Exhaustive hard triplet mining loss for Person Re-Identification

Chao XU, Xiang SUN, Ziliang CHEN, Shoubiao TAN
School of Electronics and Information Engineering, AnHui University, HeFei, P.R. China

Received: 15.10.2019 • Accepted/Published Online: 19.06.2020 • Final Version: 25.09.2020

Abstract: Person reidentification (Re-ID) is an important task in computer vision and has many applications in video-based surveillance. Recently, the triplet loss has been popular in the deep learning framework for person Re-ID. It is particularly important to note that the selection of hard triplets has significant influence on the performance of the learned deep model. However, the existing triplet losses only focus on some specific forms of hard triplets, thus leading to weaker generalization capability. To address this issue, we propose a novel variant of the triplet loss, named exhaustive hard triplet mining loss (EHTM), which is able to deal with various forms of hard triplets in a comprehensive manner. Moreover, the proposed loss comprises a term to facilitate distinguishing different identities by directly narrowing intraclass distances and indirectly enlarging interclass distances. We also provide an effective training strategy to further enhance model performance. Extensive experiments on several benchmark datasets show that our method outperforms state-of-the-art approaches by a large margin.

Key words: Person reidentification, triplet loss, hard triplet mining

1. Introduction

Person re-identification (Re-ID) serves as an important part in a video-based surveillance system. Person Re-ID aims to match the person with the same identity across nonoverlapping views in a network of cameras, which has attained a great deal of attention due to its broad applications. The major challenge of person Re-ID arises from large appearance variations caused by pose variation, illumination change, view-point difference, and background clutter. Although great progress has been made addressing this problem, person Re-ID remains an open problem and deserves further studies. In recent years, thanks to the advance of deep learning, a large body of novel methods are proposed to deal with this problem [1–12].

The task of person Re-ID is somewhat similar to face recognition and image retrieval, which aim to find matched ones in a set of images. Consequently, some popular face recognition methods are applied to person Re-ID tasks. One representative work is FaceNet [13], which uses a convolutional neural network (CNN) to extract embedding features and designs the triplet loss [14] (Please refer to Figure 1 for explanation) instead of the softmax loss as the constraint function. Several variants of the triplet loss are also proposed used for solving person Re-ID problems, such as the improved triplet loss [3], the quadruplet loss [2], the batch hard triplet loss (BHTri) [15] and the margin sample mining loss (MSML) [16]. It has been verified that triplet loss is superior to the classification loss in which the number of learnable parameters increases with the number of identities whereas most of the parameters are discarded when conducting query. Consequently, triplet losses have emerged as the mainstream constraint functions in dealing with the problem of person Re-ID.

*Correspondence: tsb@ustc.edu

This work is licensed under a Creative Commons Attribution 4.0 International License.
Figure 1. (a): The input is a triplet, i.e. anchor, positive (denotes different samples of the same identity), and negative (indicates samples from different identities) (b): This indicates all triplets. easy triplets: $\|a - p\|_2 + \text{margin} < \|a - n\|_2$, hard triplets: $\|a - n\|_2 < \|a - p\|_2$, semihard triplets: $\|a - p\|_2 < \|a - n\|_2 < \|a - p\|_2 + \text{margin}$. (c): The ultimate goal of optimization is to shorten the distance between anchor and positive, and lengthen the distance between anchor and negative.

However, the existing family of triplet losses have various limitations, and how to design an effective triplet loss for practical applications remains an open problem. For instance, all variants of the triplet loss require that the intraclass distance is less than the interclass distance, whereas they do not provide a clear measurement for the intraclass and the interclass distances. Although BHTri achieves impressive results on some benchmark datasets, the selection strategy of triplets does not consider some frequently-encountered forms of triplets. MSML conducts forward propagation for one batch and performs backward propagation for only one triplet, consequently rendering the training procedure hard to converge. In the triplet loss, hard triplets are more important than easy ones as they make more contributions to the discriminative power of the learned model. However, most existing methods treat their importance equally, which often makes the neural network converge to suboptimum.

