Learning Incremental Triplet Margin for Person Re-Identification

Yingying Zhang, Qiaoyong Zhong, Liang Ma, Di Xie, Shiliang Pu
Hikvision Research Institute
{zhangyingying7,zhongqiaoyong,maliang6,xiedi,pushiliang}@hikvision.com

Abstract

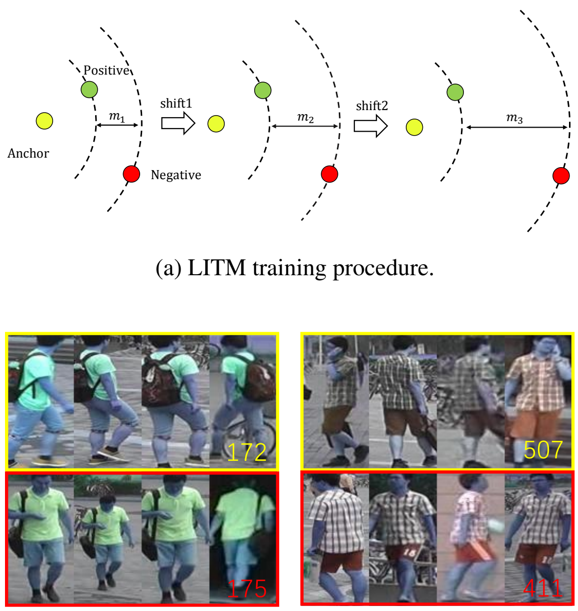
Person re-identification (ReID) aims to match people across multiple non-overlapping video cameras deployed at different locations. To address this challenging problem, many metric learning approaches have been proposed, among which triplet loss is one of the state-of-the-arts. In this work, we explore the margin between positive and negative pairs of triplets and prove that large margin is beneficial. In particular, we propose a novel multi-stage training strategy which learns incremental triplet margin and improves triplet loss effectively. Multiple levels of feature maps are exploited to make the learned features more discriminative. Besides, we introduce global hard identity searching method to sample hard identities when generating a training batch. Extensive experiments on Market-1501, CUHK03, and DukeMTMC-reID show that our approach yields a performance boost and outperforms most existing state-of-the-art methods.

Introduction

In recent years, person re-identification (ReID) has aroused concerns of more and more researchers due to its wide range of applications in security and video surveillance. It is a challenging task because of varying illumination conditions, human occlusion, background clutter and different camera views. Most existing methods use a feature vector to represent each person image and then match them with a specific metric. With the emergence of deep learning, feature representations learned with convolutional neural networks (CNN) (Krizhevsky, Sutskever, and Hinton 2012; LeCun et al. 1989) immensely outperform hand-crafted features.

Currently, the most commonly used loss functions are triplet loss, classification loss and verification loss. Triplet loss was first introduced by (Weinberger and Saul 2009). It directly optimizes a deep CNN which produces embeddings such that positive examples are closer to an anchor example than negative examples. For classification loss, each identity of person in the training set is considered as a class, and the network is trained to classify them correctly. Subsequently, the trained network is used as a feature extractor and a specific metric is chosen to rank the extracted features. In general, the performance of classification loss is superior over triplet loss, since it enforces global inter-class separability in the embedding space. However, as the number of identities increases, the number of learnable parameters grows. For scenarios with very large quantity of identities, it would be non-trivial to train a classification loss. Lastly, verification loss is used to learn a cross-image representation. The network predicts the similarity between two input images directly. During inference, all query-gallery image pairs need to go through the whole network, which is very expensive.

Triplet loss attempts to enforce a margin between the positive and negative pairs of each triplet. Surprisingly, the im-



(a) LITM training procedure.

(b) Hard identity examples (yellow: anchor, red: negative).

Figure 1: (a) illustrates the proposed LITM training procedure. By repeatedly applying a shift on the data points in the embedding space, the margin between the positive and negative pairs in the triplet is progressively increased. (b) shows two examples of hard identity pair on the Market-1501 dataset. The number on the bottom right is the identity label of each person.

pact of the margin on ReID performance has not been explored yet in the literature. Intuitively, larger margin leads to better performance. However, as shown in our experiment (Table 9), simply increasing the margin value of triplet loss does not work well. Instead, we propose a novel training strategy named Learning Incremental Triplet Margin (LITM). As shown in Figure 1(a), we learn a large margin in an multi-stage manner. Firstly, decent positions of the triplet examples in the embedding space are learned using triplet loss with a small base margin. Next, a shift of each data point in the embedding space is learned, which enlarges the gap between the positive and negative pairs. This step is repeatedly applied so that the margin gets increased in an incremental manner. Since mid-level features of the network contain more detailed information and are thus helpful to differentiate identities with similar appearance, we learn the feature shifts using multiple mid-level features. It is worth noting that all the components are implemented in the same network and optimized end-to-end. With LITM, the performance of triplet loss gets significantly improved.

