Support Neighbor Loss for Person Re-Identification

Kai Li[1], Zhengming Ding[2], Kunpeng Li[1], Yulun Zhang[1] and Yun Fu[1,3]
[1] Department of Electrical and Computer Engineering, Northeastern University, Boston, USA
[2] Department of Computer, Information and Technology, Indiana University-Purdue University
[3] College of Computer & Information Science, Northeastern University, Boston, USA
{kaili,kunpengli,yunfu}@ece.neu.edu,zd2@iu.edu,yulun100@gmail.com

ABSTRACT

Person re-identification (re-ID) has recently been tremendously boosted due to the advancement of deep convolutional neural networks (CNN). The majority of deep re-ID methods focus on designing new CNN architectures, while less attention is paid on investigating the loss functions. Verification loss and identification loss are two types of losses widely used to train various deep re-ID models, both of which however have limitations. Verification loss guides the networks to generate feature embeddings of which the intra-class variance is decreased while the inter-class ones is enlarged. However, training networks with verification loss tends to be of slow convergence and unstable performance when the number of training samples is large. On the other hand, identification loss has good separating and scalable property. But its neglect to explicitly reduce the intra-class variance limits its performance on re-ID, because the same person may have significant appearance disparity across different camera views. To avoid the limitations of the two types of losses, we propose a new loss, called support neighbor (SN) loss. Rather than being derived from data sample pairs or triplets, SN loss is calculated based on the positive and negative support neighbor sets of each anchor sample, which contain more valuable contextual information and neighborhood structure that are beneficial for more stable performance. To ensure scalability and separability, a softmax-like function is formulated to push apart the positive and negative support sets. To reduce intra-class variance, the distance between the anchor's nearest positive neighbor and furthest positive sample is penalized. Integrating SN loss on top of Resnet50, superior re-ID results to the state-of-the-art ones are obtained on several widely used datasets.

CCS CONCEPTS

• **Computing methodologies** → **Computer vision**; **Image representations**; **Object identification**; **Machine learning**; **Learning to rank**;

KEYWORDS

Person re-identification; loss function; deep neural networks

MM '18, October 22–26, 2018, Seoul, Republic of Korea

ACM ISBN 978-1-4503-5665-7/18/10…$15.00
https://doi.org/10.1145/3240508.3240674

ACM Reference Format: Kai Li, Zhengming Ding, Kunpeng Li, Yulun Zhang, and Yun Fu. 2018. Support Neighbor Loss for Person Re-Identification. In 2018 ACM Multimedia Conference (MM' 18), October 22-26, 2018, Seoul, Republic of Korea. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3240508.3240674

1 INTRODUCTION

Person re-identification (re-ID) has been attached with increasing attentions in recent years due to its importance for many real-world applications, such as video surveillance, robotics, human-computer interaction, etc. Given an image of the person of interest, the goal of re-ID is to find the other images of the same person captured by different cameras or the same cameras in different time. By the nature of this task, two fundamental problems need to be solved. The first is to generate representations that encode the most discriminative appearance cues of a person, such that his/her representations from different camera views are similar. The second is to develop appropriate distance metrics under which images of the same person are more similar than those of different persons in terms of the representation distance.

Conventional methods consider the two problems separately. Some focus on developing robust pedestrian image descriptor [15, 24, 38, 47], while others strive to develop effective distance metrics [20, 26, 27, 40]. Benefiting from the power of the deep convolutional neural network (CNN) in generating discriminative representations, as well as the availability of large-scale annotated datasets, the re-ID community has witnessed a rapid blossom in the most recent several years [2, 14, 31, 46, 53]. Unlike the conventional algorithms, deep learning based re-ID methods learn from data feature representations and distance metrics jointly in an end-to-end manner.

The majority of deep re-ID methods focus on designing new CNN architectures to model the pose variance, human body misalignment, occlusion, etc [1, 6, 18, 34, 52]. However, much less effort has been paid on the investigation of the loss. Two types of losses are extensively utilized for re-ID, i.e., verification loss and identification loss. Verification loss strives to reduce the intra-class variance while enlarge the inter-class one. This property makes verification loss a seemingly natural choice for re-ID, since the same person can have substantial appearance disparity owing to the viewpoint, pose and background variations. Deep neural networks which model the large intra-class variations shall be promising to output good verification results during the test time. Despite of the success in a number of re-ID algorithms [7, 34, 39, 52], verification loss based methods shall suffer the common problem that they are prone to show slow convergence and unstable performance in the circumstance of numerous person identities. Some well designed hard mining strategy can somewhat mitigate this problem [10, 30]. But it is ineradicable since analyzing individual pairs or triplets

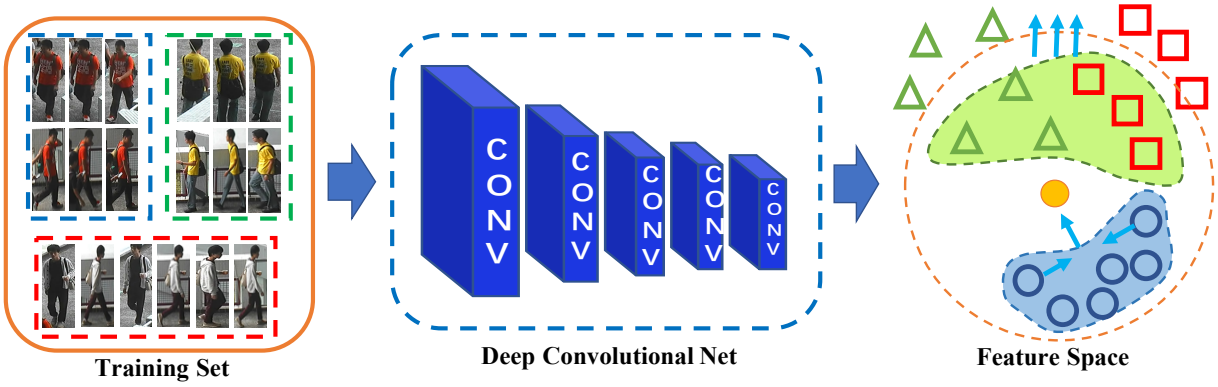


Figure 1: Framework of the proposed method. A batch of images are fed to a deep neural network to get the feature embeddings. Taking each sample as the anchor in the embedding space, we calculate its K-NN neighbors and pull the positive neighbors towards it while push the negative ones away from it. Meanwhile, the positive neighbors are squeezed together to form a compact cluster.

of a example does not employ sufficient contextual insight of the neighborhood structure, thereby leading the model fails to learn the normal associations of the person.

On the other hand, identification loss has good separation and scalable property, and it is extensively used in large-scale classification problem [9]. When applying identification loss for re-ID, unlike verification loss based methods where the similarity metric is directly encoded as the training objective, a classifier is trained on the training identities and nearest neighbor query is performed at test time using the feature representations of the trained network [39, 51]. Identification loss has good property of separating different classes, but it does not explicitly encode the intra-class variance, thus shall cause unsatisfactory results for re-ID where the intra-person variance could be significant.

Some methods [6, 18, 52] attempt to combine these two types of losses in multi-task fashion. Typically, the networks of these methods often consist of multiple branches, with each branch corresponding to one type of loss. How to balance different branches remains a challenging problem, and the multi-branch structures definitively make the models hard to train and less efficient for inference.

Instead, we propose a new loss, called support neighbor (SN) loss, which avoids the problems of both identification and verification losses. Taking each sample within a batch as the anchor, we calculate its K-NN neighbors and define the positive samples and negative samples among the K-NN neighbors as **support neighbors**. SN loss is calculated based on the support neighbors. In order to encourage intra-class cohesion, we penalize the difference of the distances of the furthest positive neighbor and the nearest positive neighbor to the anchor. To ensure class separation, we formulate a softmax-like function to maximize the similarity of positive neighbors to the anchor and meanwhile minimize the similarity of negative neighbors to the anchor.

