Mining Hard Samples Globally and Efficiently for Person Reidentification

Hao Sheng, *Member, IEEE*, Yanwei Zheng, Wei Ke, *Member, IEEE*, Dongxiao Yu,
Xiuzhen Cheng, *Fellow, IEEE*, Weifeng Lyu, and Zhang Xiong

***Abstract*—Person reidentification (ReID) is an important application of Internet of Things (IoT). ReID recognizes pedestrians across camera views at different locations and time, which is usually treated as a ranking task. An essential part of this task is the hard sample mining. Technically, two strategies could be employed, i.e., global hard mining and local hard mining. For the former, hard samples are mined within the entire training set, while for the latter, it is done in mini-batches. In literature, most existing methods operate locally. Examples include batch-hard sample mining and semihard sample mining. The reason for the rare use of global hard mining is the high computational complexity. In this article, we argue that global mining helps to find harder samples that benefit model training. To this end, this article introduces a new system to: 1) efficiently mine hard samples (positive and negative) from the entire training set and 2) effectively use them in training. Specifically, a ranking list network coupled with a multiplet loss is proposed. On the one hand, the multiplet loss makes the ranking list progressively created to avoid the time-consuming initialization. On the other hand, the multiplet loss aims to make effective use of the hard and easy samples during training. In addition, the ranking list makes it possible to globally and effectively mine hard positive and negative samples. In the experiments, we explore the performance of the global and local sample mining methods, and the effects of the semihard, the hardest, and the randomly selected samples. Finally, we demonstrate the validity of our theories using various public data sets and achieve competitive results via a quantitative evaluation.**

***Index Terms*—Convolutional neural networks, hard sample mining, listwise rank learning, multiplet loss, person reidentification (ReID).**

I. Introduction

Internet of Things (IoT) has been pervasive in recent years, and many IoT applications have been well developed. Among these applications, person reidentification (ReID) is extensively studied. Specifically, ReID recognizes pedestrians across camera views at different locations and time [1]. ReID underpins many crucial applications in video surveillance, such as long-term cross-camera tracking [2], content-based image retrieval [3], video retrieval [4], multicamera behavior analysis [5], etc. But ReID has been a challenging task due to the variation of illuminations, occlusions, viewpoints, background clutters, and image resolutions [6].

Recent studies usually treat ReID as a ranking task [7]–[13], which can be solved using three kinds of frameworks depending on how many samples are considered at a time in the loss function. The pointwise approach uses the classification network [14]–[18] to classify images into person categories, and then extracts features to calculate and rank the similarities of images. In this method, a multiclassifier is used to learn the ranking scores, and the ranking is produced by combining the outputs of the classifiers [19]. The pairwise approach uses the Siamese network [20]–[24]. It takes two images as inputs and then generates either a similarity score between the two images or a classification of an image pair, which depicts either the same pedestrian or a group of different pedestrians. Its main focus is on how to effectively concatenate the cross corresponding pairs into one. The listwise approach uses the triplet [7], [9], [11], [25], quadruplet [26], or DeepList [27] framework. The triplet network uses three images as the inputs—usually an anchor, a positive (matched with the anchor), and a negative (mismatched with the anchor) images—and outputs features by improving the loss function that minimizes the distance of the matched images, while maximizes that of the mismatched ones. The quadruplet network is an improvement over the triplet. Comparing with the triplet loss, the quadruplet network uses another pair of mismatched images to get a larger interclass variance and a smaller intraclass variance. DeepList implements a listwise loss function and uses a ranking list to train samples.

An essential part of ranking task learning is the hard sample mining [28], [29]. Although these models have achieved

Manuscript received December 12, 2019; revised February 25, 2020; accepted March 8, 2020. Date of publication March 13, 2020; date of current version October 9, 2020. This work was supported in part by the National Key Research and Development Program of China under Grant 2019YFB2101600; in part by the National Natural Science Foundation of China under Grant 61861166002, Grant 61872025, and Grant 61635002; in part by the Science and Technology Development Fund of Macau SAR (File no. 0001/2018/AFJ) Joint Scientific Research Project; in part by the Macao Science and Technology Development Fund under Grant 138/2016/A3; and in part by the Fundamental Research Funds for the Central Universities and the Open Fund of the State Key Laboratory of Software Development Environment under Grant SKLSDE2019ZX-04. *(Corresponding author: Yanwei Zheng.)*

Hao Sheng, Weifeng Lyu, and Zhang Xiong are with the State Key Laboratory of Software Development Environment, School of Computer Science and Engineering and Beijing Advanced Innovation Center for Big Data and Brain Computing, Beihang University, Beijing 100191, China (e-mail: shenghao@buaa.edu.cn; lwf@buaa.edu.cn; xiongz@buaa.edu.cn).

Yanwei Zheng is with the State Key Laboratory of Software Development Environment, School of Computer Science and Engineering, Beihang University, Beijing 100191, China, and also with the School of Computer Science and Technology, Shandong University, Qingdao 266237, China (e-mail: zhengyw@sdu.edu.cn).

Dongxiao Yu and Xiuzhen Cheng are with the School of Computer Science and Technology, Shandong University, Qingdao 266237, China (e-mail: dxyu@sdu.edu.cn; xzcheng@sdu.edu.cn).

Wei Ke is with the School of Applied Sciences, Macao Polytechnic Institute, Macao SAR 999078, China (e-mail: wke@ipm.edu.mo).

Digital Object Identifier 10.1109/JIOT.2020.2980549

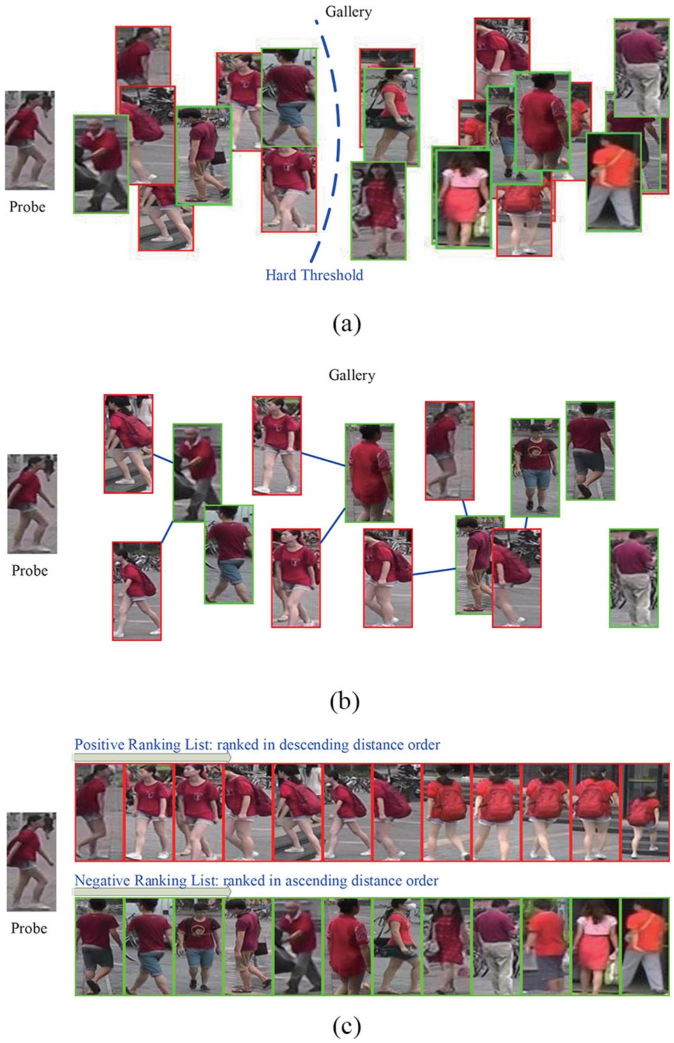

Gallery
Probe
Hard Threshold
(a)
Gallery
Probe
(b)
Positive Ranking List: ranked in descending distance order
Probe
Negative Ranking List: ranked in ascending distance order
(c)

Fig. 1. Hard sample mining. Images with red and green boundary denote positive and negative gallery samples, respectively. (a) Hard negative sample mining with a threshold. The negative samples are the batch-hard samples that are closer to the probe image than a threshold. (b) Hard negative sample mining with the semihard samples. The semihard samples are the negative samples that are the closest to the probe image but further away than their corresponding positive samples from the probe image. (c) Hard sample mining with the ranking list. The closer to the top of the lists, the harder is the sample.

impressive results on existing ReID data sets, there are still some problems.