Training of triplet loss requires sampling of triplets from all training images. The number of possible triplets grows cubically with the number of images in the training set. To train triplet loss efficiently, (Hermans, Beyer, and Leibe 2017) proposed batch hard triplet mining. Firstly, a batch of images is generated by randomly sampling P identities and K images per-identity. Then, for each sample in the batch, the hardest positive and negative samples within the batch are selected to form the triplets. It solves the impractical long training problem partially. However, sampling identities randomly may not ensure that negative pairs are hard enough. For instance, two persons with similar appearance may be dispersed to different batches so that there is no hard negative pair in one batch. To address this issue, we introduce a new identity sampling method called Global Hard Identity Searching (GHIS). We compute the pairwise mean embedding distances of all identities, which measure their dissimilarities. Then identities with small distance (similar appearance) are put together to form a batch. Figure 1b shows two examples of searched hard identities from the Market-1501 dataset. We can see that different persons may wear clothes with similar color or texture, which makes them difficult to distinguish even for human beings.

When designing the network architecture for person ReID, it is currently a best practice to adapt from a pre-trained network, e.g. ImageNet pre-trained ResNet-50 (He et al. 2016). However, vanilla ResNets were designed for the task of coarse-grained image classification. While person ReID requires a fine-grained recognition within the person category. To narrow the gap, we revisit the ReID problem carefully and propose some guidelines on its network design. Following the guidelines, we make some tweaks to the feature extractor network, which yields a strong baseline implementation of triplet loss.

In summary, the main contributions of this paper are as follows:

- We propose a novel training strategy which learns incremental triplet margin and leads to significant performance improvement.
- We introduce global hard identity searching method which samples hard identities and makes the training more efficient and effective.
- We propose some guidelines on network design for the task of person ReID and yield a strong triplet loss baseline.
- Combining all the improvements, we achieve state-of-the-art performances on common person ReID benchmarks.

Related Work

Person ReID For the task of person ReID, most existing approaches attempt to learn identity-discriminative representation of person images with supervised learning. With the recent advancements of deep learning, this field has been dominated by deep neural networks. (Xiao et al. 2016) used a classification loss to learn deep feature representations from multiple domains. (Qian et al. 2017) proposed a novel multi-scale deep learning model that is able to learn deep discriminative feature representations at different scales. Their method can automatically determine the most suitable scales for matching. (Shen et al. 2018b) proposed a Kronecker Product Matching module to generate matching confidence maps between two pedestrian images. (Guo and Cheung 2018) proposed a fully convolutional Siamese network to improve the measurement of similarity between two input images. Rather than feature learning from the whole person image, some other works exploit part-based features. (Yao et al. 2017) clustered the coordinates of maximal activations on feature maps to locate several regions of interest. (Zhao et al. 2017) embedded the attention mechanism in the network, allowing the model to decide where to focus by itself. In addition, some works attempt to incorporate extra information like human pose and appearance mask to facilitate person ReID. (Zheng et al. 2015) proposed to extract separate features of different body regions and merge them using a tree-structured fusion network. (Su et al. 2017) proposed a pose-driven deep CNN model which explicitly leverages the human part cues to learn effective feature representations.

Triplet Loss Strictly speaking, triplet loss was first introduced by (Weinberger and Saul 2009). They trained the metric with the goal that the k-nearest neighbors belong to the same class and examples of different classes can be dissociated by a large margin. Based on this work, (Schroff, Kalenichenko, and Philbin 2015) improved the loss to learn a unified embedding for face recognition. They pushed forward the concept of triplet and minimized the distance between an anchor and a positive while maximized the distance between the anchor and a negative. (Cheng et al. 2016) improved the triplet loss function by restricting positive pairs within a small distance. And this improved loss was used to train a multi-channel parts-based convolutional neural network model. Recently, (Hermans, Beyer, and Leibe 2017) summarized the works of ReID using triplet loss, and proposed some training strategies to improve the performance of triplet loss. While our work is also based on triplet loss, we investigate the influence of the margin, which has received little attention so far.

Hard Example Mining Hard example mining has been widely exploited to assist training of deep neural networks. (Shrivastava, Gupta, and Girshick 2016) proposed online hard example mining to improve the performance of object detection. (Hermans, Beyer, and Leibe 2017) extended this idea and selected the hardest positive and negative samples within a batch when generating triplets. These methods can be categorized as *local* hard example mining considering that hard examples are mined from a training batch instead of the whole training set. While the proposed GHIS searches hard identities *globally* from all identities in the training set.

Method

Since our approach is based on triplet loss, let us first briefly recap its formulation. Training of triplet loss requires carefully designed sampling of triplets. A triplet consists of an anchor image x_i^a, a positive image x_i^p of the same person as the anchor and a negative image x_i^n of a different person. Triplet loss aims to learn a feature embedding so that x_i^a is closer to x_i^p than it is to x_i^n in the embedding space. It can be formulated as follows:

$$\begin{aligned} \mathcal{L}_0 &= \sum_{i}^{N} \left[d_0^{ap} - d_0^{an} + m_0 \right]_+, \\ d_0^{ap} &= ||f_0(x_i^a) - f_0(x_i^p)||_2^2 \\ d_0^{an} &= ||f_0(x_i^a) - f_0(x_i^n)||_2^2 \end{aligned} \tag{1}$$

where $[\cdot]_+$ is the hinge function. d_0^{ap} and d_0^{an} are the squared Euclidean distance between the anchor-positive and anchor-negative pairs respectively. m_0 is the margin enforced between d_0^{an} and d_0^{ap}. N is the number of triplets in a training batch. $f_0(x_i) \in \mathbb{R}^d$ denotes the d-dimensional feature embedding of x_i. Here we apply a subscript 0 on f, d and m to indicate the base feature vector, distance and margin respectively. They will be updated later.

