 10(Feb):207–244, 2009. 1

[36] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. A discriminative feature learning approach for deep face recognition. In *European Conference on Computer Vision (ECCV)*, pages 499–515. Springer, 2016. 1, 2

[37] Chao-Yuan Wu, R Manmatha, Alexander J Smola, and Philipp Krähenbühl. Sampling matters in deep embedding learning. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. 3, 4, 7

[38] Eric P Xing, Michael I Jordan, Stuart J Russell, and Andrew Y Ng. Distance metric learning with application to clustering with side-information. In *Advances in neural information processing systems*, pages 521–528, 2003. 1

[39] Dong Yi, Zhen Lei, Shengcai Liao, and Stan Z Li. Deep metric learning for person re-identification. In *International Conference on Pattern Recognition (ICPR)*, pages 34–39, 2014. 1, 3

[40] Dong Yi, Zhen Lei, Shengcai Liao, and Stan Z Li. Learning face representation from scratch. *arXiv preprint arXiv:1411.7923*, 2014. 2

[41] Baosheng Yu, Tongliang Liu, Mingming Gong, Changxing Ding, and Dacheng Tao. Correcting the triplet selection bias for triplet loss. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 71–87, 2018. 3

[42] Yuhui Yuan, Kuiyuan Yang, and Chao Zhang. Hard-aware deeply cascaded embedding. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 814–823, 2017. 3, 7

[43] Xingcheng Zhang, Lei Yang, Junjie Yan, and Dahua Lin. Accelerated training for massive classification via dynamic class selection. In *Thirty-Second AAAI Conference on Artificial Intelligence (AAAI)*, 2018. 1

[44] Yiru Zhao, Zhongming Jin, Guo-jun Qi, Hongtao Lu, and Xian-sheng Hua. An adversarial approach to hard triplet generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 501–517, 2018. 3

[45] Yutong Zheng, Dipan K Pal, and Marios Savvides. Ring loss: Convex feature normalization for face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5089–5097, 2018. 3