The support neighbors of a sample are the most similar samples of the anchor within the batch, thereby containing the most important information of the person the anchor image captures. Unlike triplet, which contains very limited contextual information and neighborhood structure, our SN loss is exposed with more such valuable information, so deep models learned based on our SN loss can encode more discriminative information about persons. We verify the effectiveness of the proposed SN loss by integrating it on top of the common feature extraction network Resnet50 [9]. The experiments show we achieve remarkable improvements over the state-of-the-art methods on several commonly benchmarking datasets. Figure 1 shows the framework of the proposed method.

2 RELATED WORKS

Person re-ID is a very hot topic in recent years and numerous algorithms have been proposed. Here we give a brief overview of methods in this area, with an inclination of deep learning based methods. Zheng *et al* had conducted a comprehensive survey; we refer interested readers to their paper [50].

Conventional re-ID methods focus either on pedestrian image description or distance metric learning. Feature generation based methods underlie on the fact that images of the same person in different camera views should be similar in appearance. Since a single image descriptor (*i.e.*, RGB, LBP, SIFT, etc.) is often not powerful enough to encode all the information that are essential for pedestrian image matching, concatenating the feature vectors of several image descriptor is commonly used [3, 19, 25, 28, 48]. Besides directly using low-level color and texture features, some methods also strive to utilize pedestrian attributes which are more robust to image transformations [13, 23, 32].

Due to the high-dimensional nature of pedestrian image features, it is critical to learn a good distance metric to obtain the invariant factors among sample variances. The general idea of metric learning based re-ID methods is to learn some distance metrics under which the vectors of the same identities are pushed closer while the vectors of different identities are pulled further apart. The most acknowledged metric learning based person re-ID algorithm is KISSME [12], which decides whether a pair of description vectors is similar or not by formulating it as a likelihood ratio test under the assumption that the feature distances obey a Gaussian distribution with a zero mean. Inspired by KISSME, many metric learning based person re-ID algorithms have been proposed, including LFDA [40], XQDA [20], MLAPG [21].

Deep learning based methods have tremendously pushed forward the boundary of re-ID. These methods focus on designing various deep CNN structures to learn discriminative feature embeddings and/or strive to devise better loss functions for training the networks. To model the pose variations and viewpoint changes in cross-view pedestrian images, a number of methods introduce some special units into the architectures. Ahmed *et al.* [1] proposed to capture local relationship between two images via a cross-input neighborhood difference layer and designed a patch summary layer to summarize the features obtained in the previous layers. Varior *et al.* [34] proposed to use a gating function to compare features along a horizontal stripe and output a gating mask to indicate importance of the local patterns. To solve the misalignment of human bodies in different images, some methods design multi-branch model architectures, with each branch corresponding to a body part or the whole bodies, and fuse all branches together as the final feature embedding [6, 14]. Some methods introduce attention modules to emphasize the most discriminative parts of human body. Liu *et.al* [22] built a recurrent Siamese neural network which generated attention maps according to the comparison over triplets of person images. Zhao *et al.* [46] designed a network which jointly modeled representation computation and body part extraction in an end-to-end fashion by maximizing the re-identification quality.

Compared with designing various deep CNN architectures, less attentions have been attached on the loss function. The majority of the existing methods employ existing verification loss to train their models, including triplet loss [7, 36] contrastive loss [34, 35], binary identification loss [1, 17]. Recently, some seminal investigation has been made on improving the loss functions. Some approaches [39, 51] attempt to use identification loss to learn discriminative features efficiently. The combination of both verification and identification losses has also been initialized in [6, 18, 52]. Hermans *et al.* [10] recently proposed a new formulation of calculating triplet loss and mitigated the limitation of conventional triplet loss calculation mechanism. Chen *et al.* [5] proposed the quadruplet loss which added one more negative sample into the triplet and reached better generalization ability.

Our method also focuses on the loss function, but unlike existing methods which investigate pairs, triplets, or quadruplets of anchor samples, we investigate the neighbors of anchors, which require minimal data sampling and involve more contextual information, thereby securing stable and superior performance.

3 ALGORITHM

This section will introduce the proposed Support Neighbor (SN) loss in details. Before elaborating it, we will first give a brief introduction of softmax loss, the most popular identification loss; and triplet loss, the most widely used verification loss. We will analyze their limitations and show how they motivate us to develop the proposed SN loss.

Given a training dataset $\mathcal{S} = \{(\mathbf{I}_i, y_i)\}_{i=1}^N$ of N pedestrian images of M identities, where $\mathbf{I}_i$ is the i-th pedestrian image, and y_i is its corresponding identity label. The goal of (supervised) deep learning based re-ID methods is to learn from $\mathcal{S}$ a feature embedding model Ω_θ, parameterized by θ. With Ω_θ, the feature embeddings of any query image $\mathbf{I}_q$ and the gallery images $\mathcal{G} = \{\mathbf{I}_g^j\}_{j=1}^G$ can be obtained as $\mathbf{x}_q = \Omega_\theta(\mathbf{I}_q)$ and $\mathcal{G}' = \{\mathbf{x}_g^j\}_{j=1}^G$, where $\mathbf{x}_g^j = \Omega_\theta(\mathbf{I}_g^j)$. Then, the similarity of $\mathbf{x}_q$ and every $\mathbf{x}_g^j \in \mathcal{G}'$ is calculated, and the most similar one(s) are return as the re-ID results. To learn Ω_θ, some loss function $L(\theta) = f(\{\mathbf{x}_i, y_i\}_{i=1}^N)$ is to be minimized, where $\mathbf{x}_i = \Omega_\theta(\mathbf{I}_i)$.

3.1 Existing Losses

Softmax loss. Softmax loss is widely used for multi-class classification task in which the goal is to enlarge the inter-class difference:

$$L_{stm}(\theta) = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{\exp(\mathbf{W}_{y_i}^{\mathrm{T}}x_i + b_{y_i})}{\sum_{j=1}^{M}\exp(\mathbf{W}_j^{\mathrm{T}}x_i + b_j)}, \tag{1}$$

where $\mathbf{W}_j$ and $\mathbf{b}_j$ are j-th column of the weight and bias matrices. Minimizing Eq. (1) with training dataset $\mathcal{S}$, Ω_θ can be obtained. With Ω_θ, the embeddings of the query images and gallery images can be obtained. Then, the nearest neighbor(s) of $\mathbf{x}_q$ among $\mathcal{G}' = \{\mathbf{x}_g^j\}_{j=1}^G$ are regarded as the retrieval results.

One may have noticed that softmax loss only aims to ensure separation of different classes, but neglects to decrease the intra-class variance, which however is crucial for re-ID because the same person could be captured of great appearance disparity in different camera views. Thus, simply using softmax loss for re-ID may not lead to satisfactory results.

Triplet loss. Verification loss seems more suitable for re-ID because it explicitly penalizes intra-class variance, while encourages the inter-class one. Taking the famous triplet loss as an example, its objective function is

$$L_{tri}(\theta) = \sum_{i=1}^{N}\left[\|\mathbf{x}_i - \mathbf{x}_p\|_2^2 - \|\mathbf{x}_i - \mathbf{x}_n\|_2^2 + \alpha\right]_+, \tag{2}$$

where α is the distance margin, $\mathbf{x}_i$, $\mathbf{x}_p$, and $\mathbf{x}_n$ are the anchor, positive sample and negative sample, respectively.

The foremost limitation of triplet loss is that the number of triplets is cubical to the number of samples, which shall significantly slow down model convergence and lead to unstable performance. Some hard mining strategies [8, 29] can somewhat mitigate this problem. However, regardless of which type of mining is being done, it is a separate step from training and adds considerable overhead, as it requires embedding a large fraction of the data with the most recent model and computing all pairwise distances between those data points.

