Sampling Wisely: Deep Image Embedding by Top-k Precision Optimization

Jing Lu[1*] Chaofan Xu[2,3*] Wei Zhang[2] Lingyu Duan[4] Tao Mei[2]

[1]Business Growth BU, JD [2] JD AI Research [3]Harbin Institute of Technology [4]Peking University

lvjing12@jd.com,xuchaofan1994@126.com,wzhang.cu@gmail.com,lingyu@pku.edu.cn,tmei@live.com

Abstract

Deep image embedding aims at learning a convolutional neural network (CNN) based mapping function that maps an image to a feature vector. The embedding quality is usually evaluated by the performance in image search tasks. Since very few users bother to open the second page search results, top-k precision mostly dominates the user experience and thus is one of the crucial evaluation metrics for the embedding quality. Despite being extensively studied, existing algorithms are usually based on heuristic observation without theoretical guarantee. Consequently, gradient descent direction on the training loss is mostly inconsistent with the direction of optimizing the concerned evaluation metric. This inconsistency certainly misleads the training direction and degrades the performance. In contrast, in this paper, we propose a novel deep image embedding algorithm with end-to-end optimization to top-k precision, the evaluation metric that is closely related to user experience. Specially, our loss function is constructed with Wisely Sampled "misplaced" images along the top-k nearest neighbor decision boundary, so that the gradient descent update directly promotes the concerned metric, top-k precision. Further more, our theoretical analysis on the upper bounding and consistency properties of the proposed loss supports that minimizing our proposed loss is equivalent to maximizing top-k precision. Experiments show that our proposed algorithm outperforms all compared state-of-the-art deep image embedding algorithms on three benchmark datasets.

1. Introduction

Deep image embedding is a fundamental component for a wide range of applications, such as image clustering [9], visual product retrieval [41], face verification and identification [34, 25], object tracking [17], etc. The aim is to learn a CNN based mapping function f that maps an image z to a compact feature vector $f(z)$ while preserving the semantic distance. Namely, similar images should be embedded close to each other while dissimilar images should be pushed far away. Without loss of generality, the embedding quality is mostly evaluated by the performance in a visual search task [33, 22]. Since very few users bother to open the second page search results, *top-k precision* (Prec@k for short) usually dominates the user experience. Consequently, Prec@k has been considered as one of the crucial evaluation metrics for the embedding quality.

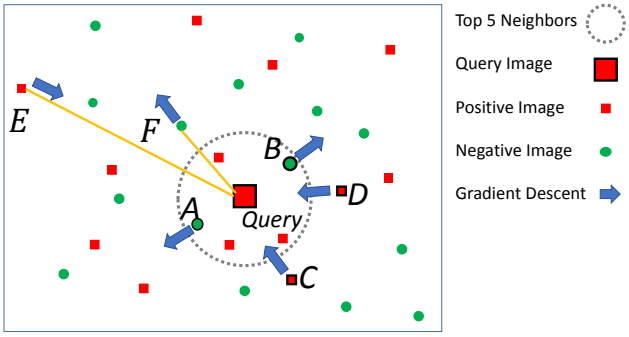


Figure 1. A toy example to illustrate our motivation. Consider a task of learning a deep image embedding model, where top 5 precision is the concerned evaluation metric. Conventional algorithms sample training images without focus on the top 5 nearest neighbor decision boundary. Since the number of candidates (batch size) is much larger than 5, the probability of sampling just besides the decision boundary is extremely small. Assuming triplet $\{Query, E, F\}$ is sampled. The gradient descent might improve the embedding but does not directly promote top-5 precision. In contrast, the proposed algorithm wisely selects the *misplaced* images besides the boundary: A, B that should be out of the boundary and C, D that should be in the boundary. The gradient descent pushes A, B, C, D in the direction of directly promoting top-5 precision from $\frac{3}{5}$ to $\frac{5}{5}$.

Recent years have witnessed a variety of emerging studies for deep image embedding. Examples include con-

*Equal Contribution. This work was done when the two authors were in JD AI Research, Beijing, China.

trastive loss [4], triplet loss [24], lifted loss [29], Position-Dependent Deep Metric [10], N-Pair Loss [27], Angular Loss [35], etc. These loss functions are usually defined over tuplets of images and encourage large similarity between images from same class and small similarity between images from different classes. A practical concern is that the effectiveness of some loss functions largely depends on the sampling strategy. Uniform sampling among training images often results in nearly zero gradient and thus terrible convergence. This motivates the studies of sampling methods in deep image embedding, including hard negative mining [26], semi-hard negative mining [25], distance weighted sampling [38], etc.

Despite being studied actively, most of the existing training losses or sampling strategies are based on heuristic observations instead of theoretical analysis. Thus, the gradient descent on the training loss is mostly inconsistent with the direction of optimizing the concerned evaluation metric. This inconsistency certainly misleads the training direction and degrades the overall performance, as shown in the toy example in Figure 1. Specially, Prec@k is closely related to user experience and thus is one of the most widely used metric for evaluating embedding quality. While to the best of our knowledge, no existing deep image embedding algorithm optimizes Prec@k as the training direction. We thus conjecture that the state-of-the-art performance could be further enhanced if models are trained in the consistent direction with Prec@k optimization.

In contrast to existing approaches, in this paper, we present a novel deep image embedding algorithm with end-to-end optimization to Prec@k. Our key idea is to construct the loss function with wisely selected images, the *misplaced* images besides the decision boundary of top-k nearest neighbor in a visual search task. In particular, misplaced images are: 1) ones similar to the query but ranked just out of the top-k boundary; 2) ones dissimilar to the query but ranked just in the top-k boundary. This motivation is shown in the toy example in Figure 1. Further more, we give theoretical analysis on the upper bounding and consistency properties which supports the equivalence of minimizing our proposed loss and optimizing Prec@k.

