



Deep Metric Learning with Hierarchical Triplet Loss

Weifeng Ge^{1,2,3}, Weilin Huang^{1,2}(✉), Dengke Dong^{1,2}, and Matthew R. Scott^{1,2}

¹ Malong Technologies, Shenzhen, China

{terrencege, whuang, dongdk, mscott}@malong.com

² Shenzhen Malong Artificial Intelligence Research Center, Shenzhen, China

³ The University of Hong Kong, Pok Fu Lam, Hong Kong

Abstract. We present a novel hierarchical triplet loss (HTL) capable of automatically collecting informative training samples (triplets) via a defined hierarchical tree that encodes global context information. This allows us to cope with the main limitation of random sampling in training a conventional triplet loss, which is a central issue for deep metric learning. Our main contributions are two-fold. (i) we construct a hierarchical class-level tree where neighboring classes are merged recursively. The hierarchical structure naturally captures the intrinsic data distribution over the whole dataset. (ii) we formulate the problem of triplet collection by introducing a new violate margin, which is computed dynamically based on the designed hierarchical tree. This allows it to automatically select meaningful hard samples with the guide of global context. It encourages the model to learn more discriminative features from visual similar classes, leading to faster convergence and better performance. Our method is evaluated on the tasks of image retrieval and face recognition, where it outperforms the standard triplet loss substantially by 1%–18%, and achieves new state-of-the-art performance on a number of benchmarks.

Keywords: Deep metric learning · Image retrieval · Triplet loss
Anchor-neighbor sampling

1 Introduction

Distance metric learning or similarity learning is the task of learning a distance function over images in visual understanding tasks. It has been an active research topic in computer vision community. Given a similarity function, images with similar content are projected onto neighboring locations on a manifold, and images with different semantic context are mapped apart from each other. With the boom of deep neural networks (DNN), metric learning has been turned from learning distance functions to learning deep feature embeddings that better fits a simple distance function, such as Euclidean distance or cosine distance. Metric learning with DNNs is referred as deep metric learning, which has recently

achieved great success in numerous visual understanding tasks, including images or object retrieval [26, 30, 34], single-shot object classification [30, 32, 34], key-point descriptor learning [12, 24], face verification [20, 22], person re-identification [23, 30], object tracking [29] and etc.

Recently, there is a number of widely-used loss functions developed for deep metric learning, such as contrastive loss [6, 27], triplet loss [22] and quadruplet loss [5]. These loss functions are calculated on correlated samples, with a common goal of encouraging samples from the same class to be closer, and pushing samples of different classes apart from each other, in a projected feature space. The correlated samples are grouped into contrastive pairs, triplets or quadruplets, which form the training samples for these loss functions on deep metric learning. Unlike softmax loss used for image classification, where the gradient is computed on each individual sample, the gradient of a deep metric learning loss often depends heavily on multiple correlated samples. Furthermore, the number of training samples will be increased exponentially when the training pairs, triplets or quadruplets are grouped. This generates a vast number of training samples which are highly redundant and less informative. Training that uses random sampling from them can be overwhelmed by redundant samples, leading to slow convergence and inferior performance.

Deep neural networks are commonly trained using online stochastic gradient descent algorithms [19], where the gradients for optimizing network parameters are computed *locally* with mini-batches, due to the limitation of computational power and memory storage. It is difficult or impossible to put all training samples into a single mini-batch, and the networks can only focus on local data distribution within a mini-batch, making it difficult to consider global data distribution over the whole training set. This often leads to local optima and slow convergence. This common challenge will be amplified substantially in deep metric learning, due to the enlarged sample spaces where the redundancy could become more significant. Therefore, collecting and creating meaningful training samples (e.g., in pairs, triplets or quadruplets) has been a central issue for deep metric learning, and an efficient sampling strategy is of critical importance to this task. This is also indicated in recent literature [1, 20, 22, 35].

Our goal of this paper is to address the sampling issue of conventional triplet loss [22]. In this work, we propose a novel hierarchical triplet loss (HTL) able to automatically collect informative training triplets via an adaptively-learned hierarchical class structure that encodes global context in an elegant manner. Specifically, we explore the underline data distribution on a manifold sphere, and then use this manifold structure to guide triplet sample generation. Our intuition of generating meaningful samples is to encourage the training samples within a mini-batch to have similar visual appearance but with different semantic content (e.g., from different categories). This allows our model to learn more discriminative features by identifying subtle distinction between the close visual concepts. Our main contribution are described as follows.

- We propose a novel hierarchical triplet loss that allows the model to collect informative training samples with the guide of a global class-level hierarchi-

cal tree. This alleviates main limitation of random sampling in training of deep metric learning, and encourages the model to learn more discriminative features from visual similar classes.

- We formulate the problem of triplet collection by introducing a new violate margin, which is computed dynamically over the constructed hierarchical tree. The new violate margin allows us to search informative samples, which are hard to distinguish between visual similar classes, and will be merged into a new class in next level of the hierarchy. The violate margin is automatically updated, with the goal of identifying a margin that generates gradients for violate triplets, naturally making the collected samples more informative.
- The proposed HTL is easily implemented, and can be readily integrated into the standard triplet loss or other deep metric learning approaches, such as contrastive loss, quadruplet loss, recent HDC [38] and BIER [17]. It significantly outperforms the standard triplet loss on the tasks of image retrieval and face recognition, and obtains new state-of-art results on a number of benchmarks.