To address the aforementioned issues, we propose a novel triplet loss named exhaustive hard triplets mining loss (EHTM). EHTM can be viewed as the integration of a variant of the triplet loss and the center loss [17]. The variant of the triplet loss takes consideration of the following forms of triplets simultaneously: 1) the positive pair and the negative pair share the same image; 2) the positive pair and the negative pair share the same identity except the case of sharing the same image; and 3) the positive pair and the negative pair have different identities. It is easy to know that the proposed triplet loss covers all the cases of triplets. The center
loss is used as an auxiliary term to make embedding features more compact by directly narrowing intraclass distances and indirectly enlarging interloss distances. To further boost model performance, we introduce a novel fine-tuned strategy named online hard triplets selection (OHTS) in the training stage. OHTS aims to adaptively select hard triplets and discard easy triplets online according to the trained model. Figure 2 provides an overview of the proposed approach.

Figure 2. The overview of the proposed approach. (a) and (b) show the overall frameworks and the distribution of samples in the embedding feature space when using the triplet loss and EHTM, respectively. (c) illustrates the fine-tuned result by using OHTS. Compared with the triplet loss, EHTM is able to produce smaller intraclass distance and larger interclass distance. As shown in (c), OHTS is helpful to further enlarge interclass distances and reduce intraclass distances.

In summary, the main contributions of this paper are three-fold: 1) We propose a novel variant of the triplet loss named exhaustive hard triplets mining loss (EHTM); 2) We propose a fine-tuned strategy in the training stage, which is able to effectively find hard triplets and discard easy triplets; 3) We conduct extensive comparative experiments to demonstrate the advantage of the proposed method over state-of-the-art triplet losses for person Re-ID tasks.
2. Related work

In this section, we review some related typical person Re-ID methods. Most traditional methods of person Re-ID can be divided into two stages, i.e. feature extraction and metric learning. Researchers have put much effort to design discriminative and robust feature representations that are able to describe person under different conditions such as pose, lighting, and viewpoint variations. Typical hand-crafted feature descriptors used for person Re-ID include Gabor features, color histograms, local binary patterns (LBP), and the combinations of several features. Meanwhile, a large body of metric learning approaches have been used in person Re-ID problems such as local Fisher discriminant analysis, large margin nearest neighbor, marginal Fisher analysis, locally adaptive decision functions, and Mahalanobis metric learning. With the prevalence of deep learning techniques in computer vision, an increasing number of deep learning methods are applied to person Re-ID tasks. Compared with traditional methods, deep learning methods are able to yield better results, and they deal with person Re-ID tasks in an end-to-end way by integrating feature representation with distance metric learning together. It is worth noting that loss functions are crucial for the performance of the learned deep models. Typical loss functions used in person Re-ID tasks include classification losses, verification losses, and triplet losses. In the family of classification losses, a softmax layer with the cross-entropy loss is frequently used, and the output of the reciprocal second FC layer is treated as embedding features. The pairwise verification loss aims to determine whether the pairwise identities belong to the same person. A triplet loss necessitates a positive pair and a negative pair when constructing a triplet. The positive pair pulls the same identity closer while the negative one pushes the different identities away. In the earlier studies, researchers adopted an offline approach to select hard triplets by leveraging multiple-branch networks. However, this approach suffers from additional computational cost and fails to select hard triplet due to the fixed network architecture. More recently, Hermans et al. have proposed a cutting-the-edge work that demonstrates the advantage of triplet losses over other losses. Following the work in, other variants of the triplet loss which also show prominent performance were introduced. In this paper, we take an overall consideration of the merit and the shortage of the existing triplet losses, and consequently develop a novel variant of the triplet loss which has a more comprehensive form rather than focusing on some specific forms of triplets.

A number of approaches were applied to selecting hard samples in the training stage for person Re-ID or other related tasks. Ahmed et al. selected hard negative samples offline and fine-tuned their model with those samples. Chen et al. used an adaptive margin threshold to adaptively select hard samples online. Shrivastava et al. selected hard samples and suppressed simple sample online in their region-based CNN detectors. We borrow ideas from the work of and and propose a hard triplet mining strategy named online hard triplet selection (OHTS) for person Re-ID tasks. Our strategy can be viewed as an extension of, whereas it is more suitable for person Re-ID tasks by determining hard or easy triplets based on a margin threshold. The following experiments show that OHTS is an effective strategy to fine-tune our model.