Recently, Hermans *et al.* [10] proposed a novel batch sampling strategy, which avoids performing triplet construction on the whole dataset, but on mini-batches, thereby significantly improving the training efficiency and performance stableness. The main idea is to randomly select a constant number of person with a constant number of images for each person in each batch. Within each batch, two variants of triplet losses are formulated. The first one is the Batch Hard (BH) variant, which selects the hardest triplets within the batch; the second one is the Batch All (BA) variant, which permutes all triplets within the batch. Though being proved effective, both the two variants have limitations. For the BH variant, it selects only one triplet for an anchor sample, risking that the typical patterns of the identity fail to be encoded. On the other hand, the BA variant permutes all possible positive-negative sample combinations, and would hence still suffer from efficiency problem.

3.2 Support Neighbor Loss

We follow the batch sampling strategy of [10] by randomly selecting P person identities, with Q image for each person in a batch. Therefore, in each mini-batch $\mathcal{M}$, we have $M = P*Q$ images. Taking each $\mathbf{x}_i \in \mathcal{M}$ as the anchor, instead of constructing pairs or triplets, we find its K-NN neighbors $\mathcal{K}_i = \mathcal{P}_i \cup \mathcal{N}_i$, which consists of positive samples $\mathcal{P}_i$ and negative samples $\mathcal{N}_i$. It is worth to be noted that samples within $\mathcal{K}_i$ are most similar to x_i, thus containing the most informative patterns specific to the identity $\mathbf{x}_i$ represents. Samples not included in $\mathcal{K}_i$ are less informative and are simply discarded for the sake of efficiency and in case that they dilute the contribution of the highly informative ones. With respect to $\mathbf{x}_i$, $\mathcal{P}_i$ and $\mathcal{N}_i$ are the most similar samples with each other, but with different labels, and we need to separate them. In this sense, they are analogous to the "support vectors" in SVM. Following this nomenclature, we call them *support neighbors*.

In some sense, we also construct a "triplet" $\{\mathbf{x}_i, \mathcal{P}_i, \mathcal{N}_i\}$ for $\mathbf{x}_i$. But unlike the conventional triplet, in our case, $\mathcal{P}_i$ and $\mathcal{N}_i$ are sample groups, rather than individual samples. These positive and negative neighbors provide more valuable contextual information and neighborhood structure that are absent from the conventional triplet. With the expanded hard positive and negative samples, more representative and discriminative information can be encoded into the deep model. Besides, for each anchor, we have only one unique "triplet" within a batch, so that stable performance is more likely to be guaranteed.

Based on the support neighbors, we develop the SN loss which aims to separate the positive neighbors from the negative ones, and meanwhile penalize the variance among the positive neighbors. Figure 2 shows the schematic illustration of the proposed loss.

Separation loss. To separate the positive and negative support neighbors, we seek to maximize the similarity between the anchor $\mathbf{x}_i$ and samples from $\mathcal{P}_i$, and meanwhile minimize the similarity between $\mathbf{x}_i$ and samples from $\mathcal{N}_i$. By this consideration, we define the separation loss as:

$$L_{spr}(\theta) = -\sum_{i=1}^{N} \log \frac{\sum_{\mathbf{x}_p \in \mathcal{P}_i} \exp\left(-\sigma D(\mathbf{x}_i, \mathbf{x}_p)\right)}{\sum_{\mathbf{x}_p \in \mathcal{P}_i} \exp\left(-\sigma D(\mathbf{x}_i, \mathbf{x}_p)\right) + \sum_{\mathbf{x}_n \in \mathcal{N}_i} \exp\left(-\sigma D(\mathbf{x}_i, \mathbf{x}_n)\right)}, \tag{3}$$

where σ is a scaling factor and $D(\cdot,\cdot)$ is the euclidean distance between two samples. We can observe that $S_{ip} = \sum_{\mathbf{x}_p \in \mathcal{P}_i} \exp\left(-\sigma D(\mathbf{x}_i, \mathbf{x}_p)\right)$ measures the similarity of $\mathbf{x}_i$ to all positive neighbors from $\mathcal{P}_i$, while $S_{in} = \sum_{\mathbf{x}_n \in \mathcal{N}_i} \exp\left(-\sigma D(\mathbf{x}_i, \mathbf{x}_n)\right)$ measures the similarity of $\mathbf{x}_i$ to all negative neighbors from $\mathcal{N}_i$. Since $\mathcal{P}_i \cup \mathcal{N}_i = \mathcal{K}_i$, we can rewrite Eq.(3) as

$$L_{spr}(\theta) = -\sum_{i=1}^{N} \log \frac{\sum_{\mathbf{x}_p \in \mathcal{P}_i} \exp\left(-\sigma D(\mathbf{x}_i, \mathbf{x}_p)\right)}{\sum_{\mathbf{x}_s \in \mathcal{K}_i} \exp\left(-\sigma D(\mathbf{x}_i, \mathbf{x}_s)\right)} \tag{4}$$

One may have noticed that our separation loss formulation in Eq. (4) has a very similar form as the softmax loss formulation in Eq. (1). Besides this, they serve the same function to separate different class. In fact, if we view $\exp\left(-\sigma D(\mathbf{x}_i, \mathbf{x}_p)\right)$ as the possibility that $\mathbf{x}_i$ and $\mathbf{x}_p$ belong to the same class, Eq. (4) and Eq. (1) are equivalent.

Squeeze loss. This loss is to penalize the variance among positive samples, and "squeeze" the positive neighbors towards the anchor

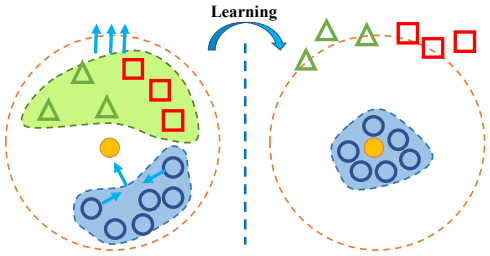


Figure 2: Schematic illustration of the proposed support neighbor loss.

to form a compact cluster. It is defined as:

$$L_{sqz}(\theta) = \sum_{i=1}^{N} \Big(\max\big(\bigcup_{\mathbf{x}_p \in \mathcal{P}_i} D(\mathbf{x}_i, \mathbf{x}_p) \big) - \min\big(\bigcup_{\mathbf{x}_p \in \mathcal{P}_i} D(\mathbf{x}_i, \mathbf{x}_p) \big) \Big), \tag{5}$$

where $\bigcup_{\mathbf{x}_p \in \mathcal{P}_i} D(\mathbf{x}_i, \mathbf{x}_p)$ is the set of distances of every positive neighbor from $\mathcal{P}_i$ to the anchor $\mathbf{x}_i$. Basically, we want to penalize the difference of the distance of an anchor to its furthest positive sample and the distance of the anchor to its nearest positive sample. In this way, a compact cluster of positive samples can be formed around the anchor point.

During training, the two losses are jointly optimized:

$$L_{sn}(\theta) = L_{spr}(\theta) + \lambda L_{sqz}(\theta). \tag{6}$$

where λ is the balancing parameter.

To solve the loss function of Eq. (6), we can follow standard deep learning models by adopting Stochastic gradient descent. Specifically, we need to calculate the gradient of $L_{sn}(\theta)$ w.r.t. θ and we take the i-th training sample $\mathbf{x}_i$ as an example shown as:

$$\begin{aligned} \frac{\partial L_{sn}(\theta)}{\partial \theta} &= \frac{\partial L_{spr}(\theta)}{\partial \theta} + \lambda \frac{\partial L_{sqz}(\theta)}{\partial \theta} \\ &= \left(\frac{\partial L_{spr}(\theta)}{\partial \mathbf{x}_i} + \lambda \frac{\partial L_{sqz}(\theta)}{\partial \mathbf{x}_i} \right) \frac{\partial \mathbf{x}_i}{\partial \theta} \end{aligned} \tag{7}$$

where the key parts are $\frac{\partial L_{spr}(\theta)}{\partial \mathbf{x}_i}$ and $\frac{\partial L_{sqz}(\theta)}{\partial \mathbf{x}_i}$, and the detailed derivatives can be found in the the Appendix. $\frac{\partial \mathbf{x}_i}{\partial \theta}$ is computed using standard backpropagation.