2 Related Work

Deep Metric Learning. Deep metric learning maps an image into a feature vector in a manifold space via deep neural networks. In this manifold space, the Euclidean distance (or the cosine distance) can be directly used as the distance metric between two points. The contribution of many deep metric learning algorithms, such as [2, 3, 5, 22, 26], is the design of a loss function that can learn more discriminant features. Since neural networks are usually trained using the stochastic gradient descent (SGD) in mini-batches, these loss functions are difficult to approximate the target of metric learning - pull samples with the same label into nearby points and push samples with different labels apart.

Informative Sample Selection. Given N training images, there are about $O(N^2)$ pairs, $O(N^3)$ triplets, and $O(N^4)$ quadruplets. It is infeasible to traverse all these training tuples during training. Schroff *et al.* [22] constructed a mini-batch of with 45 identities and each of which has 40 images. There are totally 1800 images in a mini-batch, and the approach obtained the state-of-art results on LFW face recognition challenge [8]. While it is rather inconvenient to take thousands of images in a mini-batch with a large-scale network, due to the limitation of GPU memory. For deep metric learning, it is of great importance to selecting informative training tuples. Hard negative mining [4] is widely used to select hard training tuples. Our work is closely related to that of [7, 35] which inspired the current work. Distance distribution was applied to guide tuple sampling for deep metric learning [7, 35]. In this work, we strive to a further step by constructing a hierarchical tree that aggregates class-level global context, and formulating tuple selection elegantly by introducing a new violate margin.

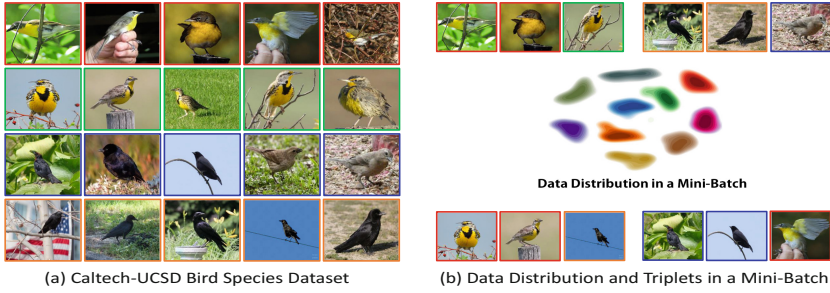


Fig. 1. (a) Caltech-UCSD Bird Species Dataset [31]. Images in each row are from the same class. There are four classes in different colors—red, green, blue and yellow. (b) Data distribution and triplets in a mini-batch. Triplets in the top row violate the triplet constrain in the traditional triplet loss. Triplets in the bottom row are ignored in the triplet loss, but are revisited in the hierarchical triplet loss. (Color figure online)

3 Motivation: Challenges in Triplet Loss

We start by revisiting the main challenges in standard triplet loss [22], which we believe have a significant impact to the performance of deep triplet embedding.

3.1 Preliminaries

Let (\mathbf{x}_i, y_i) be the i -th sample in the training set $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$. The feature embedding of \mathbf{x}_i is represented as $\phi(\mathbf{x}_i, \theta) \in \mathbb{R}^d$, where θ is the learnable parameters of a differentiable deep networks, d is the dimension of embedding and y_i is the label of \mathbf{x}_i . $\phi(\cdot, \theta)$ is usually normalized into unit length for the training stability and comparison simplicity as in [22]. During the neural network training, training samples are selected and formed into triplets, each of which $\mathcal{T}_z = (\mathbf{x}_a, \mathbf{x}_p, \mathbf{x}_n)$ are consisted of an anchor sample \mathbf{x}_a , a positive sample \mathbf{x}_p and a negative sample \mathbf{x}_n . The labels of the triplet $\mathcal{T}_z = (\mathbf{x}_a^z, \mathbf{x}_p^z, \mathbf{x}_n^z)$ satisfy $y_a = y_p \neq y_n$. Triplet loss aims to pull samples belonging to the same class into nearby points on a manifold surface, and push samples with different labels apart from each other. The optimization target of the triplet \mathcal{T}_z is,

$$l_{tri}(\mathcal{T}_z) = \frac{1}{2} \left[\|\mathbf{x}_a^z - \mathbf{x}_p^z\|^2 - \|\mathbf{x}_a^z - \mathbf{x}_n^z\|^2 + \alpha \right]_+.$$

$[\cdot]_+ = \max(0, \cdot)$ denotes the hinge loss function, and α is the violate margin that requires the distance $\|\mathbf{x}_a^z - \mathbf{x}_n^z\|^2$ of negative pairs to be larger than the distance $\|\mathbf{x}_a^z - \mathbf{x}_p^z\|^2$ of positive pairs. For all the triplets \mathcal{T} in the training set $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$, the final objective function to optimize is,

$$\mathcal{L} = \frac{1}{Z} \sum_{\mathcal{T}^z \in \mathcal{T}} l_{tri}(\mathcal{T}_z),$$

where Z is the normalization term. For training a triplet loss in deep metric learning, the violate margin plays a key role to sample selection.

3.2 Challenges

Challenge 1: Triplet Loss with Random Sampling. For many deep metric learning loss functions, such as contrastive loss [6], triplet loss [22] and quadruplet loss [5], all training samples are treated equally with a constant violate margin, which only allows training samples that violate this margin to produce gradients. For a training set $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{k=1}^N$ with N samples, training a triplet loss will generate $O(N^3)$ triplets, which is infeasible to put all triplets into a single mini-batch. When we sample the triplets over the whole training set randomly, it has a risk of slow convergence and pool local optima. We identify the problem that most of training samples obey the violate margin when the model starts to converge. These samples can not contribute gradients to the learning process, and thus are less informative, but can dominate the training process, which significantly degrades the model capability, with a slow convergence. This inspired current work that formulates the problem of sample selection via setting a dynamic violate margin, which allows the model to focus on a small set of informative samples.

