Deep View-Aware Metric Learning for Person Re-Identification

Pu Chen, Xinyi Xu and Cheng Deng*
School of Electronic Engineering, Xidian University, Xi’an 710071, China
puchen@stu.xidian.edu.cn, xyxu.xd@gmail.com, chdeng@mail.xidian.edu.cn

Abstract

Person re-identification remains a challenging issue due to the dramatic changes in visual appearance caused by the variations in camera views, human pose, and background clutter. In this paper, we propose a deep view-aware metric learning (DVAML) model, where image pairs with similar and dissimilar views are projected into different feature subspaces, which can discover the intrinsic relevance between image pairs from different aspects. Additionally, we employ multiple metrics to jointly learn feature subspaces on which the relevance between image pairs are explicitly captured and thus greatly promoting the retrieval accuracy. Extensive experiment results on datasets CUHK01, CUHK03, and PRID2011 demonstrate the superiority of our method compared with state-of-the-art approaches.

1 Introduction

Person re-identification attracts increasing attentions due to its importance in many applications, e.g., video surveillance, pedestrian retrieval, and human-computer interaction. Although significant progress has been made in these years, there are still some challenging problems existing in person re-identification: 1) dramatic changes in visual appearance; 2) dissimilarity between two images of the same pedestrian; 3) similarity between two images of different pedestrians.

To address these issues, typical methods usually focus on extracting robust features or finding a similarity measure. For instance, semantic features from different body regions are captured through a multi-stage ROI pooling pipeline [Zhao et al., 2017]. In [Zhang et al., 2016], a semantics-aware image representation is learned to capture the intrinsic structure information of persons. In [Cheng et al., 2016], an enhanced triplet loss function is proposed to learn a distance measure between two pedestrian images.

However, these approaches are incapable to efficiently discover the intrinsic relevance between image pairs. Fortunately, we find that the intrinsic relevance between image pairs in similar views is different with that in dissimilar views. As shown in Figure 1(a) and (b), the image pairs of the same pedestrian captured from similar views can be connected by some details, such as the patterns of clothing or the carried stuff. As shown in Figure 1(c) and (d), the appearances of pedestrians change dramatically and some details are missed when the image pairs of the same pedestrian are captured from dissimilar views. Even so, the image pairs can also be connected by some view-robust features. Therefore, we can exploit the views as the supplementary information to capture the intrinsic relevance between image pairs.

In this paper, we propose a novel approach called deep view-aware metric learning (DVAML) model, where the image pairs are projected into different feature subspaces according to their view information, which can discover the intrinsic relevance between image pairs from different aspects. Multiple metrics are exploited to supervise the learning of the different feature subspaces. Extensive experiments conducted on three datasets including CUHK01, CUHK03, and PRID2011 show the effectiveness of our method compared with state-of-the-art approaches.

2 Related Work

Typical person re-identification methods usually focus on feature extraction or metric learning. Widely-used features include histograms [Li and Wang, 2013; Khamis et al., 2014;
local binary patterns (LBP) [Li and Wang, 2013; Zhao et al., 2013a; Khamis et al., 2014], Gabor features [Li and Wang, 2013] and other cues [Zhang et al., 2014]. However, the changes of appearance lead to instability of these features. The basic idea of metric learning is to find a mapping function from feature space to distance space with certain merits [Cheng et al., 2016], such as Mahalanobis metric learning (KISSME) [Davis et al., 2007], Information-theoretic metric learning (ITML) [Davis et al., 2007], and large margin nearest neighbour (LMNN) [Weinberger and Saul, 2009]. Both lines of the methods regard feature extraction and metric learning processes as two separate steps, which limit the performances significantly.

Inspired by the great success of deep learning networks in computer vision and pattern recognition tasks [Yang et al., 2017b; Deng et al., 2018; Yang et al., 2018; Li et al., 2018; Yang et al., 2017a; Liu et al., 2016; Yang et al., 2017c], many researchers consider the feature and metric learning jointly in an integrated deep architecture, where feature representation can be learned under supervision of the distance loss. In [Li et al., 2014], deep learning is exploited to automatically learn features for the re-identification task. And the deep framework has been improved in [Ahmed et al., 2015] by incorporating neighbouring locations of other images. In [Ding et al., 2015], the relative distance between two images are captured by a triplet loss. To further constrain the distances of pairs, [Cheng et al., 2016] propose an enhance triplet loss function. A multi-task deep network (MTDnet) [Chen et al., 2017] are presented to consider the classification loss and the ranking loss simultaneously and takes advantages both of them during training. In order to jointly extract single-image and cross-image feature representations, [Wang et al., 2016] propose a unified triplet and siamese deep architecture.