1) Some researchers mine the hardest samples from a mini-batch. Hermans *et al.* [29] randomly sampled classes and images for a batch and selected the hardest positive and negative samples within the batch to form the triplets, which is called batch hard. Kong *et al.* [30] selected the hard samples whose scores are higher than a threshold, as shown in Fig. 1(a).
2) Some others think that the hardest sample in a mini-batch leads to bad local optimal solutions in training. Schroff *et al.* [28] chose the negative samples that are the closest to the anchors but further away than the positive samples from the anchors, which are called semihard samples, as shown in Fig. 1(b). A negative sample can be a semihard sample of multiple positive ones, or it may not be a semihard sample of any positive samples. Chen *et al.* [26] chose the negative samples that have a smaller distance than an adaptive margin. These are local mining methods that are based on a mini-batch and do not consider the global sample relationship based on the entire data set.
3) The triplet loss [28] pays more attention to obtain the correct orders on the training set, whereas the quadruplet loss [26] can achieve a larger interclass variance. However, they treat each sample equally, which neglects the implicit priority between samples.

In this article, we introduce a listwise ranking network, called LoopNet, where a positive and a negative ranking list are preserved for global hard sample mining. The positive and negative lists are sorted by the distance in descending and ascending orders, respectively, as shown in Fig. 1(c), and hence the hardest positive and negative samples are all at the tops of their respective lists. The list is updated online, i.e., when a batch of samples is trained in an iteration, the order of these samples is sorted again with the altered distance. This ensures that the selected samples at the top of the list are always the hardest ones. In addition, the ranking lists are the output of the distance calculation layer (near the end of the network) and the input of the sampling layer (the beginning of the network), which constructs a loop network (as LoopNet for short).

Building the ranking list requires every probe sample to traverse every gallery sample, which is time consuming and unacceptable. To address this problem, we propose a multiplet loss, which is capable of using multiple positive and negative samples as a whole group to conduct training. By using multiple gallery samples for each probe image, we can concurrently use some mined hard samples and some randomly selected ones in a mini-batch. Then, the randomly selected samples could be added to the list, so that the ranking list is initialized progressively. The multiplet loss also considers the priority of samples for each identity, which leads the harder samples to have greater effects.

In summary, the main contributions of this article are threefold.

1) We propose a loop network with a ranking list, which can be used to choose the hard positive and negative samples globally.
2) We introduce a multiplet loss that uses multiple positive and negative samples for each identity, which can initialize the ranking list progressively and update it in real time.
3) We explore the performance of the global (in ranking list) and local (in a mini-batch) sample mining methods, and the effects of the semihard samples, the hardest samples, and the randomly selected samples.

The remainder of this article is organized as follows. Section II discusses the related work about the loss function improvement and hard sample mining in ReID. In Section III, the LoopNet model is presented, including the ranking list model, the multiplet loss design, the sample matching, and the backpropagation calculation. Section IV introduces some details of the implementation, including the progressive building of the ranking list, and the network pipelines. Section V provides the experimental results. Section VI concludes this article and outlines the future work.

II. Related Work

A. Loss Function Improvement

In listwise ranking learning methods, many triplet loss functions with different networks are designed. Ding *et al.* [9] fed three images into the network, where two images belonged to one person and the third image did not belong to anyone. Then, the loss function was devised to make the L_2 feature distance of the matched pair smaller than the mismatched pair in each triplet. Cheng *et al.* [7] designed another loss function to train the network models in order to make the distance between the matched pairs less than a predefined threshold and less than the mismatched pairs in the learned feature space. Liu *et al.* [31] focused on parts of person image pairs after taking a few shots of them and adaptively comparing their appearances in triplet networks. Wang *et al.* [11] used two subnetworks for a pair of input images, and then two single-image representations and a cross-image representation were calculated. Finally, the triplet comparison objectives were combined to improve matching performance. Another improvement in [11] was that a learned metric rather than the traditional Euclidean distance was used for the triplet loss.

Zhou *et al.* [32] used the point-to-set (P2S) metric to replace the point-to-point (P2P) distances, which jointly minimized the intraclass distance and maximized the interclass one. In that triplet loss, the distances of the two-positive-image pair and the two-negative-image pair were decreased, while that of the positive-negative-image pair was increased.

Chen *et al.* [26] designed a quadruplet loss, where a new mismatched image with the probe was added to the triplet tuple to increase the distance from the negative pairs, which led to a model output with a larger interclass variance and a smaller intraclass variance compared to the triplet loss. In addition, a normalized 2-D output that was generated by a fully connected layer and Softmax was used as the distance metric in [26].

Wang *et al.* [27] replaced the image pairs with ranking lists as training samples and developed a listwise loss function with an adaptive margin to assign larger margins to harder negative samples. In it, the Plackett–Luce permutation probability model [33] and the likelihood loss function were used in the ListMLE [34] training method. It is worth noting that this method was based on the single-shot assumption, i.e., only one gallery image had the same identity as the probe image. Chen *et al.* [8] proposed a learning-to-rank loss to minimize the costs corresponding to the poor rankings of the gallery, in which the similarity differences between positive and negative matching images were accumulated.

B. Hard Sample Mining

Hard sample selection is a crucial and difficult task for fast convergence [28]. Ahmed *et al.* [22] randomly selected a negative set to train the network, and then the trained model was used to select the hard negative pairs to retrain the fully connected layer of the network. This is an offline hard sample mining method, and it is time consuming in the training stage.

The online hard sample mining methods usually choose hard samples in a mini-batch. Hermans *et al.* [29] randomly sampled images for a batch and selected the hardest positive and negative samples within the batch to form a triplet, which was called batch hard. Wang and Gupta [35] first randomly chose the negative samples for ten epochs, and then calculated the loss of all negative matches in a batch, and finally selected the top K ones with the highest losses. In FaceNet [28], all anchor-positive pairs were used in a mini-batch, while the negatives with the farther distances than the positives were selected, which were called semihard samples. Xiao *et al.* [56] proposed a margin sample mining loss, where the maximum distance of the positive pairs and the minimum distance of the negative pairs in a batch were selected to calculate the final loss.

The threshold is another strategy to mine hard samples. Chen *et al.* [26] used the combined mean of the positive pair distances and negative pair distances to set an adaptive threshold to mine the hard samples. Wang *et al.* [27] introduced an adaptive shifting parameter in a listwise loss function, which could assign larger margins to harder negative samples.

Some set-based methods are used for reranking approaches. Liu *et al.* [36] eliminated the hard negative label matches based on the reciprocal nearest neighbor. Zhong *et al.* [37] calculated the k-reciprocal nearest neighbors of R, and the union part of the $(1/2)k$-reciprocal nearest neighbors of each candidate in R to recall the hard positive gallery images.

Some probability or weight updating methods are used to control the influence of hard samples. Ye *et al.* [38] introduced a label reweighting scheme to filter out the false positives and easy negatives. In metric learning, Zhou *et al.* [39] used the local hard negative samples to provide tight constraints to fine-tune the metric locally. Li *et al.* [20] updated a probability score according to the previous epoch to increase the selection probabilities of those negative samples that are not selected over a long time. Triantafyllidou *et al.* [40] thought that the model should learn easier positive samples first and then the harder ones. They determined a positive sample's difficulty level using a score produced by the network and progressively added slightly harder positive samples to the training set. Dong *et al.* [41] first generated easy samples and then improved the poorly initialized model. As the model becomes more discriminative, challenging but reliable samples are selected.

III. LoopNet Model

Our LoopNet is based on the ranking list, so we first introduce the positive and negative ranking list model. Then, the multiplet loss is presented. Finally, the sample matching and backpropagation problems are introduced.

A. Ranking List

The ranking list is designed to mine hard samples efficiently. If we use the distance between an anchor (probe image) and a gallery image to measure the similarity of two images, the distance between the anchor and the hard positive samples should be larger than the easy positive sample, whereas that between the anchor and the hard negative sample should be smaller than the easy negative sample, i.e., the hard positive sample is farther from the anchor, whereas the hard negative is closer to the anchor. Because the rules of hard positive and negative samples are different, we build two ranking lists for both of them. The positive and negative lists are sorted by the distance in descending and ascending orders, respectively, and thus the hardest positive and negative samples are at the tops of the respective lists.

Because each person has many images, we need to identify each image when building a ranking list. If there are n_p and n_g images in the probe and gallery sets, respectively, the permutation numbers of two sets are $n_p!$ and $n_g!$, respectively. We arbitrarily choose a permutation as the standard one to define the identity of the image.

Suppose the probe permutation is $P = <p_1, p_2, \ldots, p_{n_p}>$, and the gallery one is $G = <g_1, g_2, \ldots, g_{n_g}>$, where p_i and g_j are the ith and the jth image of the selected probe and gallery permutations, respectively. The label of each image of p_i and g_j, which is the identity of one person rather than the identity of an image, is represented as $\mathrm{id}(p_i)$ and $\mathrm{id}(g_i)$, respectively, and the distance between p_i and g_j is $f(p_i, g_j)$.