In summary, we make the following contributions:

- To the best of our knowledge, we are the first to highlight the negative impact of the inconsistency between the gradient descent direction and the direction of optimizing the concerned evaluation metric.
- We propose a novel deep image embedding algorithm that directly optimize Prec@k, which can be well aligned with user experience.
- We provide convincing theoretical analysis for the equivalence of minimizing our proposed loss function and optimizing Prec@k.
- The proposed algorithm outperforms all compared state-of-the-art algorithms on 3 benchmark datasets.

2. Related Work

Our work is related to two active research areas: deep image embedding and top rank optimization.

2.1. Deep Image Embedding

Deep image embedding learns a CNN based mapping function that maps an image to a compact feature vector while preserving the semantic distances.

The loss functions are usually defined over tuplets of images and penalizes small similarity between images from the same class and large similarity between images from different classes. Examples include the contrastive loss [4], triplet loss [24], lifted loss [29], PDDM [10], N-Pair Loss [27], Clustering Loss [28], Angular Loss [35], Histogram loss [31], among others [1, 40, 7].

Sampling method is also an important research topic, since uniformly sampled triplets mostly contribute minor to the loss and gradient. To acquire informative triplets, many sampling methods are explored, including hard negative mining [26], semi-hard negative mining [25], distance weighted sampling [38], etc.

Recent research focus is moving from loss designing and sampling to ensemble models, whose research question is not what loss to train, but how to achieve independency in ensemble components. In HDC [41] and BIER [23], the independency is from boosting over images with different hardness levels. Others achieve independency through randomly bagging of labels [39] and spatial attention [14].

Two recent representative works address deep image embedding by optimizing ranking losses [3, 36]. While these two optimize the overall ranking (e.g. average precision), our Prec@k loss focuses only on the top page, which is more related to user experience.

Our work is closely related to the first 2 groups of research. Existing algorithms are usually based on heuristic observation. Gradient descent direction on the training loss is mostly inconsistent with the direction of optimizing the concerned evaluation metric. In contrast to existing works, we propose a novel deep image embedding algorithm with end-to-end optimization to Prec@k and clear theoretical guarantee. Although we did not contribute directly to ensemble methods, our function can be easily adapted to many ensemble algorithms.

2.2. Top Rank Optimization

Our work is also closely related to top rank optimization [16]. Many existing works have attempted to solve this problem in different settings, e.g., binary classification [6], single label multiclass classification [2] and embedding

or distance metric learning [19, 8]. [19] maximizes the number of positive instances ranked before the first negative instance, termed "positives at the top". [2] optimizes the top k recall of multiple class labels assuming that only single label is correct. Some algorithms define new loss functions that assign higher weights to the top positions [37]. Compared with those above, optimizing exactly the Prec@k is more challenging due to its discontinuity and non-differentiable.

One of our closely related works is Ramp Surrogate [13], a pioneer work that maximizes Prec@k. But our work fundamentally differs in the feasible range of problems. Specially, our work is applicable to problems with any class distribution. While Ramp is applicable only when any class has no smaller than k images. This is usually impossible in most real-world image embedding problems, where a great many of classes only have few training images due to the high cost of collecting images from rare classes. Besides, [13] focuses on convex losses for linear classification /ranking, while we revisit Prec@k maximization to find the optimal sampling strategy in deep image embedding.

3. Embedding for Maximizing Prec@k

In this section, we first review the problem setting of deep image embedding and some classical sampling methods. Then we present our proposed loss function and highlight our advantage with theoretical analysis.

3.1. Preliminaries

The aim of deep image embedding is to learn a CNN based mapping function $f(\cdot)$ that maps an image z to a compact feature vector $f(z) \in \mathbb{R}^d$ while preserving the semantic similarities. Semantically similar images have a higher similarity score than semantically dissimilar ones. Specially, we adopt the cosine similarity, $s_{i,j} = \frac{f(z_i)^\top f(z_j)}{||f(z_i)||||f(z_j)||}$ as the similarity score between image z_i and image z_j [1].

The triplet loss is trained on triplet $\{z_a, z_p, z_n\}$, referred as *Anchor*, *Positive* and *Negative*. The positive pair $\{z_a, z_p\}$ have same class label and negative pair $\{z_a, z_n\}$ have different class labels. Triplet loss encourages positive pairs to have higher similarity scores than negative pairs, i.e.

$$\ell^{\text{triplet}}(z_a, z_p, z_n) = [s_{a,n} - s_{a,p} + \gamma]_+, \tag{1}$$

where $\gamma > 0$ is the margin parameter. Since passing through losses over all triplets is computationally infeasible, many sampling methods are proposed to address this problem.

Uniform sampled triplets usually contributes minor to the loss and thus to gradient, which results in terrible convergence. To address this problem and acquire informative triplets, hard mining methods sample pairs with lowest $s_{a,p}$ or highest $s_{a,n}$. But this is also problematic since many mined pairs are not really hard, but noisy. Here comes the open question: *which instances are most suitable to be sampled?* In literature, some methods addresses this issue, including semi-hard mining, distance weighted sampling, etc.

Despite being studied extensively, existing sampling methods are usually based on heuristic observations. The gradient descent direction in the training process is mostly inconsistent with the direction of optimizing the concerned metrics. So there is no theoretical guarantee that the update using sampled triplets will improve the concerned metrics.

In this paper, we propose a novel image embedding algorithm that samples wisely for the images that is able to directly promotes Prec@k. This is motivated by the facts that 1) very few users bother to open the second page search results, 2) thus Prec@k is closely related to the user experience, 3) and Prec@k is one of the most widely used evaluation metrics for embedding quality.

3.2. Prec@k Maximization

Without loss of generality, the embedding quality is mostly evaluated by the performance in a visual search task. Given a query image z_a and a candidate images set $\mathcal{C} = \{z_1, ..., z_n\}$, we first calculate the embedding features $f(z_a), f(z_1), ..., f(z_n)$ and then measure the similarity $\mathbf{s} \in \mathbb{R}^n$ between z_a and n candidates, where the i-th element s_i denotes the similarity score between query z_a and candidate z_i. We further define $\mathbf{y} \in \mathcal{Y} = \{0, 1\}^n$ as the ground truth label vector, where $y_i = 1$ iff z_a and z_i have the same class label.