3.3 Implementation Details

Our framework is implemented based on the PyTorch deep learning library. The hardware environment is a PC with Intel Core CPUs (3.6GHz), 48 GB memory, and an NVIDIA GTX TITAN X GPU. Unless specifically noted otherwise, we use the same model structure and training procedure across all experiments and on all datasets.

Training. We use the ResNet50 architecture as our base network and extract feature of 2048 dimensions for each pedestrian image. The weights of the base network are initialized from the model provided by He *et al.* [9]. For all the datasets, the images are resized to 256×128 and are horizontally flipped to augment training samples. We set batch size as 128, with 4 randomly selected images for every 32 identities. We adopt the similar exponentially learning rate decaying schedule as [10]. Specifically, we remain the learning rate r_t unchanged as the base learning rate $r_0 = 2e^{-4}$ until the training epoch reaches T_s. After that, we decay the learning rate in the t-th epoch as $r_t = r_0 * 0.001^{\frac{t-T_s}{T_f-T_s}}$, where T_f is the total

Table 1: Performance comparison on Market-1501. Both single query (SQ) and multiple query (MQ) are evaluated.

	SQ		MQ	
	mAP	rank-1	mAP	rank-1
LDNS [41] (CVPR16)	29.87	55.43	46.03	71.56
GS-CNN [34] (ECCV16)	39.55	65.88	48.45	76.04
CAN [22] (TIP17)	35.9	60.3	47.9	72.1
JLML [18] (IJCAI17)	65.5	85.1	74.5	89.7
DLCE [52] (TOMM17)	59.87	79.51	70.33	85.84
LatParts [14] (CVPR17)	57.53	80.31	66.70	86.79
SSM [2] (CVPR17)	68.80	82.21	68.80	88.18
DLPAR [46] (ICCV17)	63.4	81.0	-	-
LSRO [53] (ICCV17)	56.23	78.06	56.23	85.12
PDC[31] (ICCV17)	63.41	84.14	-	-
IDE-ML [54] (CVPR17)	49.05	73.60	-	-
TriNet [10] (arxiv17)	69.14	84.92	76.42	90.53
DML [44] (CVPR18)	68.83	87.73	77.14	91.66
CSA [55](CVPR18)	68.72	88.12	-	-
Ours	**73.43**	**88.27**	**80.26**	**92.13**
IDE-ML + re-ranking [54] (CVPR17)	63.63	77.11	-	-
TriNet + re-ranking [10] (arxiv17)	81.07	86.67	87.18	91.75
CSA + re-ranking [55] (CVPR18)	71.55	89.49	-	-
Ours + re-ranking	**86.16**	**89.90**	**90.27**	**93.68**

training epoch. In our experiments, we set $r_t = 75$ and $T_f = 800$. We choose the Adam optimizer with the default hyper-parameter values for our experiments.

Testing. In the testing stage, we feed all the testing images to the CNN model to get their feature embeddings. Then we normalize the embeddings to unit vectors. Finally, we compute the Euclidean distances between the embedding vector of each query image and those of every gallery images, and the query images are ranked accordingly.

4 EXPERIMENTS

We validate the effectiveness of the proposed method on three widely used re-ID datasets, namely, Market-1501 [49], CUHK03 [17] and CUHK01 [16]. We follow the same evaluation protocol as the previous papers [1, 17] and evaluate the performance by the cumulated matching characteristics (CMC) curve, which is an estimate of the expectation of finding the correct match in the top n matches. For Market-1501, the mean average precision (mAP) scores are also reported.

4.1 Comparison with State-of-the-Art

Market-1501. This dataset includes images of 1,501 persons captured from 6 different cameras. The pedestrians are cropped with bounding-boxes predicted by DPM detector. The whole dataset is divided into training set with 12,936 images of 751 persons and testing set with 3,368 query images and 19,732 gallery images of 750 persons. There are single-query and multiple-query modes in evaluation, the difference of which is the number of images from the same identity. In multiple-query mode, all features extracted from the images of a person captured by the same camera are merged by average or max pooling, which contains more complete information than single query mode with only 1 query image.

The results for both single-query and multiple-query are shown in Table 1. We can observe that our method gains the highest mAP and rank-1 accuracy among the methods, for both single-query and multi-query cases. Specially, our method achieves about 5% performance gains relative to the most recent methods CSA [55] and DML [44] for the single-query case, while more than 3% gains relative to DML [44] for the multi-query case. Another remarkable comparison lies between our method and TriNet [10], which also focuses on improving the loss function to achieve discriminative deep feature embeddings from pedestrian images. Since our method and TriNet use the same base network (Resnet50) to extract features, the advantage of our method over TriNet for both the single and multi-query case clearly indicates the superiority of our developed loss function. Note that IDE-ML [54] is based on the Softmax loss and our SN significantly outperforms it, which shows the advantage of SN loss over traditional identification loss for re-ID.

Table 2: Performance comparison on CUHK03. Both labeled and detected pedestrian bounding boxes are experimented.

	Labeled		Detected	
	rank-1	rank-5	rank-1	rank-5
LDNS [41] (CVPR16)	62.55	90.05	54.70	84.75
GS-CNN [34] (ECCV16)	-	-	61.8	86.7
CAN [22] (TIP17)	77.6	95.2	69.2	88.5
LatParts [14] (CVPR17)	74.21	94.33	67.99	91.04
SpindleNet [45] (CVPR17)	88.5	97.8	-	-
JLML [18] (IJCAI17)	83.2	98.0	80.6	96.9
DLCE [52] (TOMM17)	-	-	83.4	97.1
DLPAR [46] (ICCV17)	85.4	97.6	81.6	97.3
SSM [2] (CVPR17)	76.63	94.59	72.70	92.40
LSRO [53] (ICCV17)	84.6	97.6	-	-
PDC[31] (ICCV17)	88.70	98.61	78.29	94.83
IDE-ML [54] (CVPR17)	61.6	-	58.5	-
TriNet [10] (arxiv17)	89.6	**99.0**	87.6	**98.2**
Ours	**90.2**	98.8	**88.0**	97.7

We further employ the re-ranking method proposed by Zhong *et al.* [54] to improve the obtained results. The re-ranked results show that our advantages over the most competitive rivalry methods turn even more significant. This further substantiates the superiority of the proposed method.

CUHK03. The CUHK03 dataset was collected by two surveillance cameras capturing 14,096 images of 1,467 persons, with 4.8 images for each person on average. The dataset provides both the automatically cropped bounding boxes with a pedestrian detector (Detected) and the manually cropped bounding boxes (Labeled). We use both labeled and detected bounding boxes for experiments. Following the testing protocol in [17], we randomly divide the identities in the dataset into non-overlapping training and test sets, with 1,367 persons for training and 100 persons for testing. We use the provided 20 partitions for generating training and test sets. During the testing stage, for each person, we randomly pick up one image from one camera view as the probe and another image from the other camera view as the gallery. The reported cumulative matching characteristic (CMC) are averaged by these 20 groups. Table 2 shows the proposed method achieves comparable results with TriNet [10], while surpassing the rest methods with large gaps, especially for the case with detected bounding boxes.