After defining these symbols, the positive ranking list can be defined as

$$\begin{aligned} \pi_i^+ &= <\pi_i^+(1), \pi_i^+(2), \ldots, \pi_i^+\left(m^+\right)> \\ \text{s.t.} \quad \mathrm{id}\left(p_{\pi_i^+(k)}\right) &= \mathrm{id}(p_i) \\ f\left(p_i, p_{\pi_i^+(j)}\right) &\geq f\left(p_i, p_{\pi_i^+(j+1)}\right) \end{aligned} \tag{1}$$

where i is the image index in the selected probe permutation, and $\pi_i^+(k)$ is the image index of the selected gallery permutation at position k in the ranking list. For example, given the fifth probe image list $\pi_5^+ = <4, 7, 3>$, then $\pi_5^+(2) = 7$ denotes that the seventh gallery image is in the second position in the list, and $(\pi_5^+)^{-1}(7) = 2$ denotes that the second position in the ranking list is the seventh image in the selected gallery permutation. The condition $\mathrm{id}(p_{\pi_i^+(k)}) = \mathrm{id}(p_i)$ ensures the gallery images in the list of the ith row have the same label with its probe image π_i^+, where $k = 1, 2, \ldots, m^+$. The condition $f(p_i, p_{\pi_i^+(j)}) \geq f(p_i, p_{\pi_i^+(j+1)})$ ensures that the positive ranking list is sorted in a descending order, where $j = 1, 2, \ldots, m^+ - 1$. The gallery images in the ith row, which have the same label with the ith probe image, are less than the images in the gallery set, which means that $m^+ < n_g$.

Similarly, the negative ranking list is defined as

$$\begin{aligned} \pi_i^- &= <\pi_i^-(1), \pi_i^-(2), \ldots, \pi_i^-(m_n)> \\ \text{s.t.} \quad \mathrm{id}\left(p_{\pi_i^-(k)}\right) &\neq \mathrm{id}(p_i) \\ f\left(p_i, p_{\pi_i^-(j)}\right) &\leq f\left(p_i, p_{\pi_i^-(j+1)}\right). \end{aligned} \tag{2}$$

The condition $\mathrm{id}(p_{\pi_i^-(k)}) \neq \mathrm{id}(p_i)$ ensures that the gallery images in the list of the ith row have different labels from its probe image π_i^-, where $k = 1, 2, \ldots, m^-$ and $m^- < n_g$. The condition $f(p_i, p_{\pi_i^-(j)}) \leq f(p_i, p_{\pi_i^-(j+1)})$ ensures that the negative ranking list is sorted in an ascending order, where $j = 1, 2, \ldots, m^- - 1$

According to these definitions, the harder the sample is, the closer it is to the top of the list, i.e., $\pi_i^+(j)$ and $\pi_i^-(j)$ are harder than $\pi_i^+(k)$ and $\pi_i^-(k)$, respectively, when $j < k$.

B. Multiplet Loss

Inspired by the quadruplet loss [26], we design a multiplet loss to achieve a smaller intraclass variance and a larger interclass variance. In addition, the multiplet loss also considers the priority of samples for each image. Finally, the multiplet loss also supports hard sample mining, which will be discussed in Section IV.

For the multiplet loss, we choose n positive images and n negative images from the gallery set for one anchor in a mini-batch, which combine to form a $(2n + 1) - \textit{tuple}$ of $<p_i, g_{i,1}^+, g_{i,2}^+, \ldots, g_{i,n}^+, g_{i,1}^-, g_{i,2}^-, \ldots, g_{i,n}^->$, where p_i, $g_{i,j}^+$, and $g_{i,k}^-$ are the probe, positive gallery, and negative gallery samples, respectively, and n is the *dimension* in the multiplet loss. The probe sample, the jth positive sample and the jth negative sample combine to form a triplet $<p_i, g_{i,j}^+, g_{i,j}^->$, and the triplet and the $(j+1)$th negative sample combine to form a quadruplet $<p_i, g_{i,j}^+, g_{i,j}^-, g_{i,j+1}^->$. The idea of the multiplet loss is to distance negative images from positive images, and distance the negative images from each other at the same time, which is shown in

$$\begin{aligned} L_i = &\sum_{j=1}^{n}\left[f\left(p_i, g_{i,j}^+\right) - f\left(p_i, g_{i,j}^-\right) + \alpha_j\right]_+ \\ &+ \sum_{j=1}^{n-1}\left[f\left(p_i, g_{i,j}^+\right) - f\left(g_j^-, g_{i,j+1}^-\right) + \beta_j\right]_+ \end{aligned} \tag{3}$$

where $[x]_+ = \max\{x, 0\}$.

If the dimension of the multiplet loss is $n = 1$, (3) transforms into the triplet loss.

The thresholds α_j and β_j are the minimal margins between the positive and negative pairs, respectively. The bigger the threshold is, the stronger the item constraint is. If $g_{i,j}^+$ and $g_{i,j}^-$ are harder than $g_{i,j+1}^+$ and $g_{i,j+1}^-$, respectively, the thresholds should be subjected to $\alpha_j \geq \alpha_{j+1}$ and $\beta_j \geq \beta_{j+1}$. We design a gradually decreasing threshold sequence, which is shown in

$$\alpha_j = \frac{1}{j}\alpha, \qquad \beta_j = \frac{1}{j}\beta \tag{4}$$

where α and β are the two basic constants.

The threshold of the quadruplet item should be smaller than that of the corresponding triplet item, because it is a relatively weaker auxiliary constraint. Thus, the constants should meet the requirement of $\alpha \geq \beta$. In our experiments, the distance measurement of $f(p, g)$ is normalized to the interval of $[0, 1]$, and so we set the constants to $\alpha = 1.0$ and $\beta = 0.5$.

Ideally, the positive and negative samples should be mined from the positive and negative ranking lists, respectively, i.e., they are calculated, respectively, by (5) and (6). However, this operation requires the ranking lists to be well initialized, so the distances between the probe sample and each gallery samples should be calculated first, which is time consuming and unacceptable. In Section IV, we will discuss the strategy to build the ranking list progressively.

$$g_{i,j}^{+} = g_{\pi_i^{+}(j)} \tag{5}$$

$$g_{i,j}^{-} = g_{\pi_i^{-}(j)}. \tag{6}$$

C. Backpropagation

Let $I\{x\}$ be the indicator function that takes a value of 1 when x is true, and otherwise takes a value of 0. Then, the partial derivatives are calculated by

$$\begin{aligned}\frac{\partial L_i}{\partial f\left(p_i, g_{i,j}^{+}\right)} &= \sum_{j=1}^{n} I\left\{f\left(p_i, g_{i,j}^{+}\right) - f\left(p_i, g_{i,j}^{-}\right) + \alpha_j > 0\right\}\\ &\quad + \sum_{j=1}^{n-1} I\left\{f\left(p_i, g_{i,j}^{+}\right) - f\left(g_j^{-}, g_{i,j+1}^{-}\right) + \beta_j > 0\right\}\end{aligned} \tag{7}$$

$$\frac{\partial L_i}{\partial f\left(p_i, g_{i,j}^{-}\right)} = -\sum_{j=1}^{n} I\left\{f\left(p_i, g_{i,j}^{+}\right) - f\left(p_i, g_{i,j}^{-}\right) + \alpha_j > 0\right\} \tag{8}$$

$$\frac{\partial L_i}{\partial f\left(g_j^{-}, g_{i,j+1}^{-}\right)} = -\sum_{j=1}^{n-1} I\left\{f\left(p_i, g_{i,j}^{+}\right) - f\left(g_j^{-}, g_{i,j+1}^{-}\right) + \beta_j > 0\right\}. \tag{9}$$

IV. Pipeline of LoopNet

In this section, we introduce the LoopNet pipeline and discuss some details of the designed layers.

A. Pipeline

The pipeline is shown in Fig. 2, and it contains the following steps.

1) The sampling layer simultaneously mines the hard samples from the ranking list and randomly chooses samples from the shuffled samples, and then outputs the image data and person labels.
2) A general network, such as GoogLeNetv3 [42], ResNet50 [43], etc., is used to train the model to represent the implicit pattern characters of the input samples.
3) A fully connected layer with 384 outputs is used to extract the features of the input samples.
4) The features are sent to another fully connected layer (classifier) to map to the person classification space, and then the Softmax loss is used to measure the error between the classification and the person label.
5) The features and person labels are also sent to a pairing layer, in which they are paired to generate the pair distances and pair labels.
6) The pair distances are normalized to the interval [0, 1] by a Softmax layer.
7) The normalized pair distances are ranked to generate the ranking list in the ranking layer.
8) The multiplet loss layer constructs multiplets using the normalized pair distances and measures the error with pair labels.
9) The multiplet loss and Softmax loss are combined to form a final loss with equal weight, which is shown in ($\lambda = 0.5$)

$$L_F = \lambda L_S + (1 - \lambda) L_M \tag{10}$$

where L_F, L_S, and L_M are the final loss, the Softmax loss, and the multiplet loss, respectively.