However, identifying informative samples from a vast number of the generated triplets is still challenging. This inspires us to strive to a further step, by sampling meaningful triplets from a structural class tree, which defines class-level relations over all categories. This transforms the problem of pushing hard samples apart from each other into encouraging a larger distance between two confusing classes. This not only reduces the search space, but also avoid over-fitting the model over individual samples, leading to a more discriminative model that generalizes better.

Challenge 2: Risk of Local Optima. Most of the popular metric learning algorithms, such as the contrastive loss, the triplet loss, and the quadruplet loss, describe similarity relationship between individual samples locally in a mini-batch, without considering global data distribution. In triplet loss, all triplet is treated equally. As shown in Fig. 1, when the training goes after several epoches, most of training triplets dose not contribute to the gradients of learnable parameters in deep neural networks. There has been recent work that aims to solve this problem by re-weighting the training samples, as in [36]. However, even with hard negative mining or re-weighting, the triplets can only see a few samples within a mini-batch, but not the whole data distribution. It is difficult for the triplet loss to incorporate the global data distribution on the target manifold space. Although the data structure in the deep feature space are changed dynamically during the training process, the relative position of data points can be roughly preserved. This allows us to explore the data distribution obtained in the previous iterations to guide sample selection in the current stage. With this prior knowledge of data structure, a triplet, which does not violate the original margin α , is possible to generate gradients that contribute to the network training, as shown in Fig. 1. Discriminative capability can be enhanced by learning from these hard but informative triplets.

4 Hierarchical Triplet Loss

We describe details of the proposed hierarchical triplet loss, which contains two main components, constructing a hierarchical class tree and formulating the hierarchical triplet loss with a new violate margin. The hierarchical class tree is designed to capture global data context, which is encoded into triplet sampling via the new violate margin, by formulating the hierarchical triplet loss.

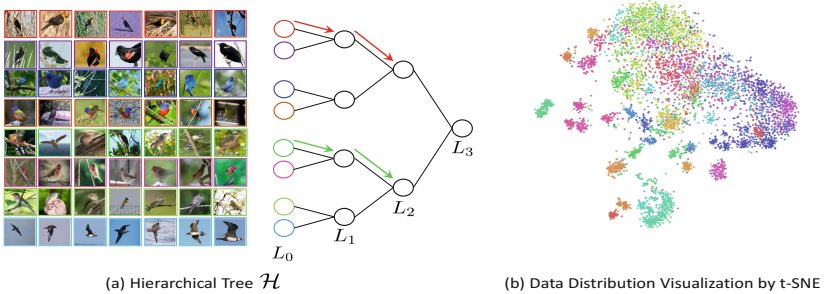


Fig. 2. (a) A toy example of the hierarchical tree \mathcal{H} . Different colors represent different image classes in CUB-200-2011 [31]. The leaves are the image classes in the training set. Then they are merged recursively until to the root node. (b) The training data distribution of 100 classes visualized by using t-SNE [16] to reduce the dimension of triplet embedding from 512 to 2.

4.1 Manifold Structure in Hierarchy

We construct a global hierarchy at the class level. Given a neural network $\phi_t(\cdot, \theta) (\in \mathbb{R}^d)$ pre-trained using the traditional triplet loss, we get the hierarchical data structure based on sample rules. Denote the deep feature of a sample \mathbf{x}_i as $\mathbf{r}_i = \phi_t(\mathbf{x}_i, \theta)$. We first calculate a distance matrix of \mathcal{C} classes in the whole training set \mathcal{D} . The distance between the p -th class and the q -th class is computed as,

$$d(p, q) = \frac{1}{n_p n_q} \sum_{i \in p, j \in q} \|\mathbf{r}_i - \mathbf{r}_j\|^2,$$

where n_p and n_q are the numbers of training samples in the p -th and the q -th classes respectively. Since the deep feature \mathbf{r}_i is normalized into unit length, the value of the interclass distance $d(p, q)$ varies from 0 to 4.

We build hierarchical manifold structure by creating a hierarchical tree, according to the computed interclass distances. The leaves of the hierarchical tree are the original image classes, where each class represents a leaf node at the 0-th level. Then hierarchy is created by recursively merging the leaf nodes at different levels, based on the computed distance matrix. The hierarchical tree is set into L levels, and the average inner distance d_0 is used as the threshold for merging the nodes at the 0-th level.

$$d_0 = \frac{1}{C} \sum_{c=1}^C \left(\frac{1}{n_c^2 - n_c} \sum_{i \in c, j \in c} \|r_i - r_j\|^2 \right).$$

where n_c is the number of samples in the c -th class. Then the nodes are merged with different thresholds. At the l -th level of the hierarchical tree, the merging threshold is set to $d_l = \frac{l(4-d_0)}{L} + d_0$. Two classes with a distance less than d_l are merged into a node at the l -th level. The node number at the l -th level is N_l . The nodes are merged from the 0-th level to the L -th level. Finally, we generate a hierarchical tree \mathcal{H} which starts from the leave nodes of original image classes to a final top node, as shown in Fig. 2(a). The constructed hierarchical tree captures class relationships over the whole dataset, and it is updated interactively at the certain iterations over the training.

4.2 Hierarchical Triplet Loss

We formulate the problem of triplet collection into a hierarchical triplet loss. We introduce a dynamical violate margin, which is the main difference from the conventional triplet loss using a constant violate margin.