Learning Incremental Triplet Margin

To learn large margin between positive and negative pairs, we propose a novel multi-stage training strategy. Firstly, we train the aforementioned triplet loss $\mathcal{L}_0$ with a small base margin m_0 and obtain a base feature embedding $f_0(x_i) \in \mathbb{R}^d$. Meanwhile, we learn a feature shift vector $f_1^s(x_i)$ which is of the same dimension as $f_0(x_i)$. By adding the two vectors together, we get a shifted feature embedding $f_1(x_i)$. This process can be recursively applied by:

$$f_j(x_i) = f_{j-1}(x_i) + f_j^s(x_i), \quad \forall j \geq 1 \tag{2}$$

The feature shifts $f_j^s(x_i)$ are not learned directly. Instead, we supervise the shifted features $f_j(x_i)$ with another triplet loss $\mathcal{L}_j$. To make sure that each time of feature shifting results in better feature embedding, the margin of $\mathcal{L}_j$ is monotonically increased by $m_j = m_{j-1} + \Delta m_j$. The j-th triplet loss is defined as:

$$\begin{aligned} \mathcal{L}_j &= \sum_{i}^{N} \left[d_j^{ap} - d_j^{an} + m_j \right]_+, \\ d_j^{ap} &= ||f_j(x_i^a) - f_j(x_i^p)||_2^2 \\ d_j^{an} &= ||f_j(x_i^a) - f_j(x_i^n)||_2^2 \end{aligned} \tag{3}$$

Algorithm 1 Global Hard Identity Searching.

```
Require: Training set of $n$ identities, feature extractor
Ensure: Hard identity sets of all $n$ identities $S$
Compute the mean distance matrix $\bar{D}$ with Equation (5)
Set diagonal elements $\{D_{u,u}\}$ in $\bar{D}$ to infinity
for each identity $u = 1, \dots, n$ do
    Find the $g$ most similar candidate identities $C_u$ according to $\bar{D}_{u,*}$
    Generate hard identity set $S_u$ by randomly sampling $q$ identities from $C_u$
end for
return $S = \{S_1, \dots, S_n\}$
```

The incremental design makes the learning of large margin easier. The gap between distance of negative pairs d^{an} and that of positive pairs d^{ap} gets enlarged progressively. And we empirically demonstrate that larger margin trained by this way leads to better performance.

All of the triplet losses at different stages are optimized jointly. The final loss is the weighted sum of all losses:

$$\mathcal{L} = \sum_{j=0}^{M} \lambda_j \mathcal{L}_j \tag{4}$$

where M is the number of times that feature shifting is applied. λ_j is the weight used to balance different losses. In our experiments, we set M to 2 and λ_j to 1 in all stages.

Exploiting Multiple Levels of Features

High-level feature maps of a neural network contain coarse semantic information, while mid-level features contain detailed structure information. In previous works, only high-level features have been exploited. We argue that mid-level features are important for fine-grained visual recognition tasks like person ReID. Our multi-stage framework easily enjoys the benefits of different levels of features. Specifically, we learn the base feature embedding $f_0(\cdot)$ using high-level features, which serves as a decent starting point. Then mid-level features are exploited to learn the feature shifts $f_j^s(\cdot)$, which requires a closer look at subtle appearance differences between two persons. See Figure 2 for the pipeline of our framework.

Global Hard Identity Searching

To produce triplets with high quality negative pairs, we introduce a global hard identity searching method. When generating a training batch, (Hermans, Beyer, and Leibe 2017) sample P identities randomly. Instead we define a metric to measure the dissimilarity between two identities and put similar identities together in the same batch. Given a training set with n identities, we randomly sample K examples per-identity. Then we compute the pairwise mean distance matrix $\bar{D}$ of the n identities. $\bar{D}$ is an $n \times n$ symmetric matrix. $\bar{D}_{u,v}$ measures the dissimilarity between identity u and v, which is defined as:

$$\bar{D}_{u,v} = \frac{1}{K^2} \sum_{l=1}^{K} \sum_{r=1}^{K} ||f_M(x_l^u) - f_M(x_r^v)||_2^2 \tag{5}$$