3 Our Architecture

Fig. 3 shows the framework of the proposed method. The input images of the network are first embedded by the convolution neural network (CNN). Then the output feature maps are respectively projected to different feature subspaces through two full connected layers that do not share parameters. Finally, multiple metric loss functions are used to guide network optimization.

3.1 View-Aware Feature Embedding

Given a training set \( \{x_1, y_1^1, y_1^2 \}_{i=1,2,...,n} \), where \( x_i \) denotes the \( i \)th pedestrian image, \( y_1^1 \) and \( y_1^2 \) are the identity label and the view label of \( x_i \) respectively. The view label \( y_2^i \) is set as 0 when the pedestrian is in front or back views, while as 1 when the pedestrian is in side views. Furthermore, we denote the feature maps of input images extracted by CNN as \( F = \{f(x_i)\}_{i=1,2,...,n} \) and combine them into four kinds of pairs as shown on Figure 2:

\[
\begin{align*}
PS &= \{(f(x_i), f(x_j)) | y_1^i = y_1^j, y_2^i = y_2^j \} \\
PD &= \{(f(x_i), f(x_j)) | y_1^i = y_1^j, y_2^i \neq y_2^j \} \\
NS &= \{(f(x_i), f(x_j)) | y_1^i \neq y_1^j, y_2^i = y_2^j \} \\
ND &= \{(f(x_i), f(x_j)) | y_1^i \neq y_1^j, y_2^i \neq y_2^j \}
\end{align*}
\]  

where \( 1 \leq i, j \leq n \), the feature maps of image pairs \( (f(x_i), f(x_j)) \in \{PS, NS\} \) are projected to one feature subspace by the full connection layer FC1, while the \( (f(x_k), f(x_l)) \in \{PD, ND\} \) are projected to another one by FC2. The outputs of FC1 and FC2 are defined as follows:

\[
\begin{align*}
g_1(f(x_i)) &= W_1 \ast f(x_i) + b_1 \\
g_2(f(x_k)) &= W_2 \ast f(x_k) + b_2
\end{align*}
\]  

where \( W_1 \) and \( W_2 \) are the weights of FC1 and FC2 while \( b_1 \) and \( b_2 \) represent the biases. \( \ast \) refers to the convolution operation.

The CNN is built inspired by [Xiao et al., 2016; Simonyan and Zisserman, 2014] and details of the structure are listed in Table 1. Each ReLU layer is followed by a Batch Normalization (BN) layer which accelerates the convergence process and avoids manually tweaking the initialization of weights and biases. For training the CNN from scratch, we randomly dropout 50% neurons of the fc7_1 and fc7_2 layers.

Besides, We learn a view classifier to choose the corresponding feature subspace which the feature maps of images are projected into. The view classifier is optimized by minimizing the sigmoid loss. The first and second layers of the view classifier are convolutional layers with the kernel size \( 32 \times 7 \times 7 \) and \( 32 \times 5 \times 5 \) respectively. The kernels size of third convolutional layer is \( 64 \times 3 \times 3 \) and all the three layers are followed by a pooling layer. Then two fully connected layers that contain 512 neurons receive the output of the third convolutional layer.