Anchor Neighbor Sampling. We randomly select l' nodes at the 0-th level of the constructed hierarchical tree \mathcal{H} . Each node represents an original class, and collecting classes at the 0-th level aims to preserve the diversity of training samples in a mini-batch, which is important for training deep networks with batch normalization [9]. Then $m - 1$ nearest classes at the 0-th level are selected for each of the l' nodes, based on the distance between classes computed in the feature space. The goal of collecting nearest classes is to encourage model to learn discriminative features from the visual similar classes. Finally, t images for each class are randomly collected, resulting in n ($n = l'mt$) images in a mini-batch \mathcal{M} . Training triplets within each mini-batch are generated from the collected n images based on class relationships. We write the anchor-neighbor sampling into A-N sampling for convenience (Fig. 3).

Triplet Generation and Dynamic Violate Margin. Hierarchical triplet loss (computed on a mini-batch of \mathcal{M}) can be formulated as,

$$\mathcal{L}_{\mathcal{M}} = \frac{1}{2Z_{\mathcal{M}}} \sum_{T^z \in \mathcal{T}^{\mathcal{M}}} [\|\mathbf{x}_a^z - \mathbf{x}_p^z\| - \|\mathbf{x}_a^z - \mathbf{x}_n^z\| + \alpha_z]_+.$$

where $\mathcal{T}^{\mathcal{M}}$ is all the triplets in the mini-batch \mathcal{M} , and $Z_{\mathcal{M}} = A_{l'm}^2 A_t^2 C_t^1$ is the number of triplets. Each triplet is constructed as $T^z = (\mathbf{x}_a, \mathbf{x}_p, \mathbf{x}_n)$, and the training triplets are generated as follows. $A_{l'm}^2$ indicates randomly selecting two classes - a positive class and a negative class, from all $l'm$ classes in the mini-batch. A_t^2 means selecting two samples - a anchor sample (\mathbf{x}_a^z) and a positive sample (\mathbf{x}_p^z), from the positive class, and C_t^1 means randomly selecting a negative sample (\mathbf{x}_n^z) from the negative class. $A_{l'm}^2$, A_t^2 and C_t^1 are notations in combinatorial mathematics. See reference [13] for details.

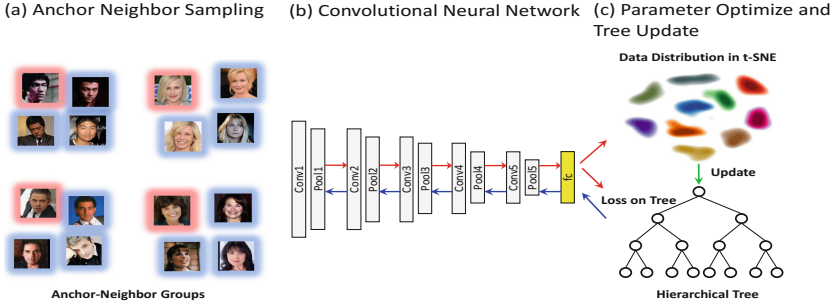


Fig. 3. (a) Sampling strategy of each mini-batch. The images in red stand for anchors and the images in blue stand for the nearest neighbors. (b) Train CNNs with the hierarchical triplet loss. (c) Online update of the hierarchical tree. (Color figure online)

α_z is a dynamic violate margin, which is different from the constant margin of traditional triplet loss. It is computed according to the class relationship between the anchor class y_a and the negative class y_n over the constructed hierarchical class tree. Specifically, for a triplet \mathcal{T}_z , the violate margin α_z is computed as,

$$\alpha_z = \beta + d_{\mathcal{H}(y_a, y_n)} - s_{y_a},$$

where β ($= 0.1$) is a constant parameter that encourages the image classes to reside further apart from each other than the previous iterations. $\mathcal{H}(y_a, y_n)$ is the hierarchical level on the class tree, where the class y_a and the class y_n are merged into a single node in the next level. $d_{\mathcal{H}(y_a, y_n)}$ is the threshold for merging the two classes on \mathcal{H} , and $s_{y_a} = \frac{1}{n_{y_a}^2 - n_{y_a}} \sum_{i, j \in y_a} \|\mathbf{r}_i - \mathbf{r}_j\|^2$ is the average distance between samples in the class y_a . In our hierarchical triplet loss, a sample \mathbf{x}_a is encouraged to push the nearby points with different semantic meanings apart from itself. Furthermore, it also contributes to the gradients of data points which are very far from it, by computing a dynamic violate margin which encodes global class structure via \mathcal{H} . For every individual triplet, we search on \mathcal{H} to encode the context information of the data distribution for the optimization objective. Details of training process with the proposed hierarchical triplet loss are described in Algorithm 1.

Implementation Details. All our experiments are implemented using Caffe [10] and run on an NVIDIA TITAN X(Maxwell) GPU with 12 GB memory. The network architecture is a GoogLeNet [28] with batch normalization [9] which is pre-trained on the ImageNet dataset [21]. The 1000-way fully connected layer is removed, and replace by a d dimensional fully connected layer. The new added layer is initialized with random noise using the “Xaiver” filler. We modify the memory management of Caffe [10] to ensure it can take 650 images in a mini-batch for GoogLeNet with batch normalization. The input images are resized and cropped into 224×224 , and then subtract the mean value. The optimization method used is the standard SGD with a learning rate $1e^{-3}$.

Table 1. Comparisons on the In-Shop Clothes Retrieval Dataset [15].