Formally, Prec@k is defined as the fraction of positive instances in the top k positions, i.e.,

$$\text{Prec@}k(\mathbf{s}, \mathbf{y}) = \frac{\sum_{i=1}^n y_i \mathbb{I}_{[s_i \geq s_{[k]}]}}{k}, \tag{2}$$

where k is a small constant, usually set to 3, 5, 10, $s_{[k]}$ denotes the k-th largest element of vector $\mathbf{s}$, and $\mathbb{I}_{[A]}$ is an indicator with the value 1 if the A is true, and 0 otherwise.

We further define the precision loss,

$$\ell_{\text{Prec@}k}(\mathbf{s}, \mathbf{y}) = \sum_{i=1}^{n} (1 - y_i) \mathbb{I}_{[s_i \geq s_{[k]}]}, \tag{3}$$

which indicates the number of negative images in the k top ranked candidates. Obviously, minimizing $\ell_{\text{Prec@}k}$ is equivalent to maximizing Prec@k. But optimizing $\ell_{\text{Prec@}k}$ is still challenging since it is complicated and incontinuous.

In this work, we construct our proposed loss function with wisely selected training images. And provide sufficient theoretical analysis to show that optimizing our objective function is equivalent to maximizing Prec@k..

[1] Euclidean distance is also widely used in literature. Actually, when vectors are scaled to the same norm, large cosine similarity is equivalent to small Euclidean distance.

3.3. The Loss Function

Following the original design of triplet loss, our loss function also promotes large margin. Namely, we are not satisfied when positive images have higher similarity scores than negative ones, i.e. $s_{i:y_i=1} > s_{j:y_j=0}$, but encourage a large margin between them, $s_{i:y_i=1} > s_{j:y_j=0} + \gamma$. For conciseness, we define the large margin similarity score vector,

$$\hat{\mathbf{s}} = \mathbf{s} + \gamma(1 - \mathbf{y}). \tag{4}$$

The large margin requirement can be simplified as a direct comparison between elements of $\hat{\mathbf{s}}$, i.e. $\hat{s}_{i:y_i=1} > \hat{s}_{j:y_j=0}$. We thus will use the ranking of $\hat{s}_i$ instead of s_i in the rest of this paper. We define set $\mathcal{K}$ as the set of top k ranked images according to $\hat{s}_i$, i.e. $\mathcal{K} = \{z_i \in \mathcal{C} : \hat{s}_i \geq \hat{s}_{[k]}\}$.

Different from the triplet loss, our loss is not defined on a single positive z_p and a single negative z_n, but on two *sets* of positive and negative images selected from the candidate set, denoted as $\mathcal{P} \subset \mathcal{C}, \mathcal{N} \subset \mathcal{C}$. The loss is defined as

$$\ell_k(\mathbf{s}, \mathbf{y}) = \sum_{z_i \in \mathcal{N}} \hat{s}_i - \sum_{z_i \in \mathcal{P}} \hat{s}_i. \tag{5}$$

Our key question is: which images to choose to form $\mathcal{P}, \mathcal{N}$?

We define the *ideal* solution as the embedding with similarity score $\mathbf{s}$ that minimizes the precision loss $\mathbf{s} = \arg\min \ell_{\text{Prec@}k}(\mathbf{s}, \mathbf{y})$. Note that there may be more than one ideal solution. Intuitively, an ideal ranking should fill positive images into $\mathcal{K}$ and push negative images out of $\mathcal{K}$. Given a similarity ranking of $\hat{s}_i$, we set $\mathcal{P}$ and $\mathcal{N}$ as the set of "*misplaced*" images compared with an ideal ranking.

3.3.1 Case 1: not enough positive candidates, $n_+ < k$

We first consider the case where not enough positive candidates are available in $\mathcal{C}$ to fill up $\mathcal{K}$. This often occurs in practice, since collecting large number of images for each rare class is costly. In some benchmark datasets (e.g. Online Products [29]), many classes only have few training images.

To minimize $\ell_{\text{Prec@}k}$, an ideal solution should rank all n_+ positive images and any $k - n_+$ negative images in $\mathcal{K}$. But the order inside $\mathcal{K}$ makes no difference to $\ell_{\text{Prec@}k}$.

So, given a ranking, these positive images ranked out $\mathcal{K}$ are misplaced,

$$\mathcal{P} = \{z_i \in \mathcal{C} \backslash \mathcal{K} : y_i = 1\}. \tag{6}$$

where $\mathcal{C}\backslash\mathcal{K}$ means the relative complement set of $\mathcal{K}$ in $\mathcal{C}$, i.e. the set of candidates out of top k. We can easily get this reasoning: among the total n_+ positive images, the other $n_+ - |\mathcal{P}|$ are in $\mathcal{K}$. So the number of negative images in $\mathcal{K}$ is $k - (n_+ - |\mathcal{P}|)$. Note that not all negative images in $\mathcal{K}$ are misplaced, since even in an ideal ranking, there are still $k - n_+$ negative in $\mathcal{K}$.

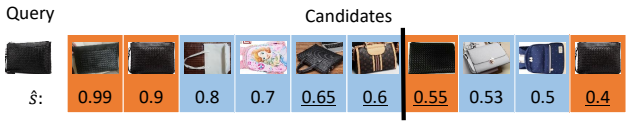


Figure 2. A toy example of loss calculation when $n_+ < k$. Here $k = 6, n_+ = 4, n = 10$. For easy illustration, we re-order the candidates in descending order of $\hat{s}_i$. The orange elements are positive and the blue are negative. The misplaced positive set $\mathcal{P}$ consists of all positive images out of top k, $\sum_{z_i \in \mathcal{P}} \hat{s}_i = 0.55 + 0.4$, and $|\mathcal{P}| = 2$. The last 2 positive images in top k wrongly take the place of $\mathcal{P}$, which is denoted as set $\mathcal{N}$. $\sum_{z_i \in \mathcal{N}} \hat{s}_i = 0.65 + 0.6$. Note that the first two negative images (with similarity score 0.8 and 0.7) are not considered as misplaced ones, because even in ideal ranking, there should also be $k - n_+ = 2$ negative images in top k. In conclusion, $\ell_k = 0.65 + 0.6 - (0.55 + 0.4)$.