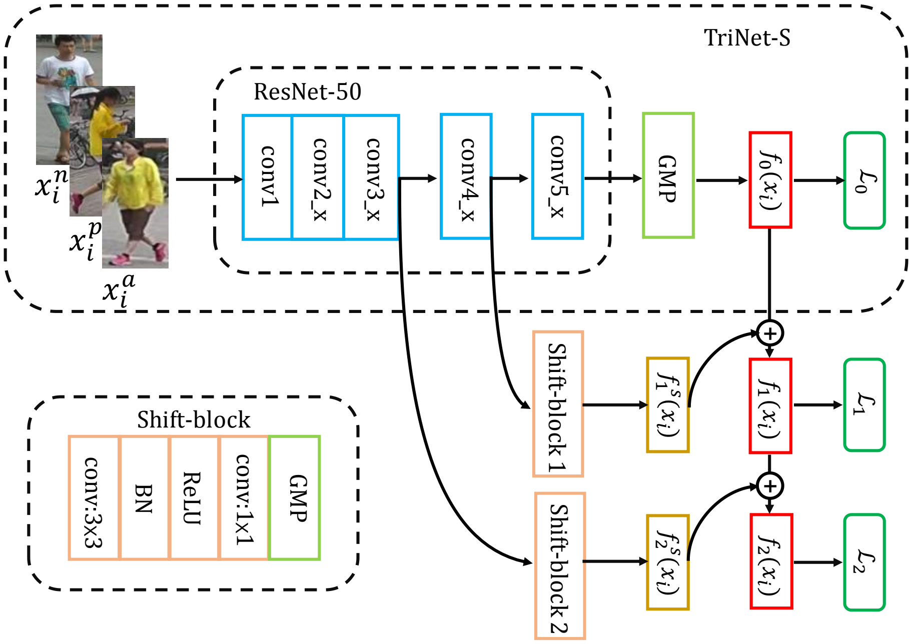


Figure 2: Pipeline of the proposed LITM approach. Upper: a strong triplet loss baseline network (TriNet-S). Lower: learning feature shift vectors for recursively shifted features.

The searching procedure is described in Algorithm 1. After computing the mean distance matrix $\bar{D}$, we set its diagonal elements $\{\bar{D}_{u,u}\}$ to infinity to prevent each identity itself from being sampled as its hard identity. Then for each identity u, we find g identities of the smallest distances to u as its candidate hard identities. To introduce more randomness, q ($q < g$) identities are sampled from the g candidates as the final hard identity set. Each identity and its q hard identities form an identity group which contains $q+1$ identities. When generating a training batch, we sample $P/(q+1)$ identity groups, which results in P identities in total. After a training batch is generated, we use batch hard triplet mining (Hermans, Beyer, and Leibe 2017) to sample hard triplets within the batch.

In our experiments, we notice that the hard identity set of a identity rarely changes during training. To cover more identity permutations and make the training more stable, we apply GHIS and random identity sampling in an alternating way. We first train the network with random identity sampling for two epochs, followed by GHIS for one epoch. This procedure is repeatedly applied. As for other hyper-parameters, we set $g = 5$, $q = 3$, $P = 20$ and $K = 4$.

TriNet-S: A Strong Triplet Loss Baseline Network

For the network architecture, we consider TriNet proposed in (Hermans, Beyer, and Leibe 2017) as a reference. TriNet is adapted from ResNet-50 (He et al. 2016), where the last fully connected layer is replaced with two new fully connected layers. The first layer reduces the feature dimension from 2048 to 1024. And the second layer further reduces the dimension to 128, which serves as final feature embedding. We argue that this configuration is not optimal. For person ReID which involves fine-grained recognition, we propose the following guidelines on network design.

- A fully convolutional architecture is preferable to learn spatial-aware features.
- Global maximum pooling results in sharper and thus more discriminative responses than global average pooling.
- Resolution matters. Large feature map size is preferable as more detailed information is preserved.

Following these guidelines, we make some tweaks to TriNet. Firstly, we remove the last two fully connected layers and use the globally pooled features as final feature embedding. Secondly, we replace the global average pooling (GAP) of ResNet-50 with global maximum pooling (GMP). Thirdly, we reduce the stride of the first convolutional layer in the conv5_x block from 2 to 1, which doubles the feature map size. With these tweaks alone, we achieve significant performance gain. We refer to our implementation as TriNet-S, which serves as a strong triplet loss baseline.

Network Architecture

Combining TriNet-S and LITM, our network architecture is shown in Figure 2. Following (Hermans, Beyer, and Leibe 2017), we use ResNet-50 as our backbone network. The 2048-dimensional output of GMP is utilized as the base feature embedding $f_0(\cdot)$. The feature maps of conv4_x and conv3_x are fed into two shift blocks respectively, producing two feature shift vectors: $f_1^s(\cdot)$ and $f_2^s(\cdot)$. Then the shifted features are created by adding the shift vector to the base feature vector. The shift block is a tiny sub-network as shown in

Figure 2. The number of channels of the first 3×3 convolution is kept the same as its input channels (i.e. 1024 for shift-block1 and 512 for shift-block2). The second 1×1 convolution is utilized to increase the feature dimension to 2048. Notably, feature maps in shift-block2 are down-sampled by half by setting stride of the first convolution to 2. During training, all the three triplet losses are optimized jointly. During inference, only the final shifted feature embedding $f_2(\cdot)$ is used.

Experiments

Datasets

We evaluate the proposed approach on three large-scale person ReID datasets, namely Market-1501 (Zheng et al. 2015), CUHK03 (Li et al. 2014) and DukeMTMC-reID (Ristani et al. 2016; Zheng, Zheng, and Yang 2017).