R@	1	10	20	30	40	50
FashionNet+Joints [15]	41.0	64.0	68.0	71.0	73.0	73.5
FashionNet+Poselets [15]	42.0	65.0	70.0	72.0	72.0	75.0
FashionNet [15]	53.0	73.0	76.0	77.0	79.0	80.0
HDC [38]	62.1	84.9	89.0	91.2	92.3	93.1
BIER [18]	76.9	92.8	95.2	96.2	96.7	97.1
Ours baseline	62.3	85.1	89.0	91.1	92.4	93.4
A-N sampling	75.3	91.8	94.3	96.2	96.7	97.5
HTL	80.9	94.3	95.8	97.2	97.4	97.8

Algorithm 1. Training with hierarchical triplet loss

Input: Training data $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{k=1}^N$. Network $\phi(\cdot, \theta)$ is initialized with a pretrained ImageNet model. The hierarchical class tree \mathcal{H} is built according to the features of the initialized model. The margin α_z for any pair of classes is set to 0.2 at the beginning.

Output: The learnable parameters θ of the neural network $\phi(\cdot, \theta)$.

```

1 while not converge do
2    $t \leftarrow t + 1$  ;
3   Sample anchors randomly and their neighborhoods according to  $\mathcal{H}$  ;
4   Compute the violate margin for different pairs of image classes by
   searching through the hierarchical tree  $\mathcal{H}$  ;
5   Compute the hierarchical triplet loss in a mini-batch  $\mathcal{L}_{\mathcal{M}}$ ;
6   Backpropagate the gradients produced at the loss layer and update
   the learnable parameters ;
7   At each epoch, update the hierarchical tree  $\mathcal{H}$  with current model.

```

5 Experimental Results and Comparisons

We evaluate the proposed hierarchical triplet loss on the tasks of image retrieval and face recognition. Extensive experiments are conducted on a number of benchmarks, including *In-Shop Clothes Retrieval* [15] and *Caltech-UCSD Birds 200* [31] for image retrieval, and LFW [8] for face verification. Descriptions of dataset and implementation details are presented as follows.

5.1 In-Shop Clothes Retrieval

Datasets and Performance Measures. The *In-Shop Clothes Retrieval* dataset [15] is very popular in image retrieval. It has 11735 classes of clothing items and 54642 training images. Following the protocol in [15, 38], 3997



Fig. 4. Anchor-Neighbor visualization on *In-Shop Clothes Retrieval* training set [15]. Each row stands for a kind of fashion style. The row below each odd row is one of neighborhoods of the fashion style in the odd row.

classes are used for training (25882 images) and 3985 classes are for testing (28760 images). The test set are partitioned into the query set and the gallery set, both of which has 3985 classes. The query set has 14218 images and the gallery set has 12612 images. As in Fig. 4, there are a lot image classes that have very similar contents.

For the evaluation, we use the most common Recall@ K metric. We extract the features of each query image and search the K most similar images in the gallery set. If one of the K retrieved images have the same label with the query image, the recall will increase by 1, otherwise will be 0. We evaluate the recall metrics with $K \in \{1, 2, 4, 8, 16, 32\}$.

Implementation Details. Our network is based on GoogLeNet V2 [9]. The dimension d of the feature embedding is 128. The triplet violate margin is set to 0.2. The hierarchical tree has 16 levels including the leaves level which contains the images classes. At the first epoch, the neural network is trained with the standard triplet loss which samples image classes for mini-batches randomly. Then during the training going on, the hierarchical tree is updated and used in the following steps. Since there are 3997 image classes for training and there many similar classes, the whole training needs 30 epoch and the batch size is set to 480. For every 10 epoch, we decrease the learning rate by multiplying 0.1. The testing codes are gotten from HDC [38].

Result Comparison. We compare our method with existing state-of-the-art algorithms and our baseline—triplet loss. Table 1 lists the results of image retrieval on *In-Shop Clothes Retrieval*. The proposed method achieves 80.9% Recall@1, and outperforms the baseline algorithm—triplet loss by 18.6%. It indicates that our algorithm can improve the discriminative power of the original

Table 2. Comparison with the state-of-art on the CUB-200-2011 dataset [31].

R@	1	2	4	8	16	32
LiftedStruct [26]	47.2	58.9	70.2	80.2	89.3	93.2
Binomial deviance [30]	52.8	64.4	74.7	83.9	90.4	94.3
Histogram loss [30]	50.3	61.9	72.6	82.4	88.8	93.7
N-Pair-Loss [25]	51.0	63.3	74.3	83.2	-	-
HDC [38]	53.6	65.7	77.0	85.6	91.5	95.5
BIER [18]	55.3	67.2	76.9	85.1	91.7	95.5
Ours baseline	55.9	68.4	78.2	86.0	92.2	95.5
HTL	57.1	68.8	78.7	86.5	92.5	95.5

triplet loss by a large margin. State-of-the-art algorithms, including HDC [38], and BIER [18], used boosting and ensemble method to take the advantage of different features and get excellent results. Our method demonstrates that by incorporate the global data distribution into deep metric learning, the performance will be highly improved. The proposed hierarchical loss get 80.9% Recall@1, which is 4.0% higher than BIER [18] and 18.8% higher than HDC [38].

5.2 Caltech-UCSD Birds 200-2011

Datasets and Performance Measures. The *Caltech-UCSD Birds 200* dataset (CUB-200-2011) [31] contains photos of 200 bird species with 11788 images. CUB-200-2011 serves as a benchmark in most existing work on deep metric learning and image retrieval. The first 100 classes (5864 images) are used for training, and the rest (5924 images) of classes are used for testing. The rest images are treated as both the query set and the gallery set. For the evaluation, we use the same Recall@ K metric as in Section *In-Shop Clothes Retrieval*. Here, $K \in \{1, 2, 4, 8, 16, 32\}$.