So among the $k - (n_+ - |\mathcal{P}|)$ negative images in $\mathcal{K}$, which $k - n_+$ are regarded as properly placed and which $|\mathcal{P}|$ are misplaced? We follow a commonly used principle in optimization, *minimal* necessary update, to minimize the *forgotten* of previously learnt knowledge due to each update [5]. Obviously, pushing candidates just beside the top-k boundary is a smaller change than pushing others.

We thus consider the first $k - n_+$ as properly placed, and the later $|\mathcal{P}|$ as misplaced. So,

$$\mathcal{N} = \{z_i \in \mathcal{K} : y_i = 0, \hat{s}_i < \hat{s}^-_{[k-n_+]}\}. \tag{7}$$

where $\hat{\mathbf{s}}^- \in \mathbb{R}^{n-n_+}$ is a sub-vector of $\hat{\mathbf{s}}$ containing only the similarity scores of negative images. An example in Fig 2.

We now highlight our key advantage compared with existing sampling methods for triplet loss. Our choice of images in $\mathcal{P}$ and $\mathcal{N}$ for loss calculation is not heuristic but with clear theoretical guarantee. From the optimization perspective, minimizing our proposed loss function ℓ_k is equivalent to minimizing $\ell_{\text{Prec@}k}$. This claim is supported by two properties of ℓ_k: *upper bounding* and *consistency*. [2]

Theorem 1. ***Upper bounding****: For any $n_+ < k$ and $\mathbf{s}$,*

$$\ell_k(\mathbf{s}, \mathbf{y}) \geq \gamma \ell_{Prec@k}(\mathbf{s}, \mathbf{y}) - \gamma(k - n_+) \tag{8}$$

Remark 1: The constant term $k - n_+$ is the optimal value of $\ell_{\text{Prec@}k}$, reached by ideal solutions. Note that adding a constant does not affect the optimization process.

Theorem 2. ***Consistency****: For any $n_+ < k$, when there is a large margin γ between positive images and negative images that should be out of $\mathcal{K}$ (the $k - n_+ + 1$-th ranked negative image), i.e. $s^+_{[n_+]} - s^-_{[k-n_++1]} \geq \gamma$, we have $\ell_k(\mathbf{s}, \mathbf{y}) = \ell_{Prec@k}(\mathbf{s}, \mathbf{y}) - (k - n_+) = 0$. Here $\mathbf{s}^+ \in \mathbb{R}^{n_+}$ and $\mathbf{s}^- \in \mathbb{R}^{n-n_+}$ are two sub-vectors of $\mathbf{s}$ containing the similarity scores of positive and negative images.*

[2] All proofs are provided in Supplementary Materials.

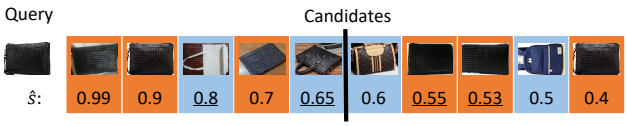


Figure 3. An example of our loss when $n_+ \geq k$. Here $k = 5, n_+ = 6, n = 10$. All negative in top 5 are misplaced, so $\sum_{z_i \in \mathcal{N}} \hat{s}_i = 0.8 + 0.65$ and $|\mathcal{N}| = 2$. In all positive candidates out of top 5, only the first 2 are misplaced. $\sum_{z_i \in \mathcal{P}} \hat{s}_i = 0.55 + 0.53$. So, $\ell_k = (0.8 + 0.65) - (0.55 + 0.53)$.

Remark 2: This theorem indicates that there exists an optimal solution $\mathbf{s}$ that minimizes both ℓ_k and $\ell_{\text{Prec@}k}$ simultaneously. Combining with the upper bounding property, we conclude that optimizing our proposed loss is equivalent to minimizing the original precision loss, which demonstrates our clear advantage over existing sampling methods.

3.3.2 Case 2: enough positive candidates, $n_+ \geq k$

An ideal solution should fill up $\mathcal{K}$ with k positive candidates and left the other $n_+ - k$ positive out of $\mathcal{K}$. Thus given a ranking, all negative in $\mathcal{K}$ are misplaced:

$$\mathcal{N} = \{z_i \in \mathcal{K} : y_i = 0\} \tag{9}$$

In $\mathcal{K}$ the other $k - |\mathcal{N}|$ are all positive. So the number of positive out of $\mathcal{K}$ is $n_+ - (k - |\mathcal{N}|)$. We regard the top $|\mathcal{N}|$ of them as misplaced,

$$\mathcal{P} = \{z_i \in \mathcal{C} \backslash \mathcal{K} : y_i = 1, \hat{s}_i \geq \hat{s}^+_{[k]}\} \tag{10}$$

where $\hat{\mathbf{s}}^+ \in \mathbb{R}^{n_+}$ is a sub-vector of $\hat{\mathbf{s}}$ for the similarity scores of positive images. An example is in Figure 3. Upper bounding and consistency still hold in this case. [3]

In practice, the two sampling strategies in Case 1&2 work together. During training, each image in the current batch takes turns to be the query z_a. The algorithm is summarized in Algorithm 1.

3.3.3 Our Advantages

• **Low time complexity**. For case 1, we rank the top k from n candidates in $O(n \log k)$. For case 2, to find $\mathcal{P}$, we need an additional ranking of positive images in time $O(n_+ \log n_+)$. In summary, the time complexity is no larger than $O(n \log \max(n_+, k))$.

• **Semi-hard mining**. By sampling $\mathcal{P}$ and $\mathcal{N}$ besides the top-k boundary, we implicitly mines semi-hard instances. For example, the hardest negative 0.8 and 0.7 in Fig 2 and the hardest positive 0.4 in Fig 3 are not sampled, which avoids noisy and unstable gradient.

[3] Proof in the Supplementary Materials.