3.2 Multiple Metric Loss

Different with other methods, we utilize two metric loss functions to constrain the distances between image pairs in similar views and in dissimilar views separately. Figure 4 illustrates the failure case of other methods that constrain the distances between all image pairs with one loss. In Figure 4(a), the loss decreases the distances between \( (x_1, x_2) \) and between
Figure 3: Architecture of the proposed deep view-aware metric learning (DVAML) model. The input of the network is a set of images. Then the feature maps output by the CNN are respectively projected to different feature subspaces through two full connected layers. Finally, three loss functions are used to guide network optimization.

\[(x_1, x_3)\) by giving \(x_1\) a upwards and a downwards pulling force respectively. Simultaneously, \(x_1\) is given a left and a right pushing force to increase the distances between \(x_1, x_3\) and between \(x_1, x_5\). Therefore, \(x_1\) will still in place. In Figure 4(b), image pairs in similar views and in dissimilar views are first projected into two different feature subspaces as introduced in 3.1 section. Then \(L^1\) and \(L^2\) are used to supervise the learning of the different feature subspaces respectively. \(L^1\) decreases the distance between \((x_1, x_2)\) by pulling \(x_1\) upwards and increases the distance between \((x_1, x_5)\) by pushing \(x_1\) right, while \(L^2\) decreases the distance between \((x_1, x_3)\) by pulling \(x_1\) downwards and increases the distance between \((x_1, x_4)\) by pushing \(x_1\) left. \(x_1\) moves finally towards the correct direction in both feature subspaces.

The loss function \(L^1\) based on the lifted structured feature embedding[Song et al., 2016] is defined as:

\[
L^1_{i,j} = \log \left[ \sum_{(i,k) \in NS} \exp(\beta_1 - D_{i,k}) \right] + \sum_{(j,l) \in NS} \exp(\beta_1 - D_{j,l}) + D_{i,j} 
\]

\[
L^1 = \frac{1}{2|PS|} \sum_{(i,j) \in PS} \max(0, L^1_{i,j})^2 
\]

where \(L^1_{i,j}\) is used to limit the distance between \(x_i\) and \(x_j\) in \(PS\) less than all pairs in \(NS\) which contain \(x_i\) or \(x_j\) with a margin \(\beta_1\).

Correspondingly, the loss function \(L^2\) is defined as:

\[
L^2_{i,j} = \log \left[ \sum_{(i,k) \in ND} \exp(\beta_2 - D_{i,k}) \right] + \sum_{(j,l) \in ND} \exp(\beta_2 - D_{j,l}) + D_{i,j} 
\]

\[
L^2 = \frac{1}{2|PD|} \sum_{(i,j) \in PD} \max(0, L^2_{i,j})^2 
\]

where \(L^2_{i,j}\) is used to limit the distance between \(x_i\) and \(x_j\) in \(PD\) less than all pairs in \(ND\) which contain \(x_i\) or \(x_j\) with a margin \(\beta_2\). The value \([PS]\) and \([PD]\) are respectively equal to the number of image pairs belong to \(PS\) and \(PD\) in each batch.

Noted that the distances between images pairs in similar and dissimilar views are respectively denoted as:

\[
D_{i,j} = \begin{cases} 
\|g_1(f(x_i)) - g_1(f(x_j))\|_2 & (i,j) \in S \\
\|g_2(f(x_i)) - g_2(f(x_j))\|_2 & (i,j) \in D 
\end{cases} 
\]

in which \(S\) denotes the set of image pairs in similar views including \(PS\) and \(NS\), while \(D\) denotes the set of image pairs with dissimilar views consisting of \(PD\) and \(ND\).

Considering that the loss function \(L^1\) and \(L^2\) only constrains the distances between image pairs in similar views and dissimilar views separately. So we use \(L^3\) constrain the distances between image pairs of same person and between image pairs of different person overall. The loss function \(L^3\) is defined as:

\[
L^3_{i,j} = \log \left[ \sum_{(i,k) \in N} \exp(\beta_3 - D_{i,k}) \right] + \sum_{(j,l) \in N} \exp(\beta_3 - D_{j,l}) + D_{i,j} 
\]

\[
L^3 = \frac{1}{2|PS| + |PD|} \sum_{(i,j) \in P} \max(0, L^3_{i,j})^2 
\]

in which \(P\) denotes the set of the image pairs of same identity consist of \(PS\) and \(PD\). \(N\) constrains all the image pairs in \(NS\) and \(ND\).