Implementation Details. The dimension d of the feature embedding is 512. The triplet violate margin is set to 0.2. As in the previous section, the hierarchical tree is still set to 16 levels. All the training details are almost the same with the *In-Shop Clothes Retrieval* dataset. But since there are only 100 image classes for training, the dataset is very easy to get overfitting. When we train 10 epoches, the training stopped. The batch size is set to 50. For every 3 epoch, we decrease the learning rate by multiplying 0.1.

Result Comparison. Table 2 lists the results of image retrieval on *Caltech-UCSD Birds 200-2011*. The baseline—triplet loss already get the state-of-art results with 55.9% Recall@1 compared with the previous state-of-art HDC 54.6% and BIER 55.3%. If we use the anchor-neighbor sampling and the hierarchical loss, we get 57.1% Recall@1. Since there are only 100 classes and 6000 images for training, the network is very easy to get overfitting. The performance gain gotten by the hierarchical loss is only 1.2% Recall@1.

5.3 Cars-196 [11] and Stanford Online Products [26]

Details of the Cars-196 and Stanford Online Products [26] are described in [11, 26]. The dimension of the feature embedding is set to 512. The triplet violate margin is set to 0.2, with a hierarchical tree of $depth = 16$. The whole training needs 30 epoch and the batch size is set to 50. For every 10 epoch, we decrease the learning rate by multiplying 0.1.

Results are presented in Table 3, where the proposed HTL outperforms our baseline, BIER and HDC, with clear margins on both datasets. Specifically, on the Cars-196, HTL achieves 81.4% Recall@1, which outperforms original triplet loss by 2.2%, and previous state-of-art by 3.4%. On the Stanford Online Products, HTL achieves 74.8% Recall@1, outperforming triplet loss by 2.2%, and previous state-of-art by 2.1%. These results demonstrate that the proposed HTL can improve original triplet loss efficiently, and further proved the generalization ability of HTL.

Table 3. Comparison with the state-of-art on the cars-196 and Stanford products.

	Cars-196						Stanford online products			
R@	1	2	4	8	16	32	1	10	100	100
HDC	73.7	83.2	89.5	93.8	96.7	98.4	69.5	84.4	92.8	97.7
BIER	78.0	85.8	91.1	95.1	97.3	98.7	72.7	86.5	94.0	98.0
Baseline	79.2	87.2	92.1	95.2	97.3	98.6	72.6	86.2	93.8	98.0
HTL(depth = 16)	81.4	88.0	92.7	95.7	97.4	99.0	74.8	88.3	94.8	98.4

5.4 LFW Face Verification

Datasets and Performance Measures. The *CASIA-WebFace* dataset [37] is one of the publicly accessible datasets for face recognition. It has been the most popular dataset for the training of face recognition algorithms, such as in [1, 14, 33]. *CASIA-WebFace* has 10575 identities and 494414 images. We following the testing protocol in [37] to test the performance of our algorithms. The face verification results on LFW dataset [8] is reported.

Implementation Details. Since the triplet loss is very sensitive to the noise, we clear the *CASIA-WebFace* using the pre-trained model of VGG-Face [20] and manually remove some noises. About 10% images are removed. Then the remained faces are used to train a SoftMax classifier. The network parameters are initialized by a pre-trained ImageNet model. We fine-tune the pre-trained classification network for face recognition using the hierarchical loss.

Result Comparison. The triplet loss gets 98.3% accuracy on the LFW face verification task, which is 1.12% lower than the SpereFace [14]—99.42% which uses the same dataset for training. When we substitute the triplet loss with the

hierarchical triplet loss, the results comes to 99.2. It’s comparable with state-of-art results. This indicates that the hierarchical triplet loss has stronger discriminative power than triplet loss. While, since the triplet based method are very sensitive to noise, the hierarchical triplet loss get inferior performance compared with SphereFace [14] 99.42% and FaceNet [22] 99.65%.

5.5 Sampling Matter and Local Optima

Sampling Matter. We investigate the influence of batch size on the test set of *In-Shop Clothes Retrieval*. Figure 5(a) shows that when the batch size grows from 60 to 480, the accuracy increases in the same iterations. When the training continues, the performance will fluctuates heavily and get overfitting. Besides, when come to the same results at 60% Recall@1, both the anchor-neighbor sampling with triplet loss and the hierarchical loss converge at about 2 times faster than random sampling (Batch Size = 480). Figure 5(b) shows the compares the convergence speed of the triplet loss (our baseline), the hierarchical triplet loss and the HDC [38] on the test set of *Caltech-UCSD Birds 200*. Compared to the 60000 iterations (see in [38]), the hierarchical triplet loss converges in 1000 iterations. The hierarchical triplet loss with anchor-Neighborhood sampling converge faster traditional and get better performance than HDC [38].

Pool Local Optima. In Tables 1 and 2, we can find that the triplet loss get inferior performance than the hierarchical triplet loss on both the *In-Shop Clothes Retrieval* and *Caltech-UCSD Birds 200*. In the Fig. 5, the accuracy of the triplet loss start to fluctuate when the training continues going after the loss drops to very low. In fact, there are always very few or zeros triplets in mini-batch even when the network isn’t gotten the best results. Then they don’t produce gradients and will decay the learnable parameters in networks by SGD [19]. So we incorporate the hierarchical structure to make points in the mini-batch know the position of point that are already far away, and then attempt to push them further from itself and its neighborhood classes.