Algorithm 1 Sampling Wisely for Deep Image Embedding

Receive a batch $z_1, ..., z_{n+1}$, with class labels.
Calculate embedding $f(z_1), f(z_1), ..., f(z_{n+1})$.
for $j = 1, ..., n+1$ **do**
  Assign z_j as the query image z_a, other n images as candidate set $\mathcal{C}(z_a)$. Get label $\mathbf{y}(z_a) \in \{0, 1\}^n$ from class labels. $n_+(z_a) = ||\mathbf{y}(z_a)||$.
  if $n_+(z_a) < k$ **then**
    Sample $(\mathcal{P}, \mathcal{N})$ as Case 1.
  else
    Sample $(\mathcal{P}, \mathcal{N})$ as Case 2.
  end if
  Forward, calculate $[\ell_k]_j$ as Eq. (5).
end for
Sum up $\ell_k = \sum_j^{n+1} [\ell_k]_j$, Backward, update f.

• **Prec@k maximization**. Our selected candidates directly promotes Prec@k which is a widely used metric for embedding quality evaluation and closely related to user experience. This is a clear advantage over algorithms where training loss and the real evaluation metric are inconsistent.

4. Experiments

We conduct extensive experiments to examine the proposed deep image embedding algorithm on *image retrieval* and *clustering* tasks. The algorithms are implemented in Pytorch and are publicly available at `https://github.com/BG2CRW/top_k_optimization`.

4.1. Benchmark Datasets

CUB-200-2011 [32] has 200 species of birds with 11,788 images. We split 100 species (5,864 images) for training and 100 species (5,924 images) for testing.

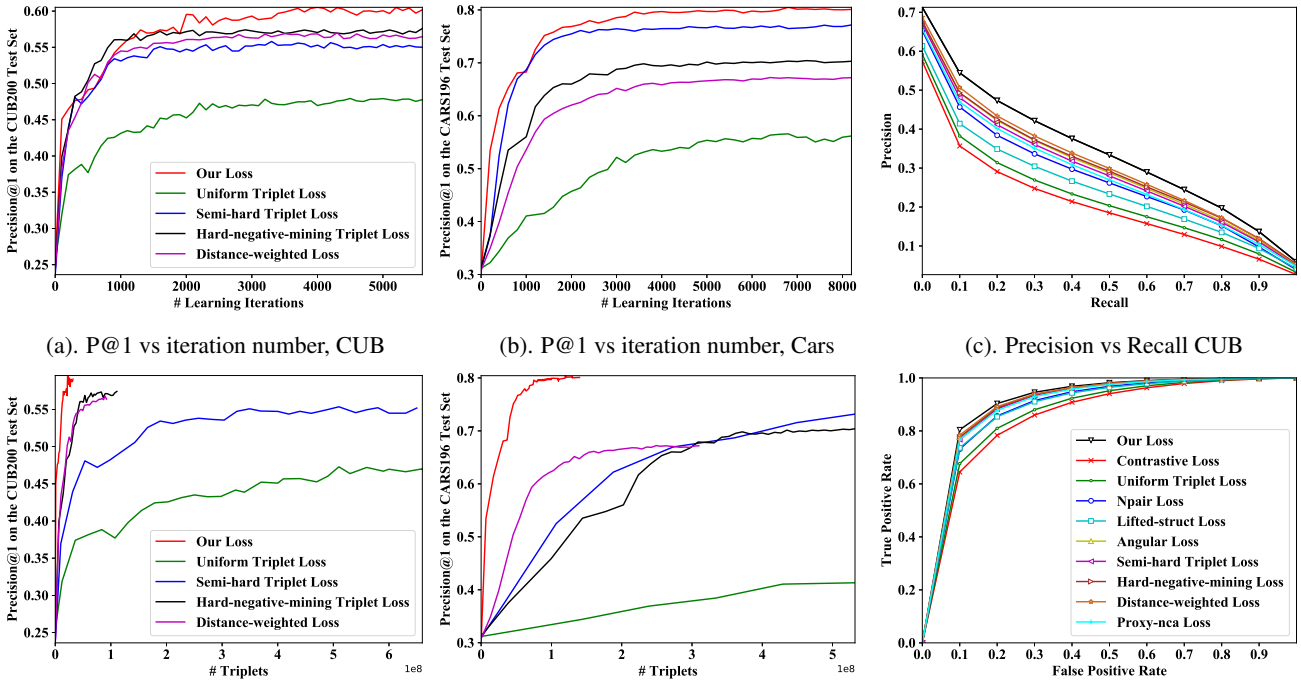
Stanford Cars [15] is composed by 16,185 cars images of 196 classes, where the first 98 classes (8,054 images) for training and the other 98 classes (8,131 images) for testing.

Stanford Online Products [29] has 120,053 images of 22,634 online products (classes) from eBay.com. We split 11,318 classes (59,551 images) for training and the other 11,316 classes (60,502 images) for testing.

4.2. Compared Algorithms

First, our superiority is in that we wisely select informative images besides the decision boundary. To exam this superiority, we compare with triplet sampling methods including, **uniform sampling**, **hard mining** [26], **semi-hard negative mining** [25] and **distance weighted sampling** [38].

Second, we compare with state-of-the-art loss functions for deep image embedding, including **Contrastive Loss** [29], **Triplet Loss** [24], **Lifted Structure Loss** [29], **N-Pair Loss** [27], **Angular Loss** [35], **Proxy NCA Loss** [21].



(a). P@1 vs iteration number, CUB (b). P@1 vs iteration number, Cars (c). Precision vs Recall CUB

(d). P@1 vs triplet number, CUB (e). P@1 vs triplet number, Cars (f). ROC, CUB

Figure 4. The top-1 precision on test data along the training process (a,b,d,e). Precision vs Recall, ROC (c, f). We outperform all baselines. Additional figures are in Supplementary Materials. a&d (b&e) are results from one single run. We zoomed in to the left for clear illustration.

4.3. Evaluation Metrics

For the retrieval task, we test on our target metric, Precision at top 1, 3, 5, 10. For comprehensive comparison, we also report other widely used metrics, mAP, Precision vs Recall curve, ROC curve, Recall [18] at top 1, 3, 5, 10. Recall@k=1 if any positive is ranked in top k and 0 otherwise, which is a much easier metric than precision.