CUHK01. The CUHK01 dataset consists of 3,884 pedestrian images of 971 persons captured by two surveillance cameras. Each person has 4 images, 2 for each camera view. There are two experimental

Table 3: Performance comparison on CUHK01 with 486 test IDs.

	rank-1	rank-5	rank-10
TCP-CNN [7] (CVPR16)	53.7	84.3	91.0
LDNS [41] (CVPR16)	69.09	86.87	91.77
DCSL [43] (IJCAI16)	76.5	**94.2**	**97.5**
SSSVM [42] (CVPR16)	66.0	89.1	92.8
GOG [25] (CVPR16)	57.8	79.1	86.2
WARCA [11] (ECCV16)	65.6	85.3	90.5
CAN [22] (TIP17)	67.2	87.3	92.5
JLML [18] (IJCAI17)	69.8	88.4	93.3
DLPAR [46] (ICCV17)	75.0	93.5	95.7
Ours	**79.3**	94.0	97.2

Table 4: Performance comparison on CUHK01 with 100 test IDs. "Ours" refers the results obtained from the model trained from trained data within the dataset. "Ours (fine-tune)" denotes that results for the model pretrained from the auxiliary dataset and finetuned with training data within the dataset.

	rank-1	rank-5	rank-10
DeepReID [17] (CVPR14)	27.9	58.2	73.5
IDLA [1] (CVPR15)	65.0	88.7	93.1
Deep Ranking [4] (TIP16)	50.4	70.0	84.8
SIR-CIR [36] (CVPR16)	71.8	91.6	96.0
PersonNet [37] (ArXiv16)	71.1	90.1	95.0
EDM [29] (ECCV16)	69.4	90.8	96.0
DCSL [43] (IJCAI16)	89.6	97.8	98.9
DLPAR [46] (ICCV17)	88.5	98.4	99.6
Ours	90.1	98.4	99.0
Ours (fine-tune)	**93.8**	**99.0**	**99.7**

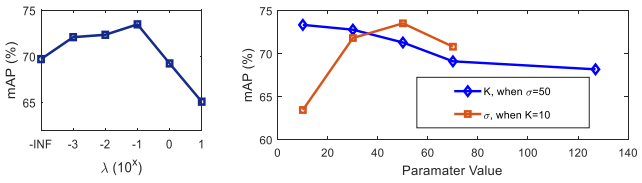
settings, one with 486 identifies for testing and the other with 100 identities for testing. We experiment with both settings.

For the first setting, since there are only a small number of training identities, our network would overfit the training data with high possibility, if directly trained on them. To avoid this, we adopt the strategy of existing methods [1, 46] by firstly training our model with the CUHK03 dataset and then finetuning the model with the 485 training identities. The results in Table 3 show that our method obtains the highest overall matching accuracy, exceeding the most competitive algorithm DCSL [46] by 2.8% in rank-1 accuracy, while being only slightly inferior for the rank-5 and 10 matching accuracies.

For the second setting, the 851 training identities are sufficient enough for our method to avoid the overfiting problem. We directly train our model on the training data. We can see from Table 4 our method achieves very high matching accuracy and is generally better than those of the existing methods. Besides directly training on the 851 training identities, we also train a model finetuned from the pretrained model obtained from the CUHK03 dataset, same as we did for the first setting. The results in Table 3 show that even high performance is achieved.

4.2 Analytic Study

Parameter Analysis. The proposed Support Neighbor (SN) loss in Eq. (6) consists of two the components, the separation loss and the squeeze loss, which are balanced by the hyper-parameter λ. The separation loss serves to enlarge the margins between the positive



(a) (b)

Figure 3: Parameter analysis. (a) the change of mAP value with respect to λ. (b) The change of mAP with respect to parameters K and σ.

neighbors and negative neighbors for an anchor, while the squeeze loss aims to penalize the variance among the positive neighbors. To evaluate the impact of the two losses to the performance, we vary λ with the values in $\{0, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10\}$ and calculate the mAP value on the Market1501 dataset. Note $\lambda = 0$ means the squeeze loss is not considered, and only the separation loss component is used to train the network. The results in the Figure 3(a) show that when simultaneously considering both the two types of losses, higher mAP can be obtained. This substantiates the squeeze loss indeed contributes to more discriminative embeddings than that using the separation loss alone. We also observe the mAP reaches the peak when $\lambda = 0.1$, and thus we use this setting for all the experiments.

Besides λ, there are two other hyper-parameters σ and K in the proposed method. σ is the scaling factor for the separation loss, while K is the number of nearest neighbors selected for calculating the loss. We analyze the impact of the two parameters separately by fixing one while evaluating the other. Figure 3(b) shows the result. We can observe that while fixing σ, the mAP value drops consistently as we enlarge the value for K. For $K = 127$, i.e., all the other samples in a batch are regarded as the neighbors for any given sample, the lowest mAP is obtained. This shows that the incorporation of the less informative samples for calculating the loss may "washing out" the contribution of the high informative ones. A similar observation can be found in [10], where the *Batch Hard* loss (only the hardest triplet is selected for calculating the loss) leads to better performance than the *Batch All* loss (all samples in a batch are used to construct triplets and contribute to the loss).

As to the scaling factor σ, a smaller value for it will leads to a big separation loss. The experimental results show that it is favored to set its value bigger than 30.

Impact of Gallery Size. One of the most challenging problems of applying person re-ID in practice is how to secure satisfactory performance when the gallery set is significantly enlarged. To study the performance of the proposed method for addressing this problem, we employ the distractor set provided along with the Market1501 dataset, use it to supply the gallery set and evaluate the re-ID performance with the enlarged gallery set. The distractor set includes 500 thousand image bounding boxes, consisting of the persons not belonging to any of the original 1,501 identities and false alarms on the background. The distractor set is more than 25 times larger than the original gallery set (about 19 thousands), which makes the retrieval more difficult and aligns better with realistic setting. We progressively add the distractors into the gallery set by random selection. The rank-1 accuracy and mAP value of our method and several existing ones are shown in Figure 4. We can observe that the performances of all the methods degenerate as the gallery

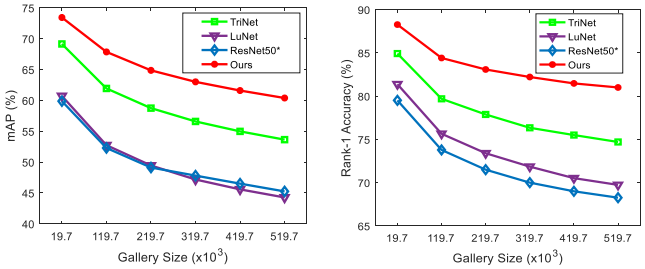


Figure 4: Impact of gallery size. Data of the compared methods (namely, TriNet, LuNet and Resnet50*) are from [10].

set is enlarged. Comparatively speaking, our method consistently maintains the best performance with all gallery sizes, and there is a clear trend that our advantage becomes even more significant as the gallery set turns larger.

Result Visualization. Figure 5 shows some retrieval results on the Market1501 dataset. We can observe that the proposed method is fairly robust to get the correct retrieval results and fails only in some very hard cases that are challenging even for human. Figure 6 shows the learned embedding of the test set of the Market1501 dataset. We can see that images of the same person are closely distributed for the majority of the case. This substantiates the effectiveness of the proposed loss on guiding to learn effective deep embeddings.

5 CONCLUSION

This paper proposed a new loss function for person re-identification, called Support Neighbor (SN) loss, which comprises of the separation component and squeeze component. The separation component serves to push part the positive neighbors from the negative ones of every anchor point within a mini-batch, while the squeeze component pushes together positive neighbors of the anchor to form a compact cluster. We verify the effective of SN loss by integrating it on top of Resnet50, and the experiments show that we achieve the state-of-the-art on several benchmarking datasets. Experiments also show that the two components of SN loss complement with each other, and the advantage of SN loss becomes more significant compared with existing losses when the gallery dataset is tremendously enlarged.