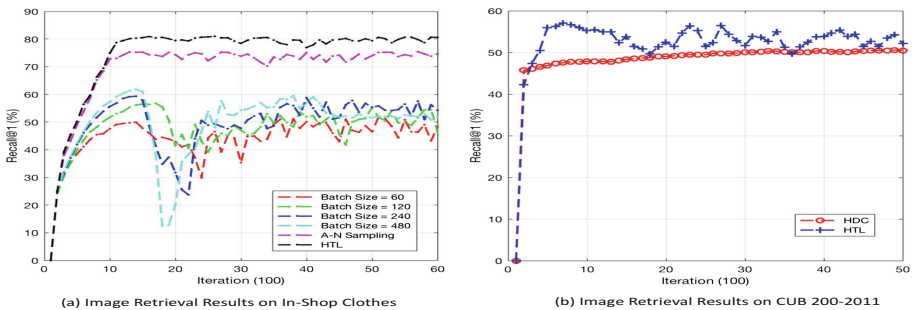


Fig. 5. (a) Image retrieval results on In-Shop Clothes [15] with various batch sizes. (b) Image retrieval results on CUB-200-2011 [31].

5.6 Ablation Study

We perform ablation studies on In-Shop Clothes and CUB-200-2011, as reported in Table 4. First, directly applying hard negative sampling (HNS) to the whole training set is difficult to obtain a performance gain. Actually, our baseline model applies a semi-HNS, which outperforms HNS. We design a strong class-level constrain - Anchor-Neighbor Sampling of HTL, which encourages the model to learn discriminative features from visual similar classes. This is the key to performance boost. Second, we integrated the proposed anchor-neighbor sampling and dynamic violate margin into HDC where a contrastive loss is used. As shown in Table 4 (bottom), HDC+ got an improvement of 7.3% R@1 on the In-Shop Clothes Retrieval, suggesting that our methods work practically well with a contrastive loss and HDC. Third, HTL with a depth of 16 achieves best performance at R@1 of 80.9%. This is used as default setting in all our experiments. We also include results of “flat” tree with depth = 1. Results suggest that the “flat” tree with the proposed dynamic violate margin improves the R@1 from 75.3% to 78.9%, and hierarchy tree improves it further to 80.9%.

Table 4. Ablation studies on In-Shop Clothes Retrieval and CUB-200-2011.

R@	In-shop clothes						CUB-200-2011					
	1	10	20	30	40	50	1	2	4	8	16	32
On triplets with sampling												
Random sampling	59.3	83.5	87.9	90.5	91.3	93.0	51.4	63.9	74.8	83.4	90.0	94.3
Hard negative mining	60.1	84.3	88.2	90.2	91.5	92.6	51.6	63.9	74.2	84.4	89.9	94.6
Semi-hard negative mining	62.3	85.1	89.0	91.1	92.4	93.4	55.9	68.4	78.2	86.0	92.2	95.5
Anchor-neighbor sampling (HTL)	75.3	91.8	94.3	96.2	96.7	97.5	56.4	68.5	78.5	86.2	92.4	95.5
HTL with A-N sampling + dynamic violate margin(α_z)												
Class proxy (flat/depth = 1)	78.9	93.4	94.8	96.0	96.5	97.5	56.0	68.1	78.2	86.2	92.3	95.5
HTL (depth = 8)	78.7	93.3	94.6	96.2	96.9	97.4	56.2	68.5	78.3	86.1	92.3	95.5
HTL (depth = 16)	80.9	94.3	95.8	97.2	97.4	97.8	57.1	68.8	78.7	86.5	92.5	95.5
HTL (depth = 32)	79.3	93.8	95.0	96.9	97.1	97.5	56.4	68.5	78.5	86.2	92.3	95.5
HDC+: contrastive loss with A-N sampling + dynamic violate margin(α_z)												
HDC	62.1	84.9	89.0	91.2	92.3	93.1	53.6	65.7	77.0	85.6	91.5	95.5
HDC+	69.4	88.6	93.4	94.1	95.3	96.5	54.1	66.3	77.2	85.6	91.7	95.5

6 Conclusion

We have presented a new hierarchical triplet loss (HTL) which is able to select informative training samples (triplets) via an adaptively-updated hierarchical tree that encodes global context. HTL effectively handles the main limitation of random sampling, which is a critical issue for deep metric learning. First, we construct a hierarchical tree at the class level which encodes global context information over the whole dataset. Visual similar classes are merged recursively to form the hierarchy. Second, the problem of triplet collection is formulated by

proposing a new violate margin, which is computed dynamically based on the designed hierarchical tree. This allows it to learn from more meaningful hard samples with the guide of global context. The proposed HTL is evaluated on the tasks of image retrieval and face recognition, where it achieves new state-of-the-art performance on a number of standard benchmarks.