We test k-means clustering with NMI and F1 score [29] on the embedded features. For intuitive demonstration, we also show the t-SNE [20].

4.4. Implementation Details

We used the PyTorch framework for all methods and follow implementation details of [21][4]. We test our loss function on two network backbones, the Inception with BN layer [30, 12] and Densenet201[11]. We use a fully connected layer as embedding layer and normalize its output. All models are first pretrained on ILSVRC 2012-CLS [5] , and then finetuned on the benchmark datasets. The embedding dimension is 64 in [32] and [15] and 512 in [29]. The inputs images are resized to 256x256 and then randomly cropped to 227x227. The numbers reported in [27] used multiple random crops during testing, but for fair comparison with other methods and following the procedure in [28], we only center crop during test. We use the ADAM optimizer. A training batch contains 64 images from randomly sampled classes. For large classes, we randomly sample 11 images for the batch, so $n_+ = 10$ (excluding 1 for query). For small classes with less than 11 images, we sample them all. All images takes turns as query z_a and all others in this batch form candidate set $\mathcal{C}$. We set $\gamma = 0.1$ and $k = 5$.

[4] https://github.com/dichotomies/proxy-nca

[5] http://image-net.org/challenges/LSVRC/2012

4.5. Comparison with Different Sampling Methods

Our advantage can be explained from the sampling perspective: to select images wisely from the candidates. We thus compare our proposed algorithm with many widely used sampling strategies. Results in Figure 4 and Table 1.

We show the top-1 accuracy on the test set in comparison with the 4 sampling strategies alone the training process. To evaluate the effectiveness of sampling strategies, we use the number of sampled triplets as the x-axis of the curves. For the proposed algorithms, we use $|\mathcal{P}| = |\mathcal{N}|$ as the number of triplets for fair comparison. For comparison on the convergence speed, we also plot the test performance vs the training iterations. We can draw several observations.

First, our proposed algorithm significantly outperforms all baseline algorithms, which validates our effectiveness. Among the four baselines, the triplet loss with uniform sampling always performs the worst, supporting the sig-

Figure 5. Example of our retrieval results on online product dataset. The left images are the queries and right images are candidates ranked in descending order of the similarities to the query. Successful cases (in green boxes) include photos of the same objects taken from different directions. Most failure cases (in red boxes) are from fine grained sub-categories.

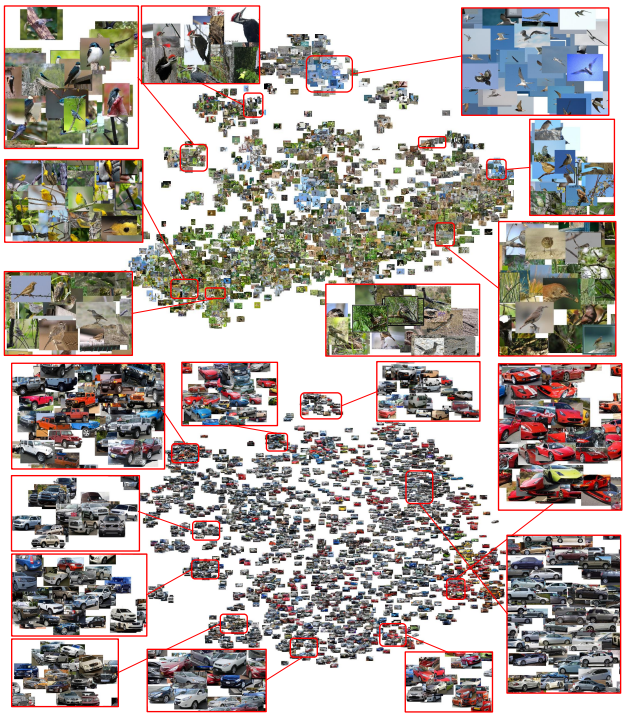


Figure 6. Barnes-Hut t-SNE of our embedding on the test split of CUB (top) and Standard Cars (bottom). The embedding generated by the proposed algorithm put similar images in clusters.

nificance of exploring wise candidate selection strategies. Among the other 3 sampling methods, no one always wins. This is due to the difference in dataset properties, including class imbalance, noisy labels, etc.

Second, when same number of triplets are sampled, our algorithm achieves the best embedding quality. This is because we select images with direct promotion to top precision. Other sampling strategies may waste gradient update on images far away from the decision boundary.

Third, our efficiency is not sacrificed for effectiveness. When all algorithms run for the same number of iterations, given that all algorithms adopt the same batch size.

4.6. Comparison with State-of-the-art Embedding Algorithms

We evaluate the proposed algorithm on image retrieval and clustering tasks in comparison with state-of-the-art embedding algorithms. The results are shown in Table 1.

First, the proposed algorithm achieves higher top precision than all state-of-the-art algorithms. This superiority is due to our wise selection of misplaced training images besides the decision boundary. Since our gradient descent direction consists with top precision optimization direction, the test precision enjoys a clear advantage compared to traditional embedding algorithms with inconsistency between training loss and the the concerned evaluation metrics.

Second, our proposed algorithm outperforms all compared algorithms in most metrics besides the top precision, including the Precision vs Recall Curve and the ROC curve in Figure 4. This is interesting since we did not aim to optimize the these metrics. We conjecture that the reason is the correlation between metrics. These results indicate that our proposed algorithm is able to learn embedding of high quality, not only learn the top k nearest neighbors.

Third, when comparing between baseline algorithms, the three winners (after our loss) on the three datasets are distance weighted sampling, angular loss and hard mining. Distance weighted sampling balances the images from various distances and thus avoids noisy hard negative; angular loss introduces scale invariance among different classes; and hard mining samples informative images for effective updates. But no one wins all games. This indicates that each algorithm has its most suitable situation, depending on the noise level, intra-class variance, etc.