TABLE I
STRUCTURE OF THE LOOPNET

Layer	Input	Output
Sampling Layer	Ranking List	Image Data, Person Labels
General Network	Image Data	Feature Map
Fully Connected Layer 1	Feature Map	Features
Fully Connected Layer 2	Features	Person Classes
Softmax Loss Layer	Person Classes	Softmax Loss
Pairing Layer	Features, Person Labels	Pair Distances, Pair Labels
Softmax Layer	Pair Distances	Normalized Pair Distances
Ranking Layer	Normalized Pair Distances, Person Labels	Ranking List
Multiplet Loss Layer	Normalized Pair Distances	Multiplet Loss
Final Loss Layer	Softmax Loss Multiplet Loss	Final Loss

One of the outputs of the network is the ranking list, and the ranking list is also the input of the network, which constructs a loop network. In the LoopNet, the sample layer chooses samples from the ranking list. The sample layer is the first layer of the network and does not have parameters to train. In addition, the sample layer only needs the sample orders in the ranking list. Therefore, the sample layer does not need the previous layer to provide residual for calculating the partial derivative. Hence, the cycle is broken between the ranking list and the sample layer in backpropagation.

The structure of the LoopNet is shown in Table I, where the final loss layer is a fictitious layer, which is implemented by the loss weight parameter of the loss layer. The size of the input image is 224 × 112 (height × width).

B. Sampling Layer

As mentioned in Section III-B, the ideally hard sample mining method of (5) and (6) is time consuming for the ranking list initialization. Our goal is to achieve hard sample mining, but avoid the image by image calculation for the ranking list initialization.

To achieve this, we propose a gradual initializing and updating method for the ranking list. Suppose there are m^{+} and m^{-} images in the positive and negative ranking lists, respectively, and the multiplet loss needs n positive and negative samples. At the beginning of training, $m^{+} = 0$ and $m^{-} = 0$ hold. At some iteration of the training, m^{+} and m^{-} can be any value that may be less than the number of positive and negative images in the gallery set. We randomly generate two numbers s^{+} and s^{-} for each mini-batch that denote the numbers of hard positive and negative samples to mine, respectively, and

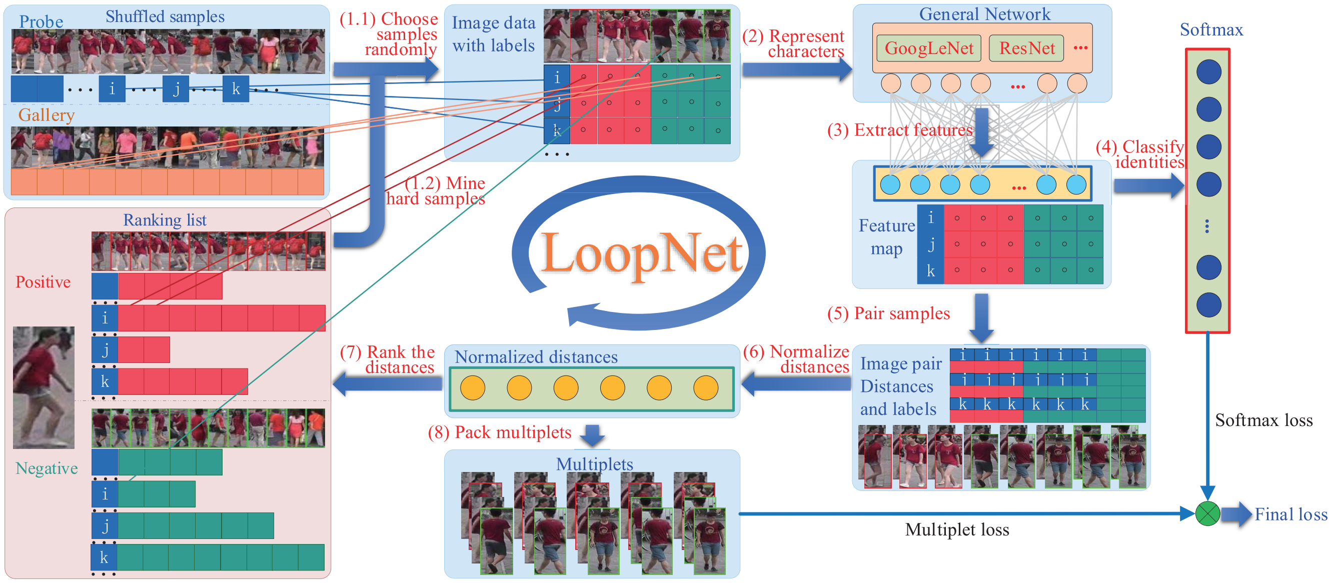


Fig. 2. Pipeline of the LoopNet model. The sample layer chooses an anchor, mines some hard samples, and randomly chooses some samples to compose a mini-batch. A batch contains many mini-batches. After feature extraction, the hard positive and negative samples are paired with their anchor to compute the distances. Then, they are packed into multiplets to calculate the multiplet loss. The features are also classified with a Softmax layer to obtain a Softmax loss. The final loss is the combination of the multiplet loss and the Softmax loss. The blue and orange lines are the positive and negative samples chosen from the shuffled samples, respectively. The red and green lines are the positive and negative samples chosen from the ranking list, respectively.

they are subject to the constraints in (11). Then, we mine s^+ positive and s^- negative samples, and randomly choose the remaining $(n - s^+)$ positive and $(n - s^-)$ negative samples to form the multiplet. The newly selected samples that are not in the ranking list will be added in the list in the ranking layer, which achieves the progressively initializing and updating of the ranking list

$$0 \le s^+ \le \min\{m^+, n\} \qquad 0 \le s^- \le \min\{m^-, n\}. \tag{11}$$

The formal description of the sampling is shown in

$$\begin{aligned} g^+_{i,j} &= \begin{cases} g_{\pi_i^+(j)}, & \text{if } j \le s^+ \\ g_{t_j}, & \text{if } j > s^+ \end{cases} \\ \text{s.t.}\quad & 1 \le t_j \le n_g, \operatorname{id}\left(g_{t_j}\right) = \operatorname{id}(p_i) \\ & \forall t_j \notin \left\{\pi_i^+(1), \dots, \pi_i^+\left(s^+\right), t_{s^++1}, t_{s^++2}, \dots, t_{j-1}\right\} \end{aligned} \tag{12}$$

where t_j is a random number, p_i is the ith probe image, and n_g is the number of gallery images, and in

$$\begin{aligned} g^-_{i,j} &= \begin{cases} g_{\pi_i^-(j)}, & \text{if } j \le s^- \\ g_{t_j}, & \text{if } j > s^- \end{cases} \\ \text{s.t.}\quad & 1 \le t_j \le n_g, \operatorname{id}\left(g_{t_j}\right) \ne \operatorname{id}(p_i) \\ & \forall t_j \notin \left\{\pi_i^-(1), \dots, \pi_i^-\left(s^-\right), t_{s^-+1}, t_{s^-+2}, \dots, t_{j-1}\right\} \\ \operatorname{id}&\left(g^-_{i,j}\right) \ne \operatorname{id}\left(g^-_{i,k}\right)(k = 1, 2, \dots, j-1) \end{aligned} \tag{13}$$

where t_j is a random number and p_i is the ith probe image.

It is important to note that the number of positive gallery images n_g is likely to be less than the dimension n in some data sets. In this case, we simply repeat the "hardest" sample n–n_g times to meet the requirement, which is shown in

$$\begin{aligned} g^+_{i,j} &= \begin{cases} g_{\pi_i^+(1)}, & \text{if } j \le n - n_g, m^+ > 0 \\ g_{\pi_i^+(j-n+n_g+1)}, & \text{if } n - n_g < j \le m^+ + n - n_g \\ g_{t_1}, & \text{if } j \le n - n_g, m^+ = 0 \\ g_{t_{j-n+n_g}}, & \text{if } m^+ + n - n_g < j \le n \end{cases} \\ \text{s.t.}\quad & 1 \le t_j \le n_g, \operatorname{id}\left(g_{t_j}\right) = \operatorname{id}(p_i) \\ & \forall t_j \notin \left\{\pi_i^+(1), \dots, \pi_i^+\left(s^+\right), t_{s^++1}, t_{s^++2}, \dots, t_{j-1}\right\} \end{aligned} \tag{14}$$

where t_j is a random number and p_i is the ith probe image.