- **Market-1501** contains altogether 32,688 images of 1,501 labeled pedestrians, which were captured under 6 camera viewpoints in a campus. Deformable Part Model (DPM) (Felzenszwalb, McAllester, and Ramanan 2008) is employed to produce pedestrian bounding boxes. This dataset is split into two non-overlapping partitions: 12,936 images from 751 identities (including 1 background category) for training and 19,732 images from 750 identities for testing. During testing, 3,368 images are chosen as query images. We adopt single-query evaluation mode in all experiments.
- **CUHK03** contains 14,096 pedestrian images of 1,467 identities. Each person image in this dataset was captured from two different cameras in the CUHK campus. It provides both DPM-detected and hand-marked bounding boxes. In this paper, we report experimental results on both image sets. We utilize the more challenging train/test split protocol proposed in (Zhong et al. 2017a) where 767 identities are used for training and the rest 700 for testing.
- **DukeMTMC-reID** is a subset of Duke-MTMC for ReID. The images were captured with 8 cameras for cross-camera tracking. It contains 16,522 training images from 702 identities, 2,228 queries from the other 702 identities and 17,661 gallery images. On this dataset, we also test our method in the single-query setting.

Evaluation Metrics

Following most existing person ReID works, we use two evaluation metrics to evaluate the performance of our method. One is the Cumulated Matching Characteristics (CMC), which considers ReID as a ranking problem. The other is mean average precision (mAP), which considers ReID as a retrieval problem.

Implementation Details

Our implementation is based on PyTorch (Paszke et al. 2017). The backbone ResNet-50 is pre-trained on ImageNet (Russakovsky et al. 2015). We use the same data augmentation across all experiments and on all datasets unless otherwise noted. The training images are randomly cropped with a ratio uniformly sampled from $[0.8, 1)$ and resized to 288×144. Random erasing (Zhong et al. 2017b) and random flipping are applied on resized images with a probability of 0.5. The hyper-parameters of random erasing data augmentation are set the same as (Zhong et al. 2017b). The number of persons P per-batch and number of images per-person K are set to 20 and 4 respectively. Hence, the mini-batch size is 80. For LITM, the base and incremental margins are set as $m_0 = 4$, $m_1 = 7$, $m_2 = 10$.

Stride	Pooling	Fully Conv.	Rank-1	mAP
2	GAP		85.3	70.6
2	GAP	✓	85.4	71.6
2	GMP	✓	89.7	75.9
1	GMP	✓	90.1	77.9

Table 1: Performance improvements of the proposed TriNet-S over the TriNet baseline on the Market-1501 dataset. The performance of TriNet in this table is slightly better than that reported in (Hermans, Beyer, and Leibe 2017) because of the random erasing data augmentation we adopt.

We use the Adam optimizer (Kingma and Ba 2014) with $\epsilon = 10^{-3}$, $\beta_1 = 0.99$ and $\beta_2 = 0.999$. The network is trained for 300 epochs in total. And a piecewise learning rate schedule is utilized, where it is fixed to 2×10^{-4} in the first 150 epochs and decayed exponentially in the rest 150 epochs.

$$lr(t) = \begin{cases} 2 \times 10^{-4} & \text{if } t \leq 150 \\ 2 \times 10^{-4} \times 10^{-3 \times \frac{t-150}{150}} & \text{if } 150 < t \leq 300 \end{cases}$$

Improvements over Triplet Loss Baseline

We first report the performance gains brought by our tweaks to the network architecture in TriNet-S on the Market-1501 dataset. As shown in Table 1, after removing the trailing fully connected layers and making the network fully convolutional, mAP gets improved by 1% from 70.6% to 71.6%. Replacing GAP with GMP brings more than 4% performance gains in terms of both Rank-1 accuracy and mAP. By reducing the stride of the conv5_x block from 2 to 1 and thus increasing the feature map resolution, we obtain an extra 2% mAP gain. Compared with the TriNet baseline, the proposed TriNet-S improves mAP by 7.3% and Rank-1 accuracy by 4.8%.

The proposed LITM training strategy is agnostic to the choice of network architecture. To validate the effectiveness of LITM, we apply it to both TriNet and TriNet-S. As shown in Table 2, LITM is able to improve TriNet by 6.3% and 3.1% in terms of mAP and Rank-1 accuracy respectively on the Market-1501 dataset. Even though TriNet-S have already greatly improved the performance over the TriNet baseline, LITM still boosts mAP by 4.4% and Rank-1 accuracy by 2.5%. Similar performance boosts are observed on the CUHK03 and DukeMTMC-reID datasets, which indicates that our approach generalizes well across different scenarios.

Network	LITM	Market-1501		CUHK03 (labeled)		DukeMTMC-reID	
		Rank-1	mAP	Rank-1	mAP	Rank-1	mAP
TriNet (Hermans, Beyer, and Leibe 2017)		84.9	69.1	55.2†	54.3†	76.4†	60.4†
TriNet (Hermans, Beyer, and Leibe 2017)	✓	88.0	75.4	60.9	59.3	79.2	65.9
TriNet-S		90.1	77.9	63.5	61.7	82.8	70.2
TriNet-S	✓	**92.6**	**82.3**	**73.1**	**71.0**	**84.8**	**74.4**

Table 2: Performance improvements of LITM to both TriNet and TriNet-S. † indicates reproduced results by us using the same training configuration as the TriNet paper.