3. The proposed approach

In this section, we first analyze several typical triplet losses, and then we present the proposed method in detail.

3.1. The triplet loss

The work on BHTri provides a rethinking of the role of the triplet loss and verifies that it is a powerful facility to tackle the problem of person Re-ID. BHTri uses each sample in a batch as an anchor to build a triplet...
$I_A, I_P, I_N$. $I_A$ and $I_P$ are the positive pair with the same identity, and $I_A$ and $I_N$ are the negative pair with different identity in a triplet. For each anchor, the most dissimilar positive pair with the same identity and the most similar negative pair with different identities are selected. The embedding features $f_A$, $f_P$, and $f_N$ corresponding to the images $I_A$, $I_P$, $I_N$ are produced via a pretrained deep network. The Euclidean distance is used as the metric to compute the loss. And BHTri is formulated as follows:

$$L_{BHTri} = \frac{1}{M = PK} \sum_{i=1}^{P} \sum_{a=1}^{K} \left( \max_{A,P} \left\| f_A - f_P \right\|_2 - \min_{A,N} \left\| f_A - f_N \right\|_2 + \delta \right) +$$  

where $\delta$ is a margin to distinguish the positive pair with the negative pair, and $M = PK$ (P classes (person identities), and then randomly sampling K images of each class) is the batch size. For a triplet, the first term represents the maximum intraclass distance and the second term denotes the minimum interclass distance. MSML [16] can be viewed as an extension of BHTri, which also tends to select harder triplet for an anchor. Meanwhile, it takes into account the absolute distances between negative pairs. Notably, MSML only picks three or four images $I_A, I_P, I_A', I_N$ in a whole batch to train deep network. And MSML is defined by:

$$L_{MSML} = \frac{1}{M} \sum_{i=1}^{P} \sum_{a=1}^{K} \left( \max_{A,P} \left\| f_A - f_P \right\|_2 - \min_{A',N} \left\| f_{A'} - f_N \right\|_2 + \delta \right)$$  

where $I_A$ and $I_P$ form the hardest positive pair with the same identity, and $I_{A'}$ and $I_N$ constitute the hardest negative pair with different identities in a batch of samples. It is easy to see that the positive pair and the negative pair can possess not only the same identity but also different identities. In contrast, BHTri only considers the case of the same identity. Theoretically, MSML is able to select harder triplet than BHTri. However, MSML wastes lots of triplets and only picks a triplet to calculate the loss for one batch, easily rendering the deep network hard to converge or converge to an undesirable local optimum.

### 3.2. Exhaustive hard triplets mining loss

Compared with previous work, the proposed EHTM loss provides a more comprehensive perspective for constructing the triplet loss. Meanwhile, EHTM includes the measurement of the intraclass distance to make the embedding subspace more discriminative. EHTM is composed of two parts and formulated by:

$$L_{EHTM} = L_{tri} + L_{center}$$  

The first part focuses on the triplet loss. Following the notations in the previous section, $L_{tri}$ is defined by:
where the notations related to $A$ and $P$ denotes different samples of the same identity, and the notations related to $N$ indicate samples from different identities. The soft-margin has been verified to be effective for pulling samples together from the same class \[3\]. Thus, we use the softplus function in $L_{tri}$:

$$g(x) = \ln(1 + \exp(x))$$

We illustrate three cases of triplets using a toy dataset in Figure 3. $L_{tri}$ (4) simultaneously takes consideration of these three cases. In the first component, the positive pair and the negative pair share one common image as demonstrated in Figure 3a. For each image in a batch, it picks the most dissimilar image from the same identity and the most similar image from different identities to form the positive pair and the negative pair, respectively. Consequently the first term constitutes $M$ (batch size) triplets. As shown in Figure 3b, the second component picks one hardest triplet that requires the positive pair and the negative pair sharing one common identity except the same image. The third component describes another case of the hardest triplet, as shown in Figure 3c, where the positive pair and the negative pair do not share the same identity. We select the greater one to join the training procedure from the second and the third terms. As a result, we choose $M+1$ triplets to train our model. The first term guarantees that the optimization of our model is not sensitive to the noise of data. The second and the third terms enable the model to seek the upper bound of positive pairs and negative pairs by adaptively selecting a harder triplet, which is helpful to find better embedding space.