C. Pair Images

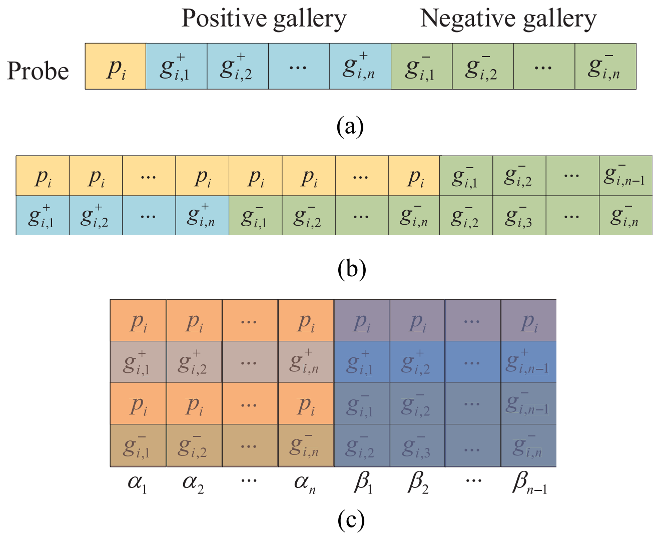
The sampling layer organizes the samples in a mini-batch, as shown in Fig. 3(a), where a probe image is followed by n positive samples and is then followed by n negative samples. Before the multiplet loss calculation, a pair layer is designed to construct the image pairs and compute their distances, including $f(p_i, g^+_{i,j}), f(p_i, g^-_{i,j})$, and $f(g^-_{i,j}, g^-_{i,j+1})$ that are used in (3), which is shown in Fig. 3(b). The Euclidean distance is used to calculate the function $f(\bullet)$.

Suppose the pair layer is the lth layer, and the error terms of the next layer are $\delta^{(l+1)} = <\delta_1^{(l+1)}, \delta_2^{(l+1)}, \dots, \delta_{3n-1}^{(l+1)}>$. Then, the error terms of this layer are calculated by

$$\delta^{(l)}_{p_i} = \sum_{j=1}^{n} \delta_k^{(l+1)} \frac{\partial f\left(p_i, g^+_{i,j}\right)}{\partial p_i} + \sum_{j=1}^{n} \delta_{n+k}^{(l+1)} \frac{\partial f\left(p_i, g^-_{i,j}\right)}{\partial p_i} \tag{15}$$

where $\delta^{(l)}_{p_i}$ denotes the error term of p_i of the lth layer

$$\delta^{(l)}_{g^+_{i,j}} = \delta_k^{(l+1)} \frac{\partial f\left(p_i, g^+_{i,j}\right)}{\partial g^+_{i,j}} \tag{16}$$



(a)

(b)

(c)

Fig. 3. Pair images. (a) Mini-batch samples. An anchor is followed by n positive samples and n negative samples, which form the mini-batch. (b) Image pairs. Each sample is paired with the anchor, and each negative sample is paired with its previous negative sample. (c) Multiplet loss triplets and quadruplets. The columns with red and blue masks are the triplets and quadruplets, respectively.

where $\delta^{(l)}_{g^+_{i,j}}$ denotes the error term of $g^+_{i,j}$ of the lth layer

$$\delta^{(l)}_{g^-_{i,j}} = \delta^{(l+1)}_{n+k} \frac{\partial f\left(p_i, g^-_{i,j}\right)}{\partial g^-_{i,j}} + \delta^{(l+1)}_{2n+k} \frac{\partial f\left(g^-_{i,j}, g^-_{i,j+1}\right)}{\partial g^-_{i,j}} I\{j < n\}$$
$$+ \delta^{(l+1)}_{2n+k-1} \frac{\partial f\left(g^-_{i,j-1}, g^-_{i,j}\right)}{\partial g^-_{i,j}} I\{j > 1\} \quad (17)$$

where $\delta^{(l)}_{g^-_{i,j}}$ denotes the error term of $g^-_{i,j}$ of the lth layer.

The pairs are combined into triplets and quadruplets in the multiplet loss layer, which are shown in Fig. 3(c) with red and blue masks, respectively. The symbols under each column are the thresholds in (3) and (4).

D. Rank Samples

As mentioned in Section IV-B, we mine s^+ positive and s^- negative hard samples, and randomly choose $(n - s^+)$ positive and $(n - s^-)$ negative nonhard samples in a mini-batch. We gradually initialize the ranking list and update it iteration by iteration in the ranking layer.

At the beginning of training, the positive and negative ranking lists are all empty. After an iteration, the sample that is in the ranking list is updated by the newly computed distance, and the sample that is not in the ranking list is appended to the list. Then, the positive and negative ranking lists are sorted by a Bitonic sorting method [44] in a descending and ascending orders, respectively.

In this strategy, the randomly chosen samples ensure that the samples that are not in the ranking list can be selected in some iteration. Actually, the positive samples are quickly appended to the list whereas the negative ones are slowly appended, because the number of negatives is much higher than the positives.

To save memory, we limit the length of the negative ranking list. After the Bitonic sorting, the samples that exceed the length limit are deleted from the list. Therefore, some samples in the ranking list may be replaced by the harder samples in an iteration.

V. Experiments

A. Data Sets

In this section, we evaluate the proposed method on three different data sets. Table II shows the camera, identity (ID), and image numbers.

Market-1501 [45] contains 32 688 bounding boxes of 1501 identities (IDs) produced by the deformable part model (DPM). Each person is captured by 2–6 nonoverlap cameras. This data set contains 2798 distractors (produced by DPM false detection) and 3819 junk images (has zero influence on the ReID accuracy) in the test set.

CUHK03 [20] is one of the largest person ReID data sets, and it has 1467 IDs from five different pairs of cameras on campus. The detected and manual labeled bounding boxes are all used for training in our experiments and have an average of 4.8 detected and manual labeled bounding boxes in each view.

Duke [46] is a subset of the DukeMTMC [47] for image-based ReID. The original data set contains 85-min high-resolution videos from eight different cameras. There are 1404 IDs appearing in more than two cameras and 408 IDs that appear in only one camera. In the training set, 702 IDs are randomly selected, and the remaining 702 IDs are used as the testing set. In the testing set, one query image for each ID in each camera is picked in the probe and the remaining images are set in the gallery. As a result, the data set contains 16 522 training images of 702 IDs, 2228 query images of the other 702 IDs, and 17 661 gallery images.

B. Settings and Evaluation Protocols

We employ the popular networks, ResNet50 [43] and GoogLeNetv3 [42] as the baselines, which are called Res and Inc (Inception) for short, respectively. The architecture of our network is introduced in Section IV-A. The models are implemented on CAFFE [48]. We train the networks 120 000 iterations on each data set.

We use a three-symbol sequence to denote the settings of the experiment. The first symbol shows the sample mining range, which is shown as follows.

1) *L:* Local mining in a mini-batch.
2) *G:* Global mining in the ranking list.

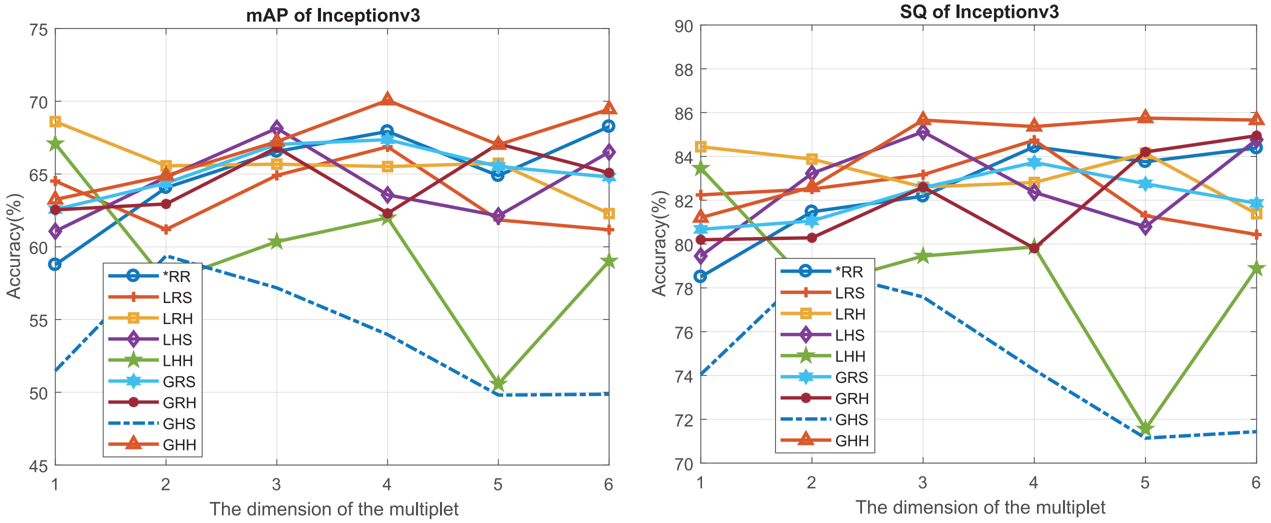
The second and third symbols indicate the positive and negative sample mining modes, respectively, which are shown as follows.