Measure (%)	Rank-1	mAP
Pose-transfer (Liu et al. 2018)	87.7	68.9
AOS (Huang et al. 2018)	86.5	70.4
MGCAM (Song et al. 2018)	83.8	74.3
MLFN (Chang et al. 2018)	90.0	74.3
HA-CNN (Li, Zhu, and Gong 2018)	91.2	75.7
AlignedReID* (Zhang et al. 2017)	91.8	79.3
Deep-Person* (Bai et al. 2017)	92.3	79.6
GCSL (Chen et al. 2018)	93.5	81.6
PCB+RPP* (Sun et al. 2018)	93.8	81.6
GSRW (Shen et al. 2018a)	92.7	82.5
SphereReID* (Fan et al. 2018)	**94.4**	83.6
LITM	92.6	82.3
LITM+GHIS	93.9	**83.9**

Table 3: Performance comparison on the Market-1501 dataset. * denotes unpublished work on arXiv.

Data Type	Labeled		Detected	
Measure (%)	Rank-1	mAP	Rank-1	mAP
HA-CNN (2018)	44.4	41.0	41.7	38.6
Pose-transfer (2018)	45.1	42.0	41.6	38.7
MGCAM (2018)	50.1	50.2	46.7	46.9
AOS (2018)	-	-	47.1	43.3
MLFN (2018)	54.7	49.2	52.8	47.8
REDA* (2017b)	58.1	53.8	55.5	50.7
PCB+RPP* (2018)	-	-	63.7	57.5
LITM	73.1	71.0	71.0	68.6
LITM+GHIS	**74.2**	**71.7**	**71.8**	**69.1**

Table 4: Performance comparison on the CUHK03 dataset. * denotes unpublished work on arXiv.

Measure (%)	Rank-1	mAP
Pose-transfer (Liu et al. 2018)	78.5	56.9
AOS (Huang et al. 2018)	79.2	62.1
MLFN(Chang et al. 2018)	81.0	62.8
HA-CNN (Li, Zhu, and Gong 2018)	80.5	63.8
Deep-Person* (Bai et al. 2017)	80.9	64.8
GSRW (Shen et al. 2018a)	80.7	66.4
SphereReID* (Fan et al. 2018)	83.9	68.5
PCB+RPP* (Sun et al. 2018)	83.3	69.2
GCSL (Chen et al. 2018)	84.9	69.5
LITM	84.8	74.4
LITM+GHIS	**85.9**	**74.5**

Table 5: Performance comparison on the DukeMTMC-reID dataset. * denotes unpublished work on arXiv.

Comparisons with the State-of-the-arts

Results on Market-1501 As shown in Table 3, GHIS further brings 1.3% and 1.6% improvements for Rank-1 accuracy and mAP respectively. Compared with 11 recently proposed methods on the Market-1501 dataset, our final result yields the best mAP (83.9%) and comparable Rank-1 accuracy (93.9%) to SphereReID. Although SphereReID achieves the best Rank-1 accuracy, its optimization is very sensitive to hyper-parameter settings. For example, a carefully designed learning rate warming up schedule is required. In GSRW, testing images are fed into the network in a pairwise manner, which is much more time consuming than our approach. PCB+RPP is trained with a three-stage process with fine-tuning, which is not an end-to-end method.

Results on CUHK03 We choose the new training/testing split protocol proposed in (Zhong et al. 2017a) instead of the original protocol for convenience. A comparison our approach with recent methods following the same evaluation protocol are listed in Table 4. LITM+GHIS outperforms the 2nd best approach (PCB+RPP) by 8.1% (71.8% vs. 63.7%) for Rank-1 accuracy and 11.6% (69.1% vs. 57.5%) for mAP. The significant performance advantage fully validates the superiority of the proposed LITM and GHIS over existing methods.

Results on DukeMTMC-reID Compared with Market-1501, pedestrian images from this dataset have more variations in illumination and background because of wider camera views and more complex scene layout. On this challenging dataset, our LITM+GHIS approach again outperforms all recent methods by a large margin as shown in Table 5. Notably, our approach outperforms SphereReID (Fan et al. 2018) by 2.0% and 6.0% in terms of Rank-1 accuracy and mAP respectively, which indicates that our improvements on training strategy and network architecture are general and work well in a wide variety of scenarios.

Ablation Studies

To further investigate the design choices of our approach, we perform extensive ablation studies on the Market-1501 dataset. In particular, we compare the behaviors of GAP and GMP, the impact of incremental triplet margin and alternative LITM structures.

Pooling	$\bar{d}^{ap}$	$\bar{d}^{an}$	$\bar{d}^{an} - \bar{d}^{ap}$
GAP	10.5	25.1	14.6
GMP	40.9	66.2	25.3

Table 6: The mean distance of positive and negative pairs regarding different global pooling methods in TriNet.

Method	pool	Rank-1	mAP
TriNet	GAP	85.4	71.6
	GMP	89.7	75.9
LITM	GAP	89.3	77.9
	GMP	92.6	82.3

Table 7: Performance improvements of GMP over GAP on the Market-1501 dataset.