ACKNOWLEDGMENTS

This research is supported in part by the NSF IIS Award 1651902, ONR Young Investigator Award N00014-14-1-0484, and U.S. Army Research Office Award W911NF-17-1-0367.

APPENDIX

In this appendix, we show the key steps of $\frac{\partial L_{spr}(\theta)}{\partial \mathbf{x}_i}$ and $\frac{\partial L_{sqz}(\theta)}{\partial \mathbf{x}_i}$.

First of all, we define $d_{ip} = \sum_{\mathbf{x}_p \in \mathcal{P}_i} \exp\big(-\sigma D(\mathbf{x}_i, \mathbf{x}_p)\big)$ and $d_{is} = \sum_{\mathbf{x}_s \in \mathcal{K}_i} \exp\big(-\sigma D(\mathbf{x}_i, \mathbf{x}_s)\big)$. $D(\mathbf{x}_i, \mathbf{x}_p) = \|f(\mathbf{x}_i|\theta) - f(\mathbf{x}_p|\theta)\|_2^2$ is the Euclidean distance for outputs of sample $\mathbf{x}_i$ and $\mathbf{x}_p$. For simplicity, we further define $\kappa = \frac{d_{ip}}{d_{is}}$, thus we have $\frac{\partial L_{spr}(\theta)}{\partial \mathbf{x}_i} = -\frac{1}{\kappa}\frac{\partial \kappa}{\partial \mathbf{x}_i}$ and for $\frac{\partial \kappa}{\partial \mathbf{x}_i}$, we need to consider three different cases for $\mathbf{x}_i$.

For the first case:

$$\begin{aligned}\frac{\partial \kappa}{\partial \mathbf{x}_i} &= \frac{\frac{\partial d_{ip}}{\partial \mathbf{x}_i} d_{is} - \frac{\partial d_{is}}{\partial \mathbf{x}_i} d_{ip}}{d_{is}^2} \\ &= \kappa\delta\Big(-\sum_{\mathbf{x}_s \in \mathcal{K}_i} \frac{\partial D(\mathbf{x}_i, \mathbf{x}_s)}{\partial \mathbf{x}_i} + \sum_{x_p \in \mathcal{P}_i} \frac{\partial D(\mathbf{x}_i, \mathbf{x}_p)}{\partial x_i}\Big) \\ &= 2\kappa\delta\Big(-\sum_{\mathbf{x}_s \in \mathcal{K}_i} (\mathbf{x}_i - \mathbf{x}_s) + \sum_{\mathbf{x}_p \in \mathcal{P}_i} (\mathbf{x}_i - \mathbf{x}_p)\Big)\end{aligned} \tag{8}$$

For the second case, we consider other sample $\mathbf{x}_q$ find $\mathbf{x}_i$ as its positive pair. Then we $d_{pi} = \sum_{\mathbf{x}_i \in \mathcal{P}(\mathbf{x}_q)} \exp\big(-\sigma D(\mathbf{x}_q, \mathbf{x}_i)\big)$ and $d_{si} = \sum_{\mathbf{x}_i \in \mathcal{K}(\mathbf{x}_q)} \exp\big(-\sigma D(\mathbf{x}_q, \mathbf{x}_i)\big)$:

$$\begin{aligned}\frac{\partial \kappa}{\partial \mathbf{x}_i} &= \frac{\frac{\partial d_{pi}}{\partial \mathbf{x}_i} d_{si} - \frac{\partial d_{si}}{\partial \mathbf{x}_i} d_{pi}}{d_{si}^2} \\ &= \kappa\delta\Big(-\sum_{\mathbf{x}_i \in \mathcal{K}(\mathbf{x}_q)} \frac{\partial D(\mathbf{x}_q, \mathbf{x}_i)}{\partial \mathbf{x}_i} + \sum_{\mathbf{x}_i \in \mathcal{P}(\mathbf{x}_q)} \frac{\partial D(\mathbf{x}_q, \mathbf{x}_i)}{\partial \mathbf{x}_i}\Big) \\ &= 2\kappa\delta\Big(-\sum_{\mathbf{x}_i \in \mathcal{K}(\mathbf{x}_q)} (\mathbf{x}_i - \mathbf{x}_q) + \sum_{\mathbf{x}_i \in \mathcal{P}(\mathbf{x}_q)} (\mathbf{x}_i - \mathbf{x}_q)\Big)\end{aligned} \tag{9}$$

where $\mathcal{K}(\mathbf{x}_q)$ is the set of K-nearest neighbors to $\mathbf{x}_q$ and $\mathcal{P}(\mathbf{x}_q)$ is the set of positive neighbors to $\mathbf{x}_q$.

For the third case, we consider other sample $\mathbf{x}_q$ find $\mathbf{x}_i$ as its neighbors but not positive pair.Then we $d_{pi} = \sum_{\mathbf{x}_i \notin \mathcal{P}(\mathbf{x}_q)} \exp\big(-\sigma D(\mathbf{x}_q, \mathbf{x}_i)\big)$, which is a constant and $d_{si} = \sum_{\mathbf{x}_i \in \mathcal{K}(\mathbf{x}_q)\ \&\ \mathbf{x}_i \notin \mathcal{P}(\mathbf{x}_q)} \exp\big(-\sigma D(\mathbf{x}_q, \mathbf{x}_i)\big)$:

$$\begin{aligned}\frac{\partial \kappa}{\partial \mathbf{x}_i} &= \frac{-\frac{\partial d_{si}}{\partial \mathbf{x}_i} d_{pi}}{d_{si}^2} = \kappa\delta\Big(-\sum_{\mathbf{x}_i \in \mathcal{K}(x_q)\ \&\ \mathbf{x}_i \notin \mathcal{P}(\mathbf{x}_q)} \frac{\partial D(\mathbf{x}_q, \mathbf{x}_i)}{\partial \mathbf{x}_i}\Big) \\ &= 2\kappa\delta\Big(-\sum_{\mathbf{x}_i \in \mathcal{K}(\mathbf{x}_q)\ \&\ \mathbf{x}_i \notin \mathcal{P}(\mathbf{x}_q)} (\mathbf{x}_i - \mathbf{x}_q)\Big).\end{aligned} \tag{10}$$

Thus, we have $\frac{\partial L_{spr}(\theta)}{\partial \mathbf{x}_i} = 2\delta\Big(\sum_{\mathbf{x}_s \in \mathcal{K}_i} (\mathbf{x}_i - \mathbf{x}_s) - \sum_{\mathbf{x}_p \in \mathcal{P}_i} (\mathbf{x}_i - \mathbf{x}_p)\Big) + 2\delta\Big(\sum_{\mathbf{x}_i \in \mathcal{K}(\mathbf{x}_q)} (\mathbf{x}_i - \mathbf{x}_q) - \sum_{\mathbf{x}_i \in \mathcal{P}(\mathbf{x}_q)} (\mathbf{x}_i - \mathbf{x}_q)\Big) + 2\delta\Big(\sum_{\mathbf{x}_i \in \mathcal{K}(\mathbf{x}_q)\ \&\ \mathbf{x}_i \notin \mathcal{P}(\mathbf{x}_q)} (\mathbf{x}_i - \mathbf{x}_q)\Big)$.