1) *R:* Randomly choose samples.
2) *S:* Choose the semihard samples.
3) *H:* Choose the top hardest samples.

For example, LRS means that it randomly chooses the positive samples and then chooses the corresponding semihard negative samples in a mini-batch. If all positive samples are reserved in a mini-batch, the LRS is equivalent to the method

TABLE II
DATA SET GROUPING FOR TRAINING, VALIDATION, AND TESTING

Dataset	#cameras	#identities				#images		
		total	train & val	test probe	test gallery	train & val	test probe	test gallery
Market-1501	6	1,501+2	751	750	750+2	12,936	3,368	19,732
CUHK03	10	1,467	1,367	100	100	26,253	952	988
Duke	8	1404 + 408	702	702	702 + 408	16,522	2,228	17,661
~~Total~~	–	~~4,782~~	~~2,820~~	~~1,552~~	~~1,962~~	~~55,711~~	~~6,548~~	~~38,381~~



(a) (b)

Fig. 4. Compare with the Inception-v3 baseline on the Market-1501 data set. (a) mAP. (b) SQ.

in [28]. Another example is GHS, which involves the global mining of the top hardest positive samples and semihard negative samples in the ranking list. A special example is *RR, which randomly chooses the positive and negative samples. The first symbol is "*," which indicates that random selection is no different from local mining in a mini-batch and global mining in the ranking list.

In the experiments, we employ the commonly used single query (SQ) accuracy, and the mean average precision (mAP) to evaluate the methods. In the experimental data sets, each ID has multiple instances. For SQ, only the first match is counted regardless of how many ground-truth matches are in the gallery [29], [45], [49]–[51] (from the query viewpoint, we do not know two images belong to one person). The mAP is the mean of the average precision (AP), which provides a more comprehensive evaluation when multiple gallery ground truths exist, because it considers both the precision and recall of an algorithm [45]. The SQ and mAP are all appropriate to evaluate performance in the data sets with several images for each ID.

C. Compare With the Baseline

Figs. 4 and 5 compare the results of different sample mining ranges and modes (detailed in Section V-B) for Inception-v3 and ResNet50, respectively. When the dimension is $n = 1$, the multiplet is equivalent to the triplet. The accuracies of the triplet and the best multiplet are shown in Tables III and IV.

Triplet of Inception-v3:

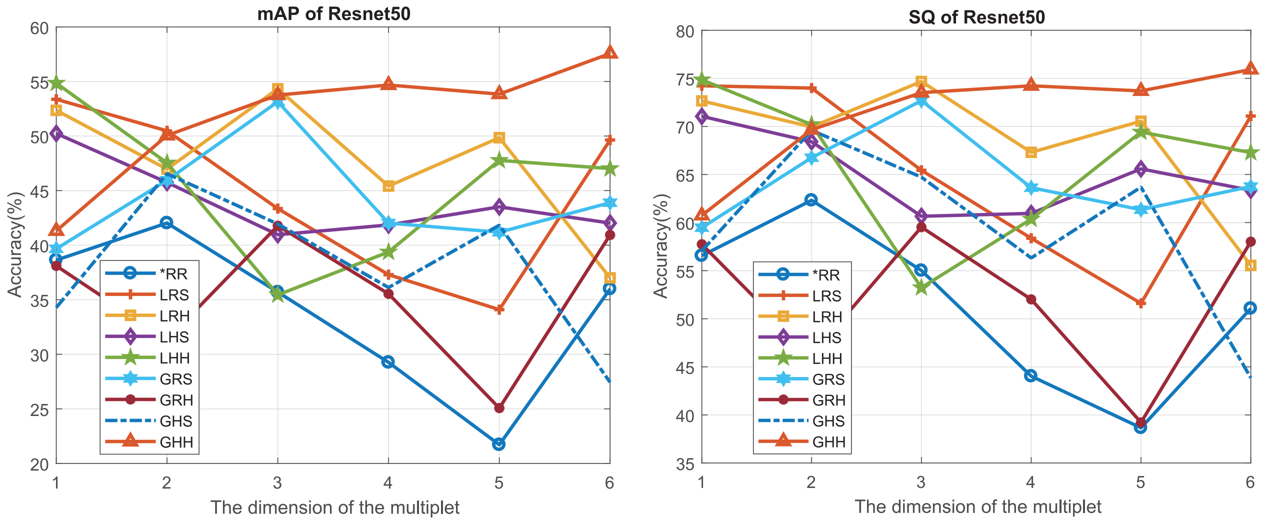
1) The *RR method underperforms the sample mining methods except GHS.
2) The LRH method outperforms all the other methods.

TABLE III
COMPARE WITH THE INCEPTION-V3 BASELINE(%)

Mode	mAP		SQ	
	Triplet	Best Multiplet	Triplet	Best Multiplet
*RR	58.76	68.26	78.50	84.44
LRS	64.52	66.87	82.24	84.74
LRH	**68.60**	68.60	**84.44**	84.44
LHS	61.07	68.14	79.45	85.12
LHH	67.09	67.09	83.46	83.46
GRS	62.56	67.36	80.67	83.73
GRH	62.54	67.07	80.20	84.95
GHS	51.48	59.38	74.05	78.89
GHH	63.25	**70.06**	81.21	**85.75**

TABLE IV
COMPARE WITH THE RESNET50 BASELINE(%)

Mode	mAP		SQ	
	Triplet	Best Multiplet	Triplet	Best Multiplet
*RR	38.63	42.03	56.56	62.32
LRS	53.37	53.37	74.23	74.23
LRH	52.35	54.33	72.65	74.67
LHS	50.23	50.23	71.05	71.05
LHH	**54.85**	54.85	**74.85**	74.85
GRS	39.66	53.14	59.47	72.68
GRH	38.10	41.78	57.75	59.53
GHS	34.28	46.55	57.07	69.60
GHH	41.31	**57.56**	60.75	**75.92**



(a) (b)

Fig. 5. Compare with the ResNet50 baseline on the Market-1501 data set. (a) mAP. (b) SQ.

TABLE V
COMPARE WITH THE STATE OF THE ART

Method	Ref.	Market-1501				CUHK03				Duke			
			SQ		mAP		SQ		mAP		SQ		mAP
		Rank-1	Rank-5	Rank-10		Rank-1	Rank-5	Rank-10		Rank-1	Rank-5	Rank-10	
LOMO+XQDA[52]	CVPR15	43.80	22.20	–	–	52.00	–	–	–	30.80	–	–	17.00
APR[51]	arXiv17	84.29	93.20	95.19	64.67	–	–	–	–	70.69	–	–	51.88
MSCAN[53]	CVPR17	80.31	–	–	57.53	67.99	91.04	95.36	–	–	–	–	–
Re-ranking[37]	CVPR17	77.11	–	–	63.63	61.60	–	–	67.60	–	–	–	–
CSBT[54]	CVPR17	42.9	–	–	20.3	55.5	84.3	–	–	–	–	–	–
P2S [32]	CVPR17	70.72	–	–	44.27	–	–	–	–	–	–	–	–
Spindle[55]	CVPR17	76.90	91.50	94.60	–	88.50	**97.80**	**98.60**	–	–	–	–	–
OMI[56]	CVPR17	82.10	–	–	–	77.70	–	–	–	68.10	–	–	–
ACRN[57]	CVPR17	83.61	92.61	95.34	62.60	62.63	89.69	94.72	70.20	72.58	84.79	88.87	51.96
LSRO[46]	ICCV17	78.06	–	–	56.23	73.10	92.70	96.70	77.40	67.70	–	–	47.10
SPGAN+LMP[58]	CVPR18	58.10	76.00	82.70	26.90	–	–	–	–	46.90	62.60	68.50	26.40
MCAM[59]	CVPR18	83.55	–	–	74.25	49.29	–	–	49.89	–	–	–	–
TJ-AIDL[60]	CVPR18	58.20	74.80	81.10	26.50	–	–	–	–	44.30	59.60	65.00	23.00
Our GHH		**85.36**	**94.30**	**96.23**	**70.06**	**89.08**	94.85	96.74	**86.59**	**75.63**	**87.30**	**89.99**	**57.90**

3) Most local mining methods outperform global mining methods.

Triplet of ResNet50:

1) The *RR method underperforms the sample mining methods except GRH and GHS.
2) The LHH method outperforms all the other methods.
3) All local mining methods outperform global mining methods.