GAP vs. GMP

By analyzing the feature maps before global pooling, we find that the great majority of elements are close to 0. Therefore, the average operation in GAP would greatly reduce the magnitude of feature vectors, which weakens the feature discriminativeness. To prove the hypothesis, we compute the mean distance of positive and negative pairs after the training converges. Table 6 shows a comparison of GAP and GMP in terms of the mean distance. By replacing GAP with GMP, mean distance of both positive and negative pairs gets significantly increased. At the same time, although trained with the same triplet margin, the gap between $\bar{d}^{an}$ and $\bar{d}^{ap}$ is enlarged from 14.6 to 25.3. And the performance also gets significantly improved as shown in Table 7. This again verifies that larger margin leads to better feature embedding.

Impact of Incremental Margin

To validate that the multi-stage triplet losses with incremental margins have learned increasingly better feature embedding, we first compare the mean distance of positive and negative pairs at different stages. As show in Table 8, from $f_0(\cdot)$ to $f_2(\cdot)$ the distance of positive pairs $\bar{d}^{ap}$ and negative pairs $\bar{d}^{ap}$, as well as their gap are progressively increased. In particular, the distance gap between positive and negative pairs increases from 25.9 to 32.7. In terms of performance, as shown in Table 9, shifted features are also superior over base features. With a single iteration of feature shifting, $f_1(\cdot)$ boosts mAP from 80.9% to 82.2%. While improvement from more iterations is marginal.

To validate the effectiveness of our incremental triplet margin strategy, we compare it with the one-stage large-margin triplet loss. Specifically, we train TriNet-S with different margins. From Table 9, we can see that performance of TriNet-S gets improved when the margin increases from 1 to 4, but degrades for larger margins. Notably, the performances of $f_0(\cdot)$, $f_1(\cdot)$ and $f_2(\cdot)$ consistently outperform their TriNet-S counterparts with the same margin value, which clearly demonstrates the superiority of the proposed LITM method.

Feature	$\bar{d}^{ap}$	$\bar{d}^{an}$	$\bar{d}^{an} - \bar{d}^{ap}$
$f_0(\cdot)$	59.0	84.9	25.9
$f_1(\cdot)$	68.9	99.1	30.2
$f_2(\cdot)$	70.9	103.6	32.7

Table 8: The mean distance of positive and negative pairs of features at different stages of LITM.

	m	Rank-1	Rank-5	Rank-10	mAP
TriNet-S	1	90.1	94.8	96.5	77.9
	4	90.9	96.6	97.5	79.1
	7	90.2	96.4	97.0	78.2
	10	89.9	95.2	96.2	77.7
$f_0(\cdot)$	4	92.1	96.9	98.0	80.9
$f_1(\cdot)$	7	92.6	97.1	98.5	82.2
$f_2(\cdot)$	10	92.6	97.5	98.5	82.3

Table 9: Performance of feature embeddings at different stages of LITM and comparison with TriNet-S with various margins on the Market-1501 dataset.

Measure (%)	Rank-1	Rank-5	Rank-10	mAP
LITM-C5C5C5	92.0	97.0	98.2	81.2
LITM-C3C4C5	90.8	96.3	97.9	79.4
LITM-C5C4C3	92.6	97.5	98.5	82.3

Table 10: Performance comparison of different LITM structures on the Market-1501 dataset.

Alternative LITM Structures

We further compare the current LITM structure in Figure 2 with two alternatives.

- LITM-C5C5C5: the base features as well as shifted features are learned from the conv5_x block.
- LITM-C3C4C5: the base features and shifted features are learned from the conv3_x, conv4_x and conv5_x blocks respectively.
- LITM-C5C4C3: the current structure we use in our experiments.

Table 10 shows the results. The C5C4C3 setting outperforms C5C5C5, which validates that mid-level features indeed help. While C3C4C5 is the worst. The reason is that a decent base feature embedding learned from high-level feature maps is critical.

Conclusion

In this paper, we verify that triplet loss is an effective tool to learn discriminative features for person ReID. However, existing training framework is far from optimal. By learning incremental triplet margin, global hard identity searching and a better network architecture, we make significant performance improvement and achieve state-of-the-art performances on common person ReID datasets. Our improvements to triplet loss may also apply to other related visual tasks, such as face recognition and object retrieval. We leave this as future work.

References

Bai, X.; Yang, M.; Huang, T.; Dou, Z.; Yu, R.; and Xu, Y. 2017. Deep-person: Learning discriminative deep features for person re-identification. *arXiv preprint arXiv:1711.10658*.

Chang, X.; Hospedales, T. M.; and Xiang, T. 2018. Multi-level factorisation net for person re-identification. In *CVPR*.

Chen, D.; Xu, D.; Li, H.; Sebe, N.; and Wang, X. 2018. Group consistent similarity learning via deep crf for person re-identification. In *CVPR*.

Cheng, D.; Gong, Y.; Zhou, S.; Wang, J.; and Zheng, N. 2016. Person re-identification by multi-channel parts-based cnn with improved triplet loss function. In *CVPR*.