4.7. Intuitive Results

We also provide qualitative results for intuitive impression of our embedding, including examples of query re-

	P/R@1	P@3	P@5	P@10	R@3	R@5	R@10	NMI	mAP	F1
CUB-200-2011										
Uniform Triplet	46.49	42.75	40.65	37.20	66.32	74.88	84.44	57.91	21.99	25.78
Hard Mining Triplet	55.30	51.54	49.57	46.02	73.08	80.30	87.66	65.60	24.10	31.15
Semi Hard Triplet	55.91	52.01	49.73	46.01	73.85	80.40	88.00	64.96	23.24	31.04
Distance Weighted	57.56	54.06	51.74	48.23	75.05	81.48	88.31	66.04	31.35	34.39
Contrastive Loss	44.60	41.05	38.59	35.09	64.35	73.13	82.82	56.36	20.13	23.58
Lifted Struct Loss	49.44	45.36	43.35	39.98	67.96	76.18	85.03	58.58	24.63	24.08
N-Pair Loss	53.44	50.29	48.08	43.94	72.89	80.35	87.54	61.87	27.23	28.09
Angular Loss	55.47	51.64	49.44	45.69	73.08	80.17	87.59	65.12	23.83	31.40
Proxy NCA Loss	53.78	49.60	47.18	43.28	72.94	80.65	88.20	62.31	26.99	27.50
Ours ℓ_k	**61.07**	**57.51**	**55.52**	**52.00**	**77.82**	**84.05**	**90.51**	**68.41**	**33.83**	**38.14**
Standford Cars										
Uniform Triplet	55.53	47.29	43.00	36.55	71.93	78.78	86.14	46.04	14.07	15.59
Hard Mining Triplet	69.56	62.44	58.06	50.81	83.03	87.36	91.74	54.55	21.47	22.91
Semi Hard Triplet	75.16	68.95	64.76	57.85	86.15	89.93	93.63	59.71	26.03	28.58
Distance Weighted	66.34	59.13	54.44	47.38	81.64	86.51	92.04	55.80	20.73	22.63
Contrastive Loss	39.16	32.28	28.44	23.55	57.58	65.81	77.22	37.35	8.06	9.09
Lifted Struct Loss	58.89	51.56	46.93	40.19	75.75	81.95	88.72	47.36	15.45	15.98
N-Pair Loss	63.13	59.59	53.88	50.11	79.40	84.99	87.29	50.75	18.39	20.07
Angular Loss	77.44	71.55	67.48	60.87	88.03	91.23	94.53	61.85	28.15	31.11
Proxy NCA Loss	76.35	69.66	65.22	58.47	87.50	90.75	94.29	60.63	27.10	29.30
Ours ℓ_k	**80.03**	**74.58**	**70.83**	**64.33**	**90.01**	**92.53**	**96.25**	**65.24**	**31.71**	**35.49**
Online Product										
Uniform Triplet	62.89	45.36	34.99	21.69	70.70	73.90	77.62	32.80	52.46	29.10
Hard Mining Triplet	74.98	60.25	50.16	31.48	82.21	83.87	87.74	28.78	**61.52**	23.88
Semi Hard Triplet	72.48	57.57	46.95	30.63	80.40	83.34	86.81	32.87	60.21	29.21
Distance Weighted	72.79	56.62	45.43	29.03	80.33	83.07	86.21	32.02	60.57	28.87
Contrastive Loss	58.85	41.33	31.50	19.40	66.42	69.59	73.39	16.64	49.11	17.46
Lifted Struct Loss	65.71	49.68	39.78	25.37	73.03	76.38	81.51	21.93	53.73	22.30
N-Pair Loss	67.53	51.68	41.59	27.03	76.22	79.60	83.44	27.72	54.75	25.24
Angular Loss	73.29	58.22	47.49	31.01	80.95	83.94	87.34	33.97	60.80	28.62
Proxy NCA Loss	67.72	52.03	41.88	27.22	76.57	79.90	83.71	27.51	54.90	24.56
Ours ℓ_k	**78.40**	**63.30**	**52.03**	**33.98**	**85.50**	**88.02**	**90.78**	**35.08**	60.77	**30.29**

Table 1. Comparison with state-of-the-art sampling methods and loss functions, numbers in percentage. The network backbone is densenet201. "P" = Precision and "R" = Recall. Our proposed loss ℓ_k outperforms all compared algorithms. This validates the superiority in wisely selecting images that directly promotes precision. Additional results on Inception is in Supplementary Materials. Note that NMI and F1 in Online Product Dataset are computed by 12 super categories for efficiency .

sults and t-sne (Figure 5 and 6). [6] Intuitively, positive images are usually ranked in the top places. The failure cases are mostly from fine grained different sub-categories in the same major category. This validates the effectiveness of our proposed algorithm.

5. Discussion and Future Directions

Although deep image embedding is an extensively studied topic, this paper spotlights a serious but long overlooked problem, the inconsistency between the gradient descent direction of the training loss and the direction of optimizing the concerned evaluation metric. Specially, no existing deep image embedding algorithms are trained to maximize Prec@k, the metric mostly related to the user experience. In this paper, we present a wise sampling strategy for deep image embedding that selects misplaced images besides the k nearest neighbor boundary, and theoretically prove that the selected images can directly maximize Prec@k.

Our main contribution of "sampling wisely" beside the boundary could also be used to improve loss functions besides Eq. 5 (e.g. angular loss). In addition, we would like to explore precision optimization in ensemble models.

[6] Some results are in the Supplementary Materials due to space limit.

References

[1] Yan Bai, Yihang Lou, Feng Gao, Shiqi Wang, Yuwei Wu, and Ling-Yu Duan. Group-sensitive triplet embedding for vehicle reidentification. *IEEE Transactions on Multimedia*, 20(9):2385–2399, 2018.

[2] Leonard Berrada, Andrew Zisserman, and M Pawan Kumar. Smooth loss functions for deep top-k classification. *arXiv preprint arXiv:1802.07595*, 2018.

[3] Fatih Cakir, Kun He, Xide Xia, Brian Kulis, and Stan Sclaroff. Deep metric learning to rank. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1861–1870, 2019.