We illustrate three cases of triplets using a toy dataset in Figure 3. $L_{tri}$ (4) simultaneously takes consideration of these three cases. In the first component, the positive pair and the negative pair share one common image as demonstrated in Figure 3a. For each image in a batch, it picks the most dissimilar image from the same identity and the most similar image from different identities to form the positive pair and the negative pair, respectively. Consequently the first term constitutes $M$ (batch size) triplets. As shown in Figure 3b, the second component picks one hardest triplet that requires the positive pair and the negative pair sharing one common identity except the same image. The third component describes another case of the hardest triplet, as shown in Figure 3c, where the positive pair and the negative pair do not share the same identity. We select the greater one to join the training procedure from the second and the third terms. As a result, we choose $M+1$ triplets to train our model. The first term guarantees that the optimization of our model is not sensitive to the noise of data. The second and the third terms enable the model to seek the upper bound of positive pairs and negative pairs by adaptively selecting a harder triplet, which is helpful to find better embedding space.

**Figure 3.** Three cases of triplets

The second part is with the form of the center loss and is denoted by:

$$L_{center} = \frac{1}{2} \lambda \sum_{A=1}^{M} \| f_A - C_A \|_2^2$$

(6)
where $C_A$ represents the $A$-class center of embedding features and $\lambda$ is a tradeoff for balancing the two parts of the proposed loss. The center loss aims to make images with the same identity closer and enlarge the margin of different identities. We use the center loss as an auxiliary term to address the weakness that the triplet loss only measures the relativity between intraclass distances and interclass distances. That is, the center loss provides a supplement for the triplet loss by measuring the absolute distance.

In summary, EHTM has advantage over other triplet losses. Compared with BHTri, EHTM considers all cases of triplets, which enable us to pick harder triplet beneficial for producing better results. MSML only chooses one triplet in a batch to join the training procedure, which suffers from the noise of data and easily makes the model converge to an undesirable local optimum. In contrast, our EHTM uses $M + 1$ triplets in a batch for training, which can resist the noise of data to a large extent. In addition, by using the center loss as an auxiliary constraint, EHTM is able to find more compact embedding space for the same identity, thus enhancing the discriminative power of the learned model.

3.3. Online hard triplet selection

For a triplet, if the intraclass distance is relatively large and the interclass distance is relatively small, we can group it into hard triplets. Otherwise, it can be grouped into easy triplets. It is a common sense that the number of easy triplets is much more than that of hard triplets with the epoch increasing. Although the loss value of each easy triplet is small, the easy triplets have a large proportion. A mass of easy triplets may stand as an obstacle for seeking fewer hard triplets in the fine-tuning stage and impose negative impact on the final optimization result, if we do not adopt an effective hard triplets mining strategy. Through our empirical study, we find that a two-stage training strategy is very useful to boost model performance. That is, we can train a model to converge in the former stage and then use hard triplets to fine-tune the obtained model in the latter stage. We hereby propose a fine-tuned method called online hard triplet selection (OHTS) to pick hard triplets to join training.

The main idea of OHTS can be summarized as follows: 1) It uses EHTM as the loss function to train the network until convergence, 2) sets a proper margin threshold $\delta$ and conducts forward propagation for a batch of samples, 3) forms triplets and calculates triplet losses, 4) divides triplets into easy and hard ones online according to the threshold, 5) performs backward propagation using hard triplets and discards easy triplets. The proposed strategy is detailed in Algorithm 1.

By applying OHTS, we can focus on hard triplets and discard easy triplets to fine-tune the deep neural network, thereby producing smaller intraclass variation and larger interclass variation. Although this approach is simple, it is effective and makes a good complement for the proposed EHTM.