References

1. Amos, B., Ludwiczuk, B., Satyanarayanan, M.: Openface: a general-purpose face recognition library with mobile applications. CMU School of Computer Science (2016)
2. Bai, S., Bai, X., Tian, Q., Latecki, L.J.: Regularized diffusion process for visual retrieval. In: AAAI, pp. 3967–3973 (2017)
3. Bai, S., Zhou, Z., Wang, J., Bai, X., Latecki, L.J., Tian, Q.: Ensemble diffusion for retrieval. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 774–783 (2017)
4. Bucher, M., Herbin, S., Jurie, F.: Hard negative mining for metric learning based zero-shot classification. In: Hua, G., Jégou, H. (eds.) ECCV 2016. LNCS, vol. 9915, pp. 524–531. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-49409-8_45
5. Chen, W., Chen, X., Zhang, J., Huang, K.: Beyond triplet loss: a deep quadruplet network for person re-identification. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017
6. Hadsell, R., Chopra, S., LeCun, Y.: Dimensionality reduction by learning an invariant mapping. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 1735–1742. IEEE (2006)
7. Harwood, B., Kumar, B.G.V., Carneiro, G., Reid, I., Drummond, T.: Smart mining for deep metric learning. In: The IEEE International Conference on Computer Vision (ICCV), October 2017
8. Huang, G.B., Ramesh, M., Berg, T., Learned-Miller, E.: Labeled faces in the wild: a database for studying face recognition in unconstrained environments. Technical report 07–49, University of Massachusetts, Amherst, October 2007
9. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. In: International Conference on Machine Learning, pp. 448–456 (2015)
10. Jia, Y., et al.: Caffe: convolutional architecture for fast feature embedding. In: Proceedings of the 22nd ACM International Conference on Multimedia, pp. 675–678. ACM (2014)
11. Krause, J., Stark, M., Deng, J., Fei-Fei, L.: 3D object representations for fine-grained categorization. In: 4th IEEE Workshop on 3D Representation and Recognition, ICCV (2013)
12. Kumar, B., Carneiro, G., Reid, I., et al.: Learning local image descriptors with deep siamese and triplet convolutional networks by minimising global loss functions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5385–5394 (2016)
13. van Lint, J.H., Wilson, R.M.: A Course in Combinatorics. Cambridge University Press, Cambridge (2001)
14. Liu, W., Wen, Y., Yu, Z., Li, M., Raj, B., Song, L.: Sphereface: deep hypersphere embedding for face recognition. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol. 1 (2017)

15. Liu, Z., Luo, P., Qiu, S., Wang, X., Tang, X.: Deepfashion: powering robust clothes recognition and retrieval with rich annotations. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1096–1104 (2016)
16. van der Maaten, L., Hinton, G.: Visualizing data using t-SNE. *J. Mach. Learn. Res.* **9**(Nov), 2579–2605 (2008)
17. Opitz, M., Waltner, G., Possegger, H., Bischof, H.: Bier - boosting independent embeddings robustly. In: The IEEE International Conference on Computer Vision (ICCV), October 2017
18. Opitz, M., Waltner, G., Possegger, H., Bischof, H.: Bier-boosting independent embeddings robustly. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5189–5198 (2017)
19. Orr, G.B., Müller, K.R.: *Neural Networks: Tricks of the Trade*. Springer, Heidelberg (2003). <https://doi.org/10.1007/978-3-642-35289-8>
20. Parkhi, O.M., Vedaldi, A., Zisserman, A., et al.: Deep face recognition. In: *BMVC*, vol. 1, p. 6 (2015)
21. Russakovsky, O., et al.: ImageNet large scale visual recognition challenge. *Int. J. Comput. Vis. (IJCV)* **115**(3), 211–252 (2015). <https://doi.org/10.1007/s11263-015-0816-y>
22. Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: a unified embedding for face recognition and clustering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 815–823 (2015)
23. Shi, H., et al.: Embedding deep metric for person re-identification: a study against large variations. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) *ECCV 2016*. LNCS, vol. 9905, pp. 732–748. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_44
24. Simo-Serra, E., Trulls, E., Ferraz, L., Kokkinos, I., Fua, P., Moreno-Noguer, F.: Discriminative learning of deep convolutional feature point descriptors. In: 2015 IEEE International Conference on Computer Vision (ICCV), pp. 118–126. IEEE (2015)
25. Sohn, K.: Improved deep metric learning with multi-class n-pair loss objective. In: *Advances in Neural Information Processing Systems*, pp. 1857–1865 (2016)
26. Song, H.O., Xiang, Y., Jegelka, S., Savarese, S.: Deep metric learning via lifted structured feature embedding. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4004–4012. IEEE (2016)
27. Sun, Y., Chen, Y., Wang, X., Tang, X.: Deep learning face representation by joint identification-verification. In: *Advances in Neural Information Processing Systems*, pp. 1988–1996 (2014)
28. Szegedy, C., et al.: Going deeper with convolutions. In: *CVPR* (2015)
29. Tao, R., Gavves, E., Smeulders, A.W.: Siamese instance search for tracking. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1420–1429. IEEE (2016)
30. Ustinova, E., Lempitsky, V.: Learning deep embeddings with histogram loss. In: *Advances in Neural Information Processing Systems*, pp. 4170–4178 (2016)
31. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: *The Caltech-UCSD birds-200-2011 dataset* (2011)
32. Waltner, G., Opitz, M., Bischof, H.: BaCoN: building a classifier from only n samples. In: *Proceedings of CVWW*, vol. 1 (2016)
33. Wen, Y., Zhang, K., Li, Z., Qiao, Y.: A discriminative feature learning approach for deep face recognition. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) *ECCV 2016*. LNCS, vol. 9911, pp. 499–515. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46478-7_31

34. Wohlhart, P., Lepetit, V.: Learning descriptors for object recognition and 3D pose estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3109–3118 (2015)
35. Wu, C.Y., Manmatha, R., Smola, A.J., Krahenbuhl, P.: Sampling matters in deep embedding learning. In: The IEEE International Conference on Computer Vision (ICCV), October 2017
36. Wu, C.Y., Manmatha, R., Smola, A.J., Krähenbühl, P.: Sampling matters in deep embedding learning. arXiv preprint [arXiv:1706.07567](https://arxiv.org/abs/1706.07567) (2017)
37. Yi, D., Lei, Z., Liao, S., Li, S.Z.: Learning face representation from scratch. arXiv preprint [arXiv:1411.7923](https://arxiv.org/abs/1411.7923) (2014)
38. Yuan, Y., Yang, K., Zhang, C.: Hard-aware deeply cascaded embedding. In: The IEEE International Conference on Computer Vision (ICCV), October 2017