Fan, X.; Jiang, W.; Luo, H.; and Fei, M. 2018. Sphereid: Deep hypersphere manifold embedding for person re-identification. *arXiv preprint arXiv:1807.00537*.

Felzenszwalb, P.; McAllester, D.; and Ramanan, D. 2008. A discriminatively trained, multiscale, deformable part model. In *CVPR*, 1–8. IEEE.

Guo, Y., and Cheung, N.-M. 2018. Efficient and deep person re-identification using multi-level similarity. In *CVPR*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *ICCV*, 770–778.

Hermans, A.; Beyer, L.; and Leibe, B. 2017. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*.

Huang, H.; Li, D.; Zhang, Z.; Chen, X.; and Huang, K. 2018. Adversarially occluded samples for person re-identification. In *CVPR*.

Kingma, D., and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *NIPS*, 1097–1105.

LeCun, Y.; Boser, B.; Denker, J. S.; Henderson, D.; Howard, R. E.; Hubbard, W.; and Jackel, L. D. 1989. Backpropagation applied to handwritten zip code recognition. *Neural computation* 1(4):541–551.

Li, W.; Zhao, R.; Xiao, T.; and Wang, X. 2014. Deepreid: Deep filter pairing neural network for person re-identification. In *CVPR*, 152–159.

Li, W.; Zhu, X.; and Gong, S. 2018. Harmonious attention network for person re-identification. In *CVPR*.

Liu, J.; Ni, B.; Yan, Y.; Zhou, P.; Cheng, S.; and Hu, J. 2018. Pose transferrable person re-identification. In *CVPR*.

Paszke, A.; Gross, S.; Chintala, S.; Chanan, G.; Yang, E.; DeVito, Z.; Lin, Z.; Desmaison, A.; Antiga, L.; and Lerer, A. 2017. Automatic differentiation in pytorch.

Qian, X.; Fu, Y.; Jiang, Y.-G.; Xiang, T.; and Xue, X. 2017. Multi-scale deep learning architectures for person re-identification. In *ICCV*.

Ristani, E.; Solera, F.; Zou, R.; Cucchiara, R.; and Tomasi, C. 2016. Performance measures and a data set for multi-target, multi-camera tracking. In *ECCV*, 17–35. Springer.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; Berg, A. C.; and Fei-Fei, L. 2015. ImageNet Large Scale Visual Recognition Challenge. *IJCV* 115(3):211–252.

Schroff, F.; Kalenichenko, D.; and Philbin, J. 2015. Facenet: A unified embedding for face recognition and clustering. *CVPR* 815–823.

Shen, Y.; Li, H.; Xiao, T.; Yi, S.; Chen, D.; and Wang, X. 2018a. Deep group-shuffling random walk for person re-identification. In *CVPR*.

Shen, Y.; Xiao, T.; Li, H.; Yi, S.; and Wang, X. 2018b. End-to-end deep kronecker-product matching for person re-identification. In *CVPR*.

Shrivastava, A.; Gupta, A.; and Girshick, R. 2016. Training region-based object detectors with online hard example mining. In *CVPR*.

Song, C.; Huang, Y.; Ouyang, W.; and Wang, L. 2018. Mask-guided contrastive attention model for person re-identification. In *CVPR*.

Su, C.; Li, J.; Zhang, S.; Xing, J.; Gao, W.; and Tian, Q. 2017. Pose-driven deep convolutional model for person re-identification. In *ICCV*.

Sun, Y.; Zheng, L.; Yang, Y.; Tian, Q.; and Wang, S. 2018. Beyond part models: Person retrieval with refined part pooling (and a strong convolutional baseline). *arXiv preprint arXiv:1711.09349v3*.

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

Xiao, T.; Li, H.; Ouyang, W.; and Wang, X. 2016. Learning deep feature representations with domain guided dropout for person re-identification. In *CVPR*.

Yao, H.; Zhang, S.; Zhang, Y.; Li, J.; and Tian, Q. 2017. Deep representation learning with part loss for person re-identification. *arXiv preprint arXiv:1707.00798*.

Zhang, X.; Luo, H.; Fan, X.; Xiang, W.; Sun, Y.; Xiao, Q.; Jiang, W.; Zhang, C.; and Sun, J. 2017. Alignedreid: Surpassing human-level performance in person re-identification. *arXiv preprint arXiv:1711.08184v2*.

Zhao, L.; Li, X.; Zhuang, Y.; and Wang, J. 2017. Deeply-learned part-aligned representations for person re-identification. In *ICCV*.

Zheng, L.; Shen, L.; Tian, L.; Wang, S.; Wang, J.; and Tian, Q. 2015. Scalable person re-identification: A benchmark. In *ICCV*, 1116–1124.

Zheng, Z.; Zheng, L.; and Yang, Y. 2017. Unlabeled samples generated by gan improve the person re-identification baseline in vitro. In *ICCV*.

Zhong, Z.; Zheng, L.; Cao, D.; and Li, S. 2017a. Re-ranking person re-identification with k-reciprocal encoding. In *CVPR*.

Zhong, Z.; Zheng, L.; Kang, G.; Li, S.; and Yang, Y. 2017b. Random erasing data augmentation. *arXiv preprint arXiv:1708.04896* 3.