4. Experiments

We conduct two sets of comprehensive experiments to evaluate the performance of the proposed method. The first is to compare EHTM with other losses including the classification loss and other variations of the triplet loss. The second mainly concentrates on the performance when OTHS and two-stream networks are used.

4.1. Datasets and evaluation metrics

In this paper, we conduct experiments on three popular datasets including Market-1501 [27], MARS [28], and DukeMTMC-reID [29]. Market-1501, one of the frequently used person Re-ID datasets, was collected by one low-resolution camera and 5 high-resolution cameras at Tsinghua University. It contains 32,668 images with
Algorithm 1 Online hard triplet selection

**Input:** a training set, a convergent CNN model

**Output:** a fine-tuned CNN model

1: set a margin threshold $\delta$
2: for $t$ in 1 to $T$ do
3: conduct forward propagation for a batch of samples
4: construct triplets and calculate triplet losses
5: for each loss in triplet losses do
6: if the loss is greater than $\delta$ then
7: the loss is grouped into hard triplet losses
8: else
9: the loss is grouped into easy triplet losses
10: end if
11: end for
12: set easy triplet losses to 0
13: conduct backward propagation for hard triplet losses
14: end for

1501 identities. Each identity was captured by at least two cameras. All images are detected using the DPM detector and hence contain 2793 false detections to simulate the real-world scenario. We conduct single-query (SQ) and multiquery (MQ) evaluation defined by [27] on this dataset. Following [27], we use 12,936 images with 751 identities for training, 19,732 images with 750 identities as gallery, and 3368 images with 750 identities as query. MARS is a large-scale video-based person Re-ID dataset, which is built from the same raw data as the Market-1501 dataset. It consists of 1,191,003 images sampled from 20,478 tracklets, which contains 1261 identities captured by 6 cameras. The tracklets are produced by the DPM detector and the GMMCP tracker. Person Re-ID is conducted on a tracklet-to-tracklet level instead of image-to-image level on this dataset. As for testing, embedding features are pooled across a tracklet so that it is essentially a multiquery setting. Following [28], we use 509,914 images with 625 identities for training and 681,089 images with 631 identities for testing. DukeMTMC-reID is a subset of the DukeMTMC for image-based person Re-ID. It comprises 36,411 images with 1404 identities captured by 8 cameras. Following [29], we use 16,522 images with 702 identities for training, 17,661 images with 702 identities as the gallery, and 2228 images with 702 identities as the query. We employ the mean average precision (mAP) and the cumulative matching characteristic (CMC) curve as the metrics to evaluate algorithm performance in the Euclidean space. The open-source evaluation code provided by [15] is used to compute these metrics.

### 4.2. Network architecture

We implement our model using the Tensorflow [30] framework. The ResNet-50 architecture pretrained on ILSVRC-2012-CLS is used as the backbone network. We use the convolution layers from ResNet-50 and add two FC layers. The first FC layer with 1024 units is followed by batch normalization and ReLU. The output of the second FC layer with 128 units is used as embedding features.

### 4.3. Implementation details

Data preprocessing is important to achieve good performance and enhance generalization capability. In our experiment, all images are resized to $128 \times 256 (W \times H)$ pixels. The employed operations of data preprocessing include subtracting mean value, random horizontal flipping, random cropping, and random erasing [31]. Specif-
ically, we resize all images to $144 \times 288(W \times H)$ pixels and then randomly crop $125 \times 256(W \times H)$ regions as the input when using random cropping. Random erasing is a recently proposed method which randomly chooses a rectangle region of the image and erases this region with random pixels. It reduces the risk of overfitting and is complementary to random cropping and flipping.

Adam optimizer $[32]$ is used for training model. The initial learning rate is set to $3 \times 10^{-4}$ in the first 15,000 epoches and decreased to $3 \times 10^{-7}$ in the next 10,000 epoches. In each epoch, we select 18 identities and randomly select 4 images for each identity. $\lambda$ is set to $1 \times 10^{-5}$ in order to balance the center loss and the triplet loss.