Triplet Summary:

1) The random selection method underperforms most of the sample mining methods, which means that sample mining is effective for the triplet.
2) These results show that mining the hardest negative samples is the most effective method in the ReID problem, and not the semihard negative samples in face recognition [28].
3) For the triplet, the local mining method outperforms global mining methods.

Multiplet of Inception-v3:

1) The *RR method underperforms the sample mining methods except GHS.
2) The GHH method outperforms all the other methods.

Multiplet of ResNet50:

1) The *RR method underperforms the sample mining methods except GRH.
2) The GHH method outperforms all the other methods.

Multiplet Summary: For the multiplet, the performance of the global hard sample mining exceeds that of the local mining method since the multiplet loss considers the objective global importance of samples. The harder the sample is, the bigger the effect will be. The multiplet loss and global hard sample mining work together to improve performance further.

Time Consumption: The local mining time consumptions of ResNet50 and GoogLeNetv3 are 14 and 18 h (120 000 iterations), respectively, whereas the global mining consumptions are 14.4 and 18.4 h, respectively. The global mining of ResNet50 and GoogLeNetv3 expend 2.9% and 2.2% more than the local mining, respectively. The time consumption difference between the global mining and the local mining is little.

D. Compare With the State of the Art

We compare the results of the proposed sample mining approaches against 13 other state-of-the-art methods on Market-1501, CUHK03, and Duke data sets using the SQ top 1, 5, and 10 ranks, and the mAP evaluation, which is shown in Table V. All the compared results come from their published papers.

It is evident that our method outperforms all of the compared state-of-the-art methods on Market-1501 and Duke, which further proved the effectiveness of our proposed method. The Spindle [55] method achieves better performance than ours at rank-5 and rank-10 on the CUHK03 data set, but it combines all current data sets together as its training data, which is much larger than ours. Even so, our rank-1 (the most important measurement) performance with respect to the SQ and mAP on CUHK03 is higher than the Spindle method.

VI. Conclusion

This article focused on the hard sample mining and designs a listwise ranking network, named LoopNet. The article proposes a positive and a negative list to mine the hardest or semihard samples globally, which is better than the local mining methods in a randomly constructed mini-batch. It also presents a multiplet loss that can be used to initialize the ranking list progressively, which allows it to avoid calculating the distances between every probe and every gallery samples before training. The multiplet loss also considers the priority of samples for each image, which makes the harder sample more effective.

From this article, we can also draw some useful conclusions for the ReID problem: 1) hard sample mining is effective for promoting the ReID performance; 2) when the triplet loss is used, local sample mining in a mini-batch can get the best performance; and 3) if we consider the priority and effectiveness of samples, the multiplet can further improve the ReID performance. In addition, the global hardest sample mining outperforms all other global sample mining methods and local mining methods.

This is a study on ReID, a popular ranking problem. One of the remaining questions is whether these conclusions can be generalized to other problems, e.g., classification, detection, and generation. Since the applications vary wildly, it becomes an interesting topic to ask whether the hard samples subject to the same distribution or whether the results of the hard mining methods can still keep consistent, which deserves further and comprehensive study.

Acknowledgment

The authors would like to thank the HAWKEYE Group.

References

[1] A. Bhuiyan, A. Perina, and V. Murino, "Exploiting multiple detections for person re-identification," *J. Imag.*, vol. 4, no. 2, p. 28, 2018.

[2] R. R. Varior, B. Shuai, J. Lu, D. Xu, and G. Wang, "A Siamese long short-term memory architecture for human re-identification," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 135–153.

[3] L. Zheng, Y. Yang, and Q. Tian, "Sift meets CNN: A decade survey of instance retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 5, pp. 1224–1244, Apr. 2018.

[4] J. Song, L. Gao, L. Liu, X. Zhu, and N. Sebe, "Quantization-based hashing: A general framework for scalable image and video retrieval," *Pattern Recognit.*, vol. 75, pp. 175–187, Mar. 2018.

[5] X. Alameda-Pineda, E. Ricci, and N. Sebe, "Multimodal behavior analysis in the wild: An introduction," in *Multimodal Behavior Analysis in the Wild*. San Diego, CA, USA: Elsevier, 2019, pp. 1–8.

[6] S. Gong, M. Cristani, C. C. Loy, and T. M. Hospedales, "The re-identification challenge," in *Person Re-Identification*. Berlin, Germany: Springer, 2014, pp. 1–20.

[7] D. Cheng, Y. Gong, S. Zhou, J. Wang, and N. Zheng, "Person re-identification by multi-channel parts-based CNN with improved triplet loss function," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1335–1344.

[8] S.-Z. Chen, C.-C. Guo, and J.-H. Lai, "Deep ranking for person re-identification via joint representation learning," *IEEE Trans. Image Process.*, vol. 25, no. 5, pp. 2353–2367, May 2016.

[9] S. Ding, L. Lin, G. Wang, and H. Chao, "Deep feature learning with relative distance comparison for person re-identification," *Pattern Recognit.*, vol. 48, no. 10, pp. 2993–3003, 2015.

[10] C. Su, S. Zhang, J. Xing, W. Gao, and Q. Tian, "Deep attributes driven multi-camera person re-identification," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 475–491.

[11] F. Wang, W. Zuo, L. Lin, D. Zhang, and L. Zhang, "Joint learning of single-image and cross-image representations for person re-identification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1288–1296.

[12] D. Yu *et al.*, "Implementing abstract MAC layer in dynamic networks," *IEEE Trans. Mobile Comput.*, early access, doi: 10.1109/TMC.2020.2971599.

[13] H. Yang, F. Li, D. Yu, Y. Zou, and J. Yu, "Reliable data storage in heterogeneous wireless sensor networks by jointly optimizing routing and storage node deployment," *Tsinghua Sci. Technol.*, to be published.

[14] T. Xiao, H. Li, W. Ouyang, and X. Wang, "Learning deep feature representations with domain guided dropout for person re-identification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1249–1258.

[15] A. Franco and L. Oliveira, "Convolutional covariance features: Conception, integration and performance in person re-identification," *Pattern Recognit.*, vol. 61, pp. 593–609, Jan. 2017.

[16] Y. Xun, J. Liu, N. Kato, Y. Fang, and Y. Zhang, "Automobile driver fingerprinting: A new machine learning based authentication scheme," *IEEE Trans. Ind. Informat.*, vol. 16, no. 2, pp. 1417–1426, Feb. 2020.

[17] Y. Xun, J. Liu, and Y. Zhang, "Side-channel analysis for intelligent and connected vehicle security: A new perspective," *IEEE Netw.*, early access, doi: 10.1109/MNET.001.1900214.

[18] J. Ning, J. Wang, J. Liu, and N. Kato, "Attacker identification and intrusion detection for in-vehicle networks," *IEEE Commun. Lett.*, vol. 23, no. 11, pp. 1927–1930, Nov. 2019.

[19] P. Li, C. J. C. Burges, and Q. Wu, "McRank: Learning to rank using multiple classification and gradient boosting," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2007, pp. 897–904.

[20] W. Li, R. Zhao, T. Xiao, and X. Wang, "DeepReID: Deep filter pairing neural network for person re-identification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 152–159.

[21] D. Yi, Z. Lei, S. Liao, and S. Z. Li, "Deep metric learning for person re-identification," in *Proc. 22nd Int. Conf. Pattern Recognit. (ICPR)*, 2014, pp. 34–39.

[22] E. Ahmed, M. J. Jones, and T. K. Marks, "An improved deep learning architecture for person re-identification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3908–3916.

[23] X. Li, W.-S. Zheng, X. Wang, T. Xiang, and S. Gong, "Multi-scale learning for low-resolution person re-identification," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 3765–3773.

[24] Q.-S. Hua *et al.*, "Faster parallel core maintenance algorithms in dynamic graphs," *IEEE Trans. Parallel Distrib. Syst.*, vol. 31, no. 6, pp. 1287–1300, Jun. 2020.

[25] D. Yu *et al.*, "Stable local broadcast in multihop wireless networks under SINR," *IEEE/ACM Trans. Netw.*, vol. 26, no. 3, pp. 1278–1291, Jun. 2018.

[26] W. Chen, X. Chen, J. Zhang, and K. Huang, "Beyond triplet loss: A deep quadruplet network for person re-identification," in *Proc. CVPR*, vol. 2, no. 6, 2017, pp. 1320–1329.

[27] J. Wang, Z. Wang, C. Gao, N. Sang, and R. Huang, "DeepList: Learning deep features with adaptive listwise constraint for person reidentification," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 3, pp. 513–524, Mar. 2017.

[28] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 815–823.

[29] A. Hermans, L. Beyer, and B. Leibe, "In defense of the triplet loss for person re-identification," 2017. [Online]. Available: arXiv:1703.07737.