For $\frac{\partial L_{sqz}(\theta)}{\partial \theta}$, we need also consider three cases, i.e., $\mathbf{x}_i$ is the anchor, $\mathbf{x}_j$ is the anchor and $\mathbf{x}_i$ becomes the closest or farthest sample:

$$\begin{aligned}\frac{\partial L_{sqz}(\theta)}{\partial \theta} &= 2(\mathbf{x}_i - x_f) - 2(\mathbf{x}_i - \mathbf{x}_c) + \sum_{\mathbf{x}_i = \mathcal{P}(\mathbf{x}_j, f)} 2(\mathbf{x}_i - x_j) \\ &\quad + \sum_{\mathbf{x}_i = \mathcal{P}(\mathbf{x}_j, c)} 2(\mathbf{x}_i - \mathbf{x}_j) \\ &= 2\mathbf{x}_c - \mathbf{x}_f + \sum_{\mathbf{x}_i = \mathcal{P}(\mathbf{x}_j, f)} 2(\mathbf{x}_i - \mathbf{x}_j) \\ &\quad + \sum_{\mathbf{x}_i = \mathcal{P}(\mathbf{x}_j, c)} 2(\mathbf{x}_i - \mathbf{x}_j)\end{aligned} \tag{11}$$

where $\mathbf{x}_f$ is the farthest sample while $\mathbf{x}_c$ is the closet sample when $\mathbf{x}_i$ works as the anchor. $\mathcal{P}(\mathbf{x}_j, f)$ is the farthest positive sample of $\mathbf{x}_j$ and $\mathcal{P}(\mathbf{x}_j, c)$ is the closest positive sample of $\mathbf{x}_j$.

So far, we have obtained the derivatives of $\frac{\partial L_j(\theta)}{\partial \mathbf{x}_i}$ and $\frac{\partial L_{sqz}(\theta)}{\partial \mathbf{x}_i}$.

Figure 5: Sample person retrieval results on Market1501. The most left (without border) in each of the six groups is the query and the rest are the top-10 ranked returns from the gallery. Correct returns are in green borders, while incorrect ones are in red borders.

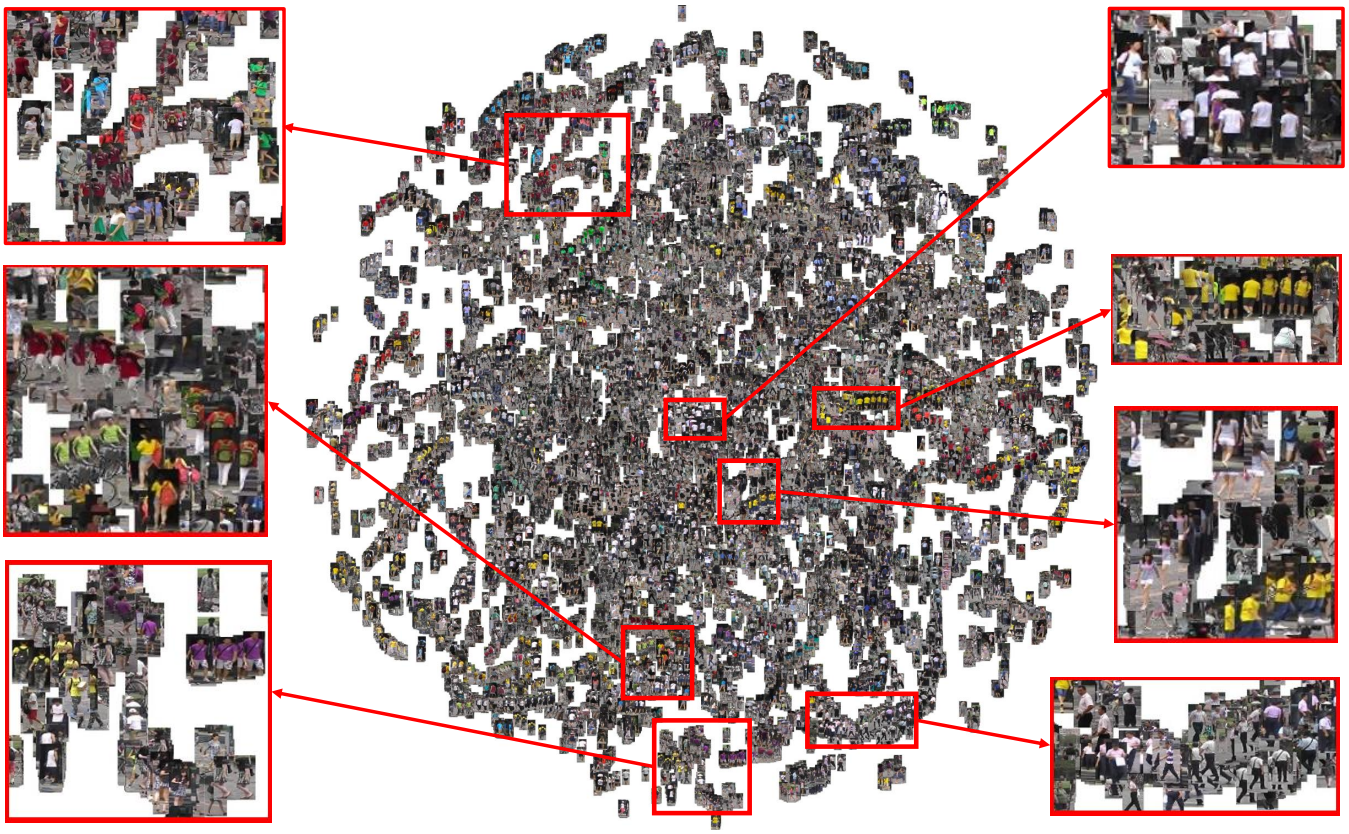


Figure 6: The Barnes-Hut t-SNE [33] of the learned embeddings for the test set of the Market-1501 dataset. We can observe that satisfactory embeddings have been achieved and images of the same identities are closely distributed.

REFERENCES

[1] Ejaz Ahmed, Michael Jones, and Tim K Marks. 2015. An Improved Deep Learning Architecture for Person Re-Identification. In *CVPR*.

[2] Song Bai, Xiang Bai, and Qi Tian. 2017. Scalable person re-identification on supervised smoothed manifold. In *CVPR*.

[3] Dapeng Chen, Zejian Yuan, Badong Chen, and Nanning Zheng. 2016. Similarity learning with spatial constraints for person re-identification. In *CVPR*.

[4] Shi-Zhe Chen, Chun-Chao Guo, and Jian-Huang Lai. 2016. Deep ranking for person re-identification via joint representation learning. *IEEE Transactions on Image Processing* 25, 5 (2016), 2353–2367.

[5] Weihua Chen, Xiaotang Chen, Jianguo Zhang, and Kaiqi Huang. 2017. Beyond triplet loss: a deep quadruplet network for person re-identification. In *CVPR*.

[6] Weihua Chen, Xiaotang Chen, Jianguo Zhang, and Kaiqi Huang. 2017. A Multi-Task Deep Network for Person Re-Identification. In *AAAI*.

[7] De Cheng, Yihong Gong, Sanping Zhou, Jinjun Wang, and Nanning Zheng. 2016. Person re-identification by multi-channel parts-based cnn with improved triplet loss function. In *CVPR*.

[8] Shengyong Ding, Liang Lin, Guangrun Wang, and Hongyang Chao. 2015. Deep feature learning with relative distance comparison for person re-identification. *Pattern Recognition* 48, 10 (2015), 2993–3003.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *CVPR*.

[10] Alexander Hermans, Lucas Beyer, and Bastian Leibe. 2017. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737* (2017).

[11] Cijo Jose and François Fleuret. 2016. Scalable metric learning via weighted approximate rank component analysis. In *ECCV*.

[12] Martin Köstinger, Martin Hirzer, Paul Wohlhart, Peter M. Roth, and Horst Bischof. 2012. Large scale metric learning from equivalence constraints. In *CVPR*.

[13] Ryan Layne, Timothy M Hospedales, Shaogang Gong, and Q Mary. 2012. Person Re-identification by Attributes.. In *BMVC*.

[14] Dangwei Li, Xiaotang Chen, Zhang Zhang, and Kaiqi Huang. 2017. Learning deep context-aware features over body and latent parts for person re-identification. In *CVPR*.

[15] Kai Li, Zhengming Ding, Sheng Li, and Yun Fu. 2018. Discriminative Semi-Coupled Projective Dictionary Learning for Low-Resolution Person Re-Identification. In *AAAI*.