4.4. Performance evaluation

We first evaluate the performance of EHTM by comparing it to some typical losses. To make a fair comparison, we only conduct subtracting mean value and do not adopt any other data preprocessing operations in the training stage. In the classification loss, the unit number of the second FC layer is set to the number of total identities in the training dataset and we use the output of the first FC layer as embedding features. We report the mAP, the rank-1, and the rank-5 values of all the compared methods in Table 1.

Table 1 shows consistent results reported in a recent paper $[15, 16]$ that the triplet loss achieves better performance than the classification loss. Comparing BHTri with the classification loss, we can observe that the mAP rises from 52.22% to 66.59% and the rank-1 rises from 77.91% to 83.05% on the Market-1501 dataset. MSML surpasses BHTri in most cases, which demonstrates that MSML is able to select harder triplets than BHTri. EHTM outperforms MSML by a large margin on most test cases, and some results are slightly worse than the best. Overall, our loss achieves the best results on all the test datasets, demonstrating the advantage of EHTM over the alternatives. We then turn to verify the benefits of OHTS and the architecture of two-stream network.


<table>
<thead>
<tr>
<th>Losses</th>
<th>Market1501</th>
<th>Mars</th>
<th>DukeMTMC-reID</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>mAP</td>
<td>rank-1</td>
<td>rank-5</td>
</tr>
<tr>
<td>Cls</td>
<td>52.22</td>
<td>77.91</td>
<td>90.57</td>
</tr>
<tr>
<td>BHTri</td>
<td>66.59</td>
<td>83.05</td>
<td>93.32</td>
</tr>
<tr>
<td>MSML</td>
<td>67.21</td>
<td>83.11</td>
<td>93.76</td>
</tr>
<tr>
<td>EHTM</td>
<td><strong>69.62</strong></td>
<td><strong>84.92</strong></td>
<td><strong>93.91</strong></td>
</tr>
</tbody>
</table>

Without specification, we use the data preprocessing operations mentioned above for training in this part of experiments. During the test procedure, we adopt the test-time augmentation. For a probe image, we first use horizontal flipping and then crop five parts, i.e. upper-left, lower-left, upper-right, lower-right, center, thereby producing ten images from an image. We extract embedding features of these ten images and use average pooling to yield embedding features for the probe image. These operations are beneficial to improve accuracy. In addition, we find that the cropping operation produces negative impact on accuracy on the MARS dataset through our empirical study. Consequently, we do not conduct the operation of cropping for training and testing on the MARS dataset.

To validate the OHTS strategy, we fine-tune the deep network on the basis of trained model using EHTM. Learning rate is set to $5 \times 10^{-6}$, and the margin $\delta$ is set to $1 \times 10^{-3}$ to determine hard or easy triplets.
Learning model takes 20,000 epoches. As shown in Table 2, EHTM with OHTS produces remarkable results on the Market-1501, the MARS, and the DukeMTMC-reID datasets, which demonstrates that EHTM with OHTS is capable of dealing with various person Re-ID tasks. Taking EHTM as the baseline, the application of OHTS leads to a consistent increment whatever on the mAP or the ranking values. The mAP values increase by +0.89%, +0.83%, +0.53% and the rank-1 values increase by +0.83%, +0.51%, +0.53% on the Market-1501 SQ, the MARS and the DukeMTMC-reID datasets, respectively. Although the baseline is considerably high, OTHS produces a meaningful margin of improvement over EHTM, demonstrating its effectiveness in boosting model performance.

The application of two-stream deep network builds upon the fact that, even though we train the same deep network under the same settings, the obtained networks have slightly difference as some operations are performed in a random way. The two-stream network with nonshared parameters has the same architecture in each branch and shares the data layer in common. Each subnet is a backbone network and trained independently. We concatenate the output of subnets for testing and consequently embedding features have 256 dimensions. Zhang et al. [37] experimentally verified that such an idea is useful to enhance model performance. Our experimental results on the application of two-stream network are reported in Table 2. We can observe that such an architecture plays a positive role in boosting model performance. We also combine EHTM, OHTS, and two-stream deep network together, and experimental results show that these three ingredients are complementary to yield better results.