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

[5] Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. Online passive-aggressive algorithms. *Journal of Machine Learning Research*, 7(Mar):551–585, 2006.

[6] Jordan Frery, Amaury Habrard, Marc Sebban, Olivier Caelen, and Liyun He-Guelton. Efficient top rank optimization with gradient boosting for supervised anomaly detection. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 20–35. Springer, 2017.

[7] Weifeng Ge. Deep metric learning with hierarchical triplet loss. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 269–285, 2018.

[8] Yanyan Geng, Ru-Ze Liang, Weizhi Li, Jingbin Wang, Gaoyuan Liang, Chenhao Xu, and Jing-Yan Wang. Learning convolutional neural network to maximize pos@ top performance measure. *arXiv preprint arXiv:1609.08417*, 2016.

[9] Xifeng Guo, Long Gao, Xinwang Liu, and Jianping Yin. Improved deep embedded clustering with local structure preservation. In *International Joint Conference on Artificial Intelligence (IJCAI-17)*, pages 1753–1759, 2017.

[10] Chen Huang, Chen Change Loy, and Xiaoou Tang. Local similarity-aware deep feature embedding. In *Advances in Neural Information Processing Systems*, pages 1262–1270, 2016.

[11] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[12] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[13] Purushottam Kar, Harikrishna Narasimhan, and Prateek Jain. Surrogate functions for maximizing precision at the top. In *International Conference on Machine Learning*, pages 189–198, 2015.

[14] 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)*, pages 736–751, 2018.

[15] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, Sydney, Australia, 2013.

[16] Maksim Lapin, Matthias Hein, and Bernt Schiele. Analysis and optimization of loss functions for multiclass, top-k, and multilabel classification. *IEEE transactions on pattern analysis and machine intelligence*, 2017.

[17] Laura Leal-Taixé, Cristian Canton-Ferrer, and Konrad Schindler. Learning by tracking: Siamese cnn for robust target association. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 33–40, 2016.

[18] Jin Li, Xuguang Lan, Xiangwei Li, Jiang Wang, Nanning Zheng, and Ying Wu. Online variable coding length product quantization for fast nearest neighbor search in mobile retrieval. *IEEE Trans. Multimedia*, 19(3):559–570, 2017.

[19] Ru-Ze Liang, Lihui Shi, Haoxiang Wang, Jiandong Meng, Jim Jing-Yan Wang, Qingquan Sun, and Yi Gu. Optimizing top precision performance measure of content-based image retrieval by learning similarity function. In *Pattern Recognition (ICPR), 2016 23rd International Conference on*, pages 2954–2958. IEEE, 2016.

[20] Laurens Van Der Maaten. *Accelerating t-SNE using tree-based algorithms*. JMLR.org, 2014.

[21] Yair Movshovitz-Attias, Alexander Toshev, Thomas K Leung, Sergey Ioffe, and Saurabh Singh. No fuss distance metric learning using proxies. 2017.

[22] Qingqun Ning, Jianke Zhu, Zhiyuan Zhong, Steven CH Hoi, and Chun Chen. Scalable image retrieval by sparse product quantization. *IEEE Transactions on Multimedia*, 19(3):586–597, 2016.

[23] Michael Opitz, Georg Waltner, Horst Possegger, and Horst Bischof. Bier-boosting independent embeddings robustly. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5189–5198, 2017.

[24] Swami Sankaranarayanan, Azadeh Alavi, and Rama Chellappa. Triplet similarity embedding for face verification. *arXiv preprint arXiv:1602.03418*, 2016.

[25] 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*, pages 815–823, 2015.

[26] Edgar Simo-Serra, Eduard Trulls, Luis Ferraz, Iasonas Kokkinos, Pascal Fua, and Francesc Moreno-Noguer. Discriminative learning of deep convolutional feature point descriptors. In *IEEE International Conference on Computer Vision*, 2016.

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

[28] Hyun Oh Song, Stefanie Jegelka, Vivek Rathod, and Kevin Murphy. Deep metric learning via facility location. In *Computer Vision and Pattern Recognition (CVPR)*, volume 8, 2017.

[29] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured fea-

ture embedding. In *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*, pages 4004–4012. IEEE, 2016.

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

[31] Evgeniya Ustinova and Victor Lempitsky. Learning deep embeddings with histogram loss. In *Advances in Neural Information Processing Systems*, pages 4170–4178, 2016.

[32] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. California Institute of Technology, 2011.

[33] Ji Wan, Dayong Wang, Steven Chu Hong Hoi, Pengcheng Wu, Jianke Zhu, Yongdong Zhang, and Jintao Li. Deep learning for content-based image retrieval: A comprehensive study. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 157–166. ACM, 2014.

[34] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Zhifeng Li, Dihong Gong, Jingchao Zhou, and Wei Liu. Cosface: Large margin cosine loss for deep face recognition. *arXiv preprint arXiv:1801.09414*, 2018.

[35] 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. IEEE, 2017.

[36] Xinshao Wang, Yang Hua, Elyor Kodirov, Guosheng Hu, Romain Garnier, and Neil M Robertson. Ranked list loss for deep metric learning. *arXiv preprint arXiv:1903.03238*, 2019.

[37] Jason Weston, Samy Bengio, and Nicolas Usunier. Large scale image annotation: learning to rank with joint word-image embeddings. *Machine learning*, 81(1):21–35, 2010.

[38] Chao-Yuan Wu, R Manmatha, Alexander J Smola, and Philipp Krahenbuhl. Sampling matters in deep embedding learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2840–2848, 2017.

[39] Hong Xuan, Richard Souvenir, and Robert Pless. Deep randomized ensembles for metric learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 723–734, 2018.

[40] Jun Liu Shiqi Wang Ling-Yu Duan Yihang Lou, Yan Bai. Feature distance adversarial network for veicle re-identification. In *Computer Vision and Pattern Recognition, 2019. CVPR 2019.* IEEE, 2019.

[41] Yuhui Yuan, Kuiyuan Yang, and Chao Zhang. Hard-aware deeply cascaded embedding. *CoRR, abs/1611.05720*, 1, 2016.