[30] T. Kong, F. Sun, A. Yao, H. Liu, M. Lu, and Y. Chen, "RON: Reverse connection with objectness prior networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5244–5252.

[31] H. Liu, J. Feng, M. Qi, J. Jiang, and S. Yan, "End-to-end comparative attention networks for person re-identification," 2016. [Online]. Available: arXiv:1606.04404.

[32] S. Zhou, J. Wang, J. Wang, Y. Gong, and N. Zheng, "Point to set similarity based deep feature learning for person re-identification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 3741–3750.

[33] J. I. Marden, *Analyzing and Modeling Rank Data, Volume 64 of Monographs on Statistics and Applied Probability*. Boca Raton, FL, USA: CRC Press, 1995.

[34] F. Xia, T.-Y. Liu, J. Wang, H. Li, and H. Li, "Listwise approach to learning to rank: Theory and algorithm," in *Proc. Int. Conf. Mach. Learn.*, 2008, pp. 1192–1199.

[35] X. Wang and A. Gupta, "Unsupervised learning of visual representations using videos," 2015. [Online]. Available: arXiv:1505.00687.

[36] Z. Liu, D. Wang, and H. Lu, "Stepwise metric promotion for unsupervised video person re-identification," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2017, pp. 2448–2457.

[37] Z. Zhong, L. Zheng, D. Cao, and S. Li, "Re-ranking person re-identification with k-reciprocal encoding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 3652–3661.

[38] M. Ye, A. J. Ma, L. Zheng, J. Li, and P. C. Yuen, "Dynamic label graph matching for unsupervised video re-identification," in *Proc. Int. Conf. Comput. Vis.*, 2017, pp. 5152–5160.

[39] J. Zhou, P. Yu, W. Tang, and Y. Wu, "Efficient online local metric adaptation via negative samples for person reidentification," in *Proc. IEEE Conf. Comput. Vis.*, vol. 6, 2017, p. 7.

[40] D. Triantafyllidou, P. Nousi, and A. Tefas, "Fast deep convolutional face detection in the wild exploiting hard sample mining," *Big Data Res.*, vol. 11, pp. 65–76, Mar. 2018.

[41] X. Dong, L. Zheng, F. Ma, Y. Yang, and D. Meng, "Few-example object detection with model communication," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access.

[42] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2818–2826.

[43] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[44] D. Nassimi and S. Sahni, "Bitonic sort on a mesh-connected parallel computer," *IEEE Trans. Comput.*, vol. C-28, no. 1, pp. 2–7, Jan. 1979.

[45] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian, "Scalable person re-identification: A benchmark," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1116–1124.

[46] Z. Zheng, L. Zheng, and Y. Yang, "Unlabeled samples generated by GAN improve the person re-identification baseline *in vitro*," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 3774–3782.

[47] E. Ristani, F. Solera, R. Zou, R. Cucchiara, and C. Tomasi, "Performance measures and a data set for multi-target, multi-camera tracking," in *Proc. Eur. Conf. Comput. Vis. Workshop Benchmarking Multi Target Tracking*, 2016, pp. 17–35.

[48] Y. Jia *et al.*, "CAFFE: Convolutional architecture for fast feature embedding," 2014. [Online]. Available: arXiv:1408.5093.

[49] L. Zheng, Y. Yang, and A. G. Hauptmann, "Person re-identification: Past, present and future," 2016. [Online]. Available: arXiv:1610.02984.

[50] Z. Zheng, L. Zheng, and Y. Yang, "A discriminatively learned CNN embedding for person reidentification," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 14, no. 1, p. 13, 2017.

[51] Y. Lin, L. Zheng, Z. Zheng, Y. Wu, and Y. Yang, "Improving person re-identification by attribute and identity learning," 2017. [Online]. Available: arXiv:1703.07220.

[52] S. Liao, Y. Hu, X. Zhu, and S. Z. Li, "Person re-identification by local maximal occurrence representation and metric learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 2197–2206.

[53] D. Li, X. Chen, Z. Zhang, and K. Huang, "Learning deep context-aware features over body and latent parts for person re-identification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 384–393.

[54] J. Chen, Y. Wang, J. Qin, L. Liu, and L. Shao, "Fast person re-identification via cross-camera semantic binary transformation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5330–5339.

[55] H. Zhao *et al.*, "Spindle net: Person re-identification with human body region guided feature decomposition and fusion," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1077–1085.

[56] T. Xiao, S. Li, B. Wang, L. Lin, and X. Wang, "Joint detection and identification feature learning for person search," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 3376–3385.

[57] A. Schumann and R. Stiefelhagen, "Person re-identification by deep learning attribute-complementary information," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2017, pp. 1435–1443.

[58] W. Deng, L. Zheng, Q. Ye, G. Kang, Y. Yang, and J. Jiao, "Image-image domain adaptation with preserved self-similarity and domain-dissimilarity for person reidentification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, vol. 1, 2018, p. 6.

[59] C. Song, Y. Huang, W. Ouyang, and L. Wang, "Mask-guided contrastive attention model for person re-identification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 1179–1188.

[60] J. Wang, X. Zhu, S. Gong, and W. Li, "Transferable joint attribute-identity deep learning for unsupervised person re-identification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 2275–2284.

Hao Sheng (Member, IEEE) received the B.S. and Ph.D. degrees from the School of Computer Science and Engineering, Beihang University, Beijing, China, in 2003 and 2009, respectively.

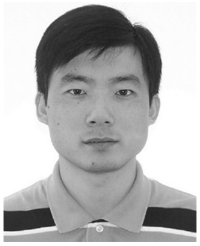
He is currently an Associate Professor with the School of Computer Science and Engineering, Beihang University. He is working on computer vision, pattern recognition, and machine learning.

Yanwei Zheng received the B.S. degree from Shandong Jianzhu University, Jinan, China, in 1999, the M.S. degree from Shandong University, Qingdao, China, in 2004, and the Ph.D. degree from the School of Computer Science and Engineering, Beihang University, Beijing, China.

He joined the University of Jinan, Jinan, where he became a Lector from 2004 to 2013. His research interests include machine learning, computer vision, and especially person reidentification.

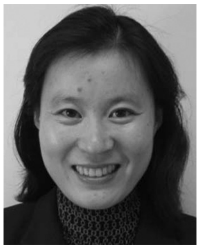
Wei Ke (Member, IEEE) received the Ph.D. degree from the School of Computer Science and Engineering, Beihang University, Beijing, China.

He is an Associate Professor of computing program with Macao Polytechnic Institute, Macao, China. His research interests include programming languages, image processing, computer graphics, and tool support for object-oriented and component-based engineering and systems. His recent research focuses on the design and implementation of open platforms for applications of computer graphics and pattern recognition, including programming tools, environments, and frameworks.



Dongxiao Yu received the B.Sc. degree from the School of Mathematics, Shandong University, Qingdao, China, in 2006, and the Ph.D. degree from the Department of Computer Science, University of Hong Kong, Hong Kong, in 2014.

He became an Associate Professor with the School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China, in 2016. He is currently a Professor with the School of Computer Science and Technology, Shandong University. His research interests include wireless networks, distributed computing, and graph algorithms.



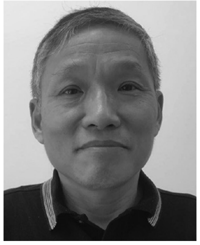
Xiuzhen Cheng (Fellow, IEEE) received the M.S. and Ph.D. degrees in computer science from the University of Minnesota, Minneapolis, MN, USA, in 2000 and 2002, respectively.

She is a Professor with the School of Computer Science and Technology, Shandong University, Qingdao, China. Her current research interests include cyber–physical systems, wireless and mobile computing, sensor networking, wireless and mobile security, and algorithm design and analysis.

Prof. Cheng received the NSF CAREER Award in 2004. She has served on the editorial boards of several technical journals and the technical program committees of various professional conferences/workshops. She also has chaired several international conferences. She worked as a Program Director for the U.S. National Science Foundation (NSF) from April 2006 to October 2006 (full time) and from April 2008 to May 2010 (part time). She is a member of ACM.

Weifeng Lyu received the Ph.D. degree in computer science from Beihang University, Beijing, China.

He is a Professor, the Dean of the School of Computer Science and Engineering, and the Vice Director of the State Key Laboratory of Software Development Environment, Beihang University and the Secretary General of the China Software Industry Association, and the Director of National Engineering Research Center for Science and Technology Resources Sharing Service. His research interests include massive information system and urban cognitive computing.



Zhang Xiong received the B.S. degree from Harbin Engineering University, Harbin, China, in 1982, and the M.S. degree from Beihang University, Beijing, China, in 1985.

He is a Professor and a Ph.D. Supervisor with the School of Computer Science and Engineering, Beihang University. He is working on computer vision, information security, and data vitalization.