Furthermore, we combine our method with the reranking approach [35]. The reranking approach reorders the images in a gallery to improve the query accuracy. Experimental results in Table 2 show that using this approach can yield significant improvement. In addition, we compare our methods with state-of-the-art methods. As shown in Table 2, the proposed method outperforms state-of-the-art approaches by a large margin, which further verifies the effectiveness of the proposed method. Importantly, our method shows its advantage on the rank-1 and the rank-5 results, which demonstrates the capability of our method in enlarging interclass variations and reducing intraclass variations. At the same time, we provide a visual search sorting result on market-1501, which shows that our algorithm can further expand the distance between classes and reduce the distance within classes. In Figure 4, the green box indicates the samples matched by the query, while the red box indicates the samples not matched by the query. The value below each image represents the European distance between the query image and the sample image. In Figure 4a, by comparing the Euclidean distance of BHtr and EHTM loss function query sorting results, we can see that EHTM can get smaller intraclass distance. In Figure 4b, by comparing the query sorting results of EHTM and OHTS, it can be seen that by using OHTS training strategy, the Euclidean distance between different categories is obviously pushed further, even not within the top 5 sorting results. This shows that OHTS training strategy can produce a larger distance between classes.

Finally, we compared the basic networks of different pretrained architectures namely Resnet50, SENET, ResNeXt-50, and achieved certain results in Table 3. With the continuous optimization of the structure of the pretrained architectures, we found that its effect will be improved to a certain extent, especially in the case of complex data sets. This shows that with the improvement of network performance, our ETHM + OHTS function will have a better performance. In addition, we also tried to experiment in MobileNet and achieved some results.
Table 2. Comparison of different methods on three datasets. We report the results of single-query (SQ) and multiquery (MQ) on the Market-1501 datasets. TSN denotes the two stream network with nonshared parameters. The top part reports the results of state-of-the-art approaches, where the best results are highlighted in bold. The middle part reports the result of the proposed method, where the best results are underlined. The bottom part reports the results of the proposed method with reranking.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Market1501 SQ</th>
<th>Market1501 MQ</th>
<th>MARS</th>
<th>DukeMTMC-reID</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>mAP</td>
<td>rank-1</td>
<td>rank-5</td>
<td>mAP</td>
</tr>
<tr>
<td>BoW+KISSME(2015) [27]</td>
<td>20.76</td>
<td>44.42</td>
<td>63.90</td>
<td>-</td>
</tr>
<tr>
<td>LSRO(ResNet)(2017) [11]</td>
<td>56.23</td>
<td>78.06</td>
<td>-</td>
<td>68.52</td>
</tr>
<tr>
<td>APR(ResNet)(2017) [33]</td>
<td>64.67</td>
<td>84.29</td>
<td>93.20</td>
<td>-</td>
</tr>
<tr>
<td>TAM+SRM(2017) [34]</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>IDE(R)+XQDA+Rerank(2017) [35]</td>
<td>61.87</td>
<td>75.14</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>SVDNET(2017) [36]</td>