We finally unite the three loss functions to a joint loss function:

\[
L = \alpha_1 L^1 + \alpha_2 L^2 + \alpha_3 L^3 
\]

where the three hyper-parameters \(\alpha_1\), \(\alpha_2\) and \(\alpha_3\) are used to balance the three loss functions.
Figure 4: The rounds $x_1$, $x_2$ and $x_3$ are images of person A, while
the triangles $x_4$ and $x_5$ are images of person B. The solid and hollow
denote the two different views of the person respectively. The red
arrowhead means the pulling force and the blue arrowhead means
the push force.

### 3.3 Optimization

The network is jointly supervised and optimized by the joint
loss. The back propagation gradient with respect to the distance $D_{i,j}$ is:

$$
\frac{\partial L}{\partial D_{i,j}} = \frac{\partial L^1}{\partial D_{i,j}} 1[(i, j) \in \text{PS}] + \frac{\partial L^2}{\partial D_{i,j}} 1[(i, j) \in \text{PD}] + \frac{\partial L^3}{\partial D_{i,j}}
$$

(11)

$$
\frac{\partial L^1}{\partial D_{i,j}} = \frac{1}{|\text{PS}|} L_{i,j}^1 1[L_{i,j}^1 > 0]
$$

(12)

$$
\frac{\partial L^2}{\partial D_{i,j}} = \frac{1}{|\text{PD}|} L_{i,j}^2 1[L_{i,j}^2 > 0]
$$

(13)

$$
\frac{\partial L^3}{\partial D_{i,j}} = \frac{1}{|\text{PS}| + |\text{PD}|} L_{i,j}^3 1[L_{i,j}^3 > 0]
$$

(14)

The gradient with respect to the distance $D_{i,k}$ is:

$$
\frac{\partial L}{\partial D_{i,k}} = \frac{\partial L^1}{\partial D_{i,k}} 1[(i, k) \in \text{NS}] + \frac{\partial L^2}{\partial D_{i,k}} 1[(i, k) \in \text{ND}] + \frac{\partial L^3}{\partial D_{i,k}}
$$

(15)

in which the gradients of the three loss functions with respect to
$D_{i,k}$ are:

$$
\frac{\partial L^1}{\partial D_{i,k}} = L_{i,k}^1 1[L_{i,k}^1 > 0] \sum_{(i,j) \in \text{PS}} \frac{-\exp[\beta_1 - D_{i,j}]}{\exp[L_{i,j}^1 - D_{i,j}]}
$$

(16)

### 4 Experiments

#### 4.1 Datasets and Settings

**Datasets:** We evaluate our method on three datasets,
CUHK03 [Li et al., 2014], CUHK01 [Li et al., 2012] and
PRID2011 [Hirzer et al., 2011]. CUHK03 dataset contains
13164 images from 1360 persons. We select 1160 persons
for training, 100 for validation and 100 for testing following the same setting as [Li et al., 2014] and [Ahmed et al., 2015]. CUHK01 [Li et al., 2012] dataset contains 971 persons captured from two camera views in a campus. We utilize the same protocol used in [Zhou et al., 2017] and [Wang et al., 2016], where 871 person images are used for training and the left for testing. Following the setting in [Chen et al., 2017], 100 persons in PRID2011 dataset are used for training, specially, 100 for probe and 649 (the remaining persons from camera B except the 100 for gallery) for gallery in test set.

For training the view classifier, we manually annotate the coarse view labels for 200 persons in CUHK03 and CUHK01. Considering the size and complexity of PRID2011 dataset, 50 persons in it are annotated. Since the view classification is treated as a simple binary classification problem, the view classifier achieves a high accuracy rate even though training samples are limited.

Considering that multiple cameras in varied views may be retrieved for a given query image in practical application, we fuse the three datasets two by two into three new datasets: Syn1 (CUHK03 and CUHK01), Syn2 (CUHK03 and PRID2011) and Syn3 (CUHK01 and PRID2011). Then we train the model and retrieve probe of the one on the two datasets for each fused. For example, the same 1260 persons in CUHK03 and 871 persons in CUHK01 are chosen as Syn1’s training set. For the probe of CUHK03, there are another 100 persons from CUHK01 in the CUHK03’s gallery besides the original 100. Equally, there are 200 persons in the CUHK01’s gallery. Noted that 749 persons are used as the gallery of PRID2011 in Syn2 and Syn3, while only 200 persons as the CUHK01’s and CUHK03’s gallery.