[16] Wei Li, Rui Zhao, and Xiaogang Wang. 2012. Human reidentification with transferred metric learning. In *ACCV*.

[17] Wei Li, Rui Zhao, Tong Xiao, and Xiaogang Wang. 2014. Deepreid: Deep filter pairing neural network for person re-identification. In *CVPR*.

[18] Wei Li, Xiatian Zhu, and Shaogang Gong. 2017. Person re-identification by deep joint learning of multi-loss classification. *arXiv preprint arXiv:1705.04724* (2017).

[19] Zhen Li, Shiyu Chang, Feng Liang, Thomas S Huang, Liangliang Cao, and John R Smith. 2013. Learning locally-adaptive decision functions for person verification. In *CVPR*.

[20] Shengcai Liao, Yang Hu, Xiangyu Zhu, and Stan Z Li. 2015. Person re-identification by Local Maximal Occurrence Representation and Metric Learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2197–2206.

[21] Shengcai Liao and Stan Z Li. 2015. Efficient PSD Constrained Asymmetric Metric Learning for Person Re-identification. In *Proceedings of the IEEE International Conference on Computer Vision*. 3685–3693.

[22] Hao Liu, Jiashi Feng, Meibin Qi, Jianguo Jiang, and Shuicheng Yan. 2017. End-to-end comparative attention networks for person re-identification. *IEEE Transactions on Image Processing* 26, 7 (2017), 3492–3506.

[23] Xiao Liu, Mingli Song, Qi Zhao, Dacheng Tao, Chun Chen, and Jiajun Bu. 2012. Attribute-restricted latent topic model for person re-identification. *Pattern recognition* 45, 12 (2012), 4204–4213.

[24] Bingpeng Ma, Yu Su, and Frédéric Jurie. 2012. Local Descriptors Encoded by Fisher Vectors for Person Re-identification. In *Proceedings of the European Conference on Computer Vision Workshops and Demonstration*. 413–422.

[25] Tetsu Matsukawa, Takahiro Okabe, Einoshin Suzuki, and Yoichi Sato. 2016. Hierarchical gaussian descriptor for person re-identification. In *CVPR*.

[26] Alexis Mignon and Frédéric Jurie. 2012. PCCA: A new approach for distance learning from sparse pairwise constraints. In *CVPR*.

[27] Sateesh Pedagadi, James Orwell, Sergio A. Velastin, and Boghos A. Boghossian. 2013. Local Fisher Discriminant Analysis for Pedestrian Re-identification. In *CVPR*. 3318–3325.

[28] Yang Shen, Weiyao Lin, Junchi Yan, Mingliang Xu, Jianxin Wu, and Jingdong Wang. 2015. Person Re-identification with Correspondence Structure Learning. In *ICCV*.

[29] Hailin Shi, Yang Yang, Xiangyu Zhu, Shengcai Liao, Zhen Lei, Weishi Zheng, and Stan Z Li. 2016. Embedding deep metric for person re-identification: A study against large variations. In *ECCV*.

[30] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. 2016. Deep metric learning via lifted structured feature embedding. In *CVPR*.

[31] Chi Su, Jianing Li, Shiliang Zhang, Junliang Xing, Wen Gao, and Qi Tian. 2017. Pose-driven Deep Convolutional Model for Person Re-identification. In *ICCV*.

[32] Chi Su, Fan Yang, Shiliang Zhang, Qi Tian, Larry S Davis, and Wen Gao. 2015. Multi-Task Learning with Low Rank Attribute Embedding for Person Re-identification. In *ICCV*.

[33] Laurens Van Der Maaten. 2014. Accelerating t-SNE using tree-based algorithms. *Journal of machine learning research* 15, 1 (2014), 3221–3245.

[34] Rahul Rama Varior, Mrinal Haloi, and Gang Wang. 2016. Gated siamese convolutional neural network architecture for human re-identification. In *ECCV*.

[35] Rahul Rama Varior, Bing Shuai, Jiwen Lu, Dong Xu, and Gang Wang. 2016. A siamese long short-term memory architecture for human re-identification. In *ECCV*.

[36] Faqiang Wang, Wangmeng Zuo, Liang Lin, David Zhang, and Lei Zhang. 2016. Joint learning of single-image and cross-image representations for person re-identification. In *CVPR*.

[37] Lin Wu, Chunhua Shen, and Anton van den Hengel. 2016. Personnet: Person re-identification with deep convolutional neural networks. *arXiv preprint arXiv:1601.07255* (2016).

[38] Ziyan Wu, Yang Li, and Richard J Radke. 2015. Viewpoint invariant human re-identification in camera networks using pose priors and subject-discriminative features. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37, 5 (2015), 1095–1108.

[39] Tong Xiao, Hongsheng Li, Wanli Ouyang, and Xiaogang Wang. 2016. Learning deep feature representations with domain guided dropout for person re-identification. In *CVPR*.

[40] Fei Xiong, Mengran Gou, Octavia I. Camps, and Mario Sznaier. 2014. Person Re-Identification Using Kernel-Based Metric Learning Methods. In *ECCV*.

[41] Li Zhang, Tao Xiang, and Shaogang Gong. 2016. Learning a discriminative null space for person re-identification. In *CVPR*.

[42] Ying Zhang, Baohua Li, Huchuan Lu, Atshushi Irie, and Xiang Ruan. 2016. Sample-specific svm learning for person re-identification. In *CVPR*.

[43] Yaqing Zhang, Xi Li, Liming Zhao, and Zhongfei Zhang. 2016. Semantics-Aware Deep Correspondence Structure Learning for Robust Person Re-Identification.. In *IJCAI*.

[44] Ying Zhang, Tao Xiang, Timothy M. Hospedales, and Huchuan Lu. 2018. Deep Mutual Learning. (2018).

[45] Haiyu Zhao, Maoqing Tian, Shuyang Sun, Jing Shao, Junjie Yan, Shuai Yi, Xiaogang Wang, and Xiaoou Tang. 2017. Spindle net: Person re-identification with human body region guided feature decomposition and fusion. In *CVPR*.

[46] Liming Zhao, Xi Li, Jingdong Wang, and Yueting Zhuang. 2017. Deeply-learned part-aligned representations for person re-identification. (2017).

[47] Rui Zhao, Wanli Ouyang, and Xiaogang Wang. 2013. Person Re-identification by Salience Matching. In *Proceedings of the IEEE International Conference on Computer Vision*. 2528–2535.

[48] Rui Zhao, Wanli Ouyang, and Xiaogang Wang. 2013. Unsupervised Salience Learning for Person Re-identification. In *CVPR*.

[49] Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. 2015. Scalable Person Re-identification: A Benchmark. In *ICCV*.

[50] Liang Zheng, Yi Yang, and Alexander G Hauptmann. 2016. Person re-identification: Past, Present and Future. *arXiv preprint arXiv:1610.02984* (2016).

[51] Liang Zheng, Hengheng Zhang, Shaoyan Sun, Manmohan Chandraker, and Qi Tian. 2017. Person re-identification in the wild. *arXiv preprint* (2017).

[52] Zhedong Zheng, Liang Zheng, and Yi Yang. 2017. A Discriminatively Learned CNN Embedding for Person Reidentification. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 14, 1 (2017), 13.

[53] Zhedong Zheng, Liang Zheng, and Yi Yang. 2017. Unlabeled samples generated by gan improve the person re-identification baseline in vitro. (2017).

[54] Zhun Zhong, Liang Zheng, Donglin Cao, and Shaozi Li. 2017. Re-ranking person re-identification with k-reciprocal encoding. In *CVPR*.

[55] Zhun Zhong, Liang Zheng, Zhedong Zheng, Shaozi Li, and Yi Yang. 2018. Camera Style Adaptation for Person Re-identification. In *CVPR*.