<td>62.10</td>
<td>82.30</td>
<td>92.30</td>
<td>-</td>
</tr>
<tr>
<td>TriNet(2017) [15]</td>
<td>69.14</td>
<td>84.92</td>
<td>94.21</td>
<td>76.42</td>
</tr>
<tr>
<td>MobileNet+DML(2017) [37]</td>
<td>68.83</td>
<td>87.73</td>
<td>-</td>
<td>77.14</td>
</tr>
<tr>
<td>SVDNET(ResNet)(2017) [16]</td>
<td><strong>69.60</strong></td>
<td>85.20</td>
<td>93.7</td>
<td>-</td>
</tr>
<tr>
<td>EHTM</td>
<td>74.94</td>
<td>88.21</td>
<td>95.93</td>
<td>81.52</td>
</tr>
<tr>
<td>EHTM+OHTS</td>
<td>75.83</td>
<td>89.04</td>
<td>95.78</td>
<td>81.91</td>
</tr>
<tr>
<td>EHTM+TSN</td>
<td>77.89</td>
<td>89.61</td>
<td>96.44</td>
<td>83.66</td>
</tr>
<tr>
<td>EHTM+OHTS+TSN</td>
<td><strong>78.31</strong></td>
<td>90.26</td>
<td>96.41</td>
<td>84.04</td>
</tr>
<tr>
<td>EHTM+Rerank</td>
<td>85.83</td>
<td>89.61</td>
<td>94.57</td>
<td>90.48</td>
</tr>
<tr>
<td>EHTM+OHTS+TSN+Rerank</td>
<td>87.96</td>
<td>91.86</td>
<td>95.64</td>
<td>91.65</td>
</tr>
</tbody>
</table>
Figure 4. Illustration of the effects of EHTM and OHTS.
<table>
<thead>
<tr>
<th>Base model</th>
<th>Methods</th>
<th>Market1501 SQ</th>
<th>Market1501 MQ</th>
<th>MARS</th>
<th>DukeMTMC-reID</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>mAP</td>
<td>rank-1</td>
<td>rank-5</td>
<td>mAP</td>
<td>rank-1</td>
</tr>
<tr>
<td>ResNet50</td>
<td>TriNet</td>
<td>74.94</td>
<td>96.29</td>
<td>67.70</td>
<td>73.35</td>
</tr>
<tr>
<td></td>
<td>EHTM</td>
<td>74.94</td>
<td>96.29</td>
<td>67.70</td>
<td>73.35</td>
</tr>
<tr>
<td></td>
<td>EHTM+OHTS</td>
<td>75.83</td>
<td>96.29</td>
<td>67.70</td>
<td>73.35</td>
</tr>
<tr>
<td>SENET</td>
<td>TriNet</td>
<td>70.55</td>
<td>96.29</td>
<td>67.70</td>
<td>73.35</td>
</tr>
<tr>
<td></td>
<td>EHTM</td>
<td>75.27</td>
<td>96.29</td>
<td>67.70</td>
<td>73.35</td>
</tr>
<tr>
<td></td>
<td>EHTM+OHTS</td>
<td>77.34</td>
<td>96.29</td>
<td>67.70</td>
<td>73.35</td>
</tr>
<tr>
<td>ResNeXt-50(32x4d)</td>
<td>TriNet</td>
<td>71.25</td>
<td>96.29</td>
<td>67.70</td>
<td>73.35</td>
</tr>
<tr>
<td></td>
<td>EHTM</td>
<td>76.59</td>
<td>96.29</td>
<td>67.70</td>
<td>73.35</td>
</tr>
<tr>
<td></td>
<td>EHTM+OHTS</td>
<td>77.25</td>
<td>96.29</td>
<td>67.70</td>
<td>73.35</td>
</tr>
<tr>
<td>MobileNet</td>
<td>TriNet</td>
<td>66.25</td>
<td>96.29</td>
<td>67.70</td>
<td>73.35</td>
</tr>
<tr>
<td></td>
<td>EHTM</td>
<td>68.32</td>
<td>96.29</td>
<td>67.70</td>
<td>73.35</td>
</tr>
<tr>
<td></td>
<td>EHTM+OHTS</td>
<td>69.11</td>
<td>96.29</td>
<td>67.70</td>
<td>73.35</td>
</tr>
</tbody>
</table>
5. Conclusion
In this paper, we have proposed a novel variant of the triplet loss called exhaustive hard triplet mining loss (EHTM), which focuses on overcoming the weakness of the existing triplet losses from a more comprehensive viewpoint. In EHTM, we also stress the integration of the center loss to make the learned embedding features more discriminative. Moreover, we have proposed a fine-tuning strategy called online hard triplets selection (OHTS) to further refine the learned model. Experimental results show that our method outperforms state-of-the-art algorithms by a large margin on some benchmark datasets.

Acknowledgments
This work was supported in part by the National Natural Science Foundation of China under Grant 61772032, and in part by the Special Fund for Key Program of Science and Technology of Anhui Province, China, under Grant 17030901015.

References