**Parameters Implementation.** All the images are resized to 128 x 64 before being fed to the network and the batchsize of the input is 64. For identity classifier, we first pretrain a model on CUHK03 and the learning rate is set to 0.001, then finetune this model with learning rate 0.002. We picked a set of optimal loss weights $\alpha_1 = 0.4$, $\alpha_2 = 0.4$, $\alpha_3 = 0.2$ experimentally. And all the margin parameters $\beta_1$, $\beta_2$, $\beta_3$ are set to 1 [Song et al., 2016].

**Evaluation Protocol:** We report the single-shot results on
we compare our method with 7 common methods including KISSME, ITML, LMNN, eSDC [Zhao et al., 2013b], kLFDA [Xiong et al., 2014], FPNN [Li et al., 2014], ID-LA [Ahmed et al., 2015]. And Table 2-7 show the evaluation results of these method.

Results on CUHK03. In Table 2, we also compare our method with two deep methods GatedSiamese [Varior et al., 2016] and MTDnet [Chen et al., 2017] except the 7 methods. From Table1, most of deep learning methods except FPNN obviously outperform the traditional methods. Ours model has achieved the top performances by 76.11% on Top1 accuracy (vs. 74.68% by the next best method).

Results on CUHK01. Table 3 extra lists the results of the P2S [Zhou et al., 2017] and ImpTrpLoss [Cheng et al., 2016] methods. From Table 3, we can see that our method outperforms the state-of-the-art methods by more than 80% on Top1 accuracy (80.54% vs. 78.50% by MTDnet-cross). Note that the Top 5 recognition rate of our method reaches 95.00%, meaning that the trained model has high probability of finding the correct person from other cameras.

Results on PRID2011. We can see from Table 4 that our method achieves 33% on Top 1 matching rate and outperforms the previous best accuracy on PRID2011 even though the training samples are limited for deep neural networks. It is noted that the Top 5 matching rate achieves 65% far beyond the accuracy of other methods.

Table 5-7 illustrates the performance of ours method and the previous methods on synthetic datasets. And it’s obvious that our method has achieved the top performance on all the three datasets. From Table 5, we can see that the matching rate of our method on CUHK03 and CUHK01 reaches 72.77% and 73.00%. It’s worth noting that the accuracy on Syn1 declines contrast to each single dataset due to the gallery images from the other dataset can interfere the retrieval of the probe. The results on Syn2 are listed in Table 6 and our method’s accuracy achieves 75.02% and 35.00% on CUHK03 and PRID2011 respectively. The accuracy of PRID2011 when trained on Syn2 dataset improves 2% compared with on single dataset, since the added training samples may improve the performance of model on small dataset. The accuracy of CUHK01 and PRID2011 listed in Table 7 reaches 76.00% and 31.00%. It is noted that we improve the performance on PRID2011 by more than 20% on the rank-5 accuracy in contrast to the previous methods.

Effect of deep view-aware metric. We employ an view-aware method to project image pairs into different feature subspaces. The method has positive effect on person re-identification since it captures the intrinsic relevance between image pairs. Besides, the proposed network is constrained by three metric loss functions, of which the two supervise separately the learning of the different feature subspaces and the last supervises the whole network. And our model wins over 3% compared to other methods on the three synthetic datasets, which further proves the effectiveness of it.

5 Conclusion
In this paper, we propose a deep view-aware metric learning (DVAML) model for person re-identification. We project image pairs in similar and dissimilar views into different feature subspaces, respectively. Then multiple loss functions are used to supervise the learning of the different feature subspaces. In addition, we also consider the actual situation that a given probe image can be retrieved from multiple cameras by synthesizing different datasets. Experiment results on CUHK03, CUHK01, PRID2011 and three synthetic datasets illustrate that our method outperforms the state-of-the-art approaches.

Acknowledgments
This work was supported in part by the National Natural Science Foundation of China under Grant 61572388 and Grant 61703327, and in part by the Key R&D Program. The Key Industry Innovation Chain of Shaanxi under Grant 2017ZDCXL-GY-05-04-02 and Grant 2017ZDCXL-GY-05-04-02.
References


