Person re-identification based on and kernel local fisher discriminant analysis and Mahanalobis distance learning

Qiangsen He
School of Electrical Engineering and Computer Science, University of Ottawa
Ottawa, Ontario, qiangsenhe@gmail.com

Abstract—**Person re-identification has been an intense research area recently. It's very important to choose robust descriptors and metric learning to improve accuracy. Mahanalobis based metric learning is a popular method to measure similarity. However, since directly extracted descriptors usually have high dimension, it's intractable to learn a high dimensional Mahanalobis matrix. Dimension reduction are used to project high dimensional descriptors to lower dimension space while preserving those discriminative information as much as possible. In this paper the kernel LFDA is used to reduce dimension given that kernelization method can greatly improve re-identification performance for nonlinearity. Then a metric matrix is learned on lower dimensional descriptors based on the limitation that the within class distance is at least 1 unit smaller than the minimum inter class distance. This method turns to have excellent performance compared with other adcanced metric learning.**

1. Introduction

Person re-identification(re-ID) has received increasing attention in recent years. The task of re-ID is to match a given person with a list of identities with known labels. This problem is very challenging caused by many factors like low image resolution, occlusion, background noise and different camera color response, etc. According to sample size of each person, re-ID can be divided into single shot and multi-shot re-ID. In the single shot re-ID problem, since only one image is provided in each camera for each person, it might be quite confusing when different people have similar pose or clothes. Also, in the multishots case, there might exist quite much difference even in different frames of the same person for different pose and illuminations. Therefore, good descriptors are supposed to be robust to illumination change and occlusions.

Since most datasets are well images with well cropped bounding boxes, classical re-ID focus on descriptors extraction and metric learning. It's important to select a proper descriptor for images. Most descriptors are color and texture based descriptors. One simple descriptor is the histogram descriptor. In [] the image is divided into a few horizontal strides, for each slide the color histogram is extracted, then the color histograms are concatenated together as the whole image's descriptor. This descriptor is simple but has low performance for it doesn't consider the texture and pixel spatial distribution and is not robust to person with similar color but different texture. In [SDALF] the symmetry and asymmetry of foreground is considered. The maximally stable color region is also exploited in this paper. In [covariance] the covariance descriptors of local patches are used to represent people.

Given descriptors of individuals, a matching algorithm is needed to re-identify people. To compute similarity, the Mahanalobis distance based metric learning is very popular. Many works have been focusing on learning a Mahanalobis based matrix to improve performance. That is, suppose there are two d-dimensional vectors of two persons, $\boldsymbol{x}$,$\boldsymbol{y}$, a matrix M is used the calculate the distance $D(\boldsymbol{x},\boldsymbol{y}) = (\boldsymbol{x}-\boldsymbol{y})^T\boldsymbol{M}(\boldsymbol{x}-\boldsymbol{y})$. The matrix M can be learned by many limitations like LMNN[].

For descriptors with high dimension($d \geq 1000$), it's hard to directly learn a Mahanalobis distance matrix M for the small sample size $n(n << d)$. A popular method is to use principal component analysis(PCA) to reduce dimension. PCA is a very popular preprocessing method which uses an orthogonal transformation to convert a set of observations of possibly corre lated variables into a set of values of linearly uncorrelated variables called principal components. One problem is PCA doesn't consider the information between classes thus many discriminant information will be lost after dimension reduction. Linear discriminant analysis(LDA) is also used to reduce dimension. There are a few differences between LDA and PCA, LDA is a supervised method while PCA is unsupervised, and LDA is more used independently while PCA is more used as a preprocessing method. The advantage of LDA is it distinguishes different classes by maximize the ratio of inter-class scatter matrix versus intra-class scatter matrix. The local fisher discriminant analysis(LDFA) combines LDA and local projection preserving(LPP) to exploit the correlation of neighbor points of sample points. Moreover, the kernelization version of LFDA proposed to improve performance.

Instead of using KLFDA as a unique subspace learning method, in this paper, KLFDA is used to reduce dimension. Again with the lower dimension data, a Mahanalobis distance matrix M is learned based on the limitation that intra distance is at least 1 unit smaller than inter class distance.

This is inspired by the [TDL] paper. A target function respect M is created to penalize big intra class distance and small inter class distance. So we transformed re-ID into a optimization problem in the lower dimension space. With the target function, the gradient descent method is used to get a optimal M.

2. Related work

3. Hierarchical gaussian descriptor

The hierarchical gaussian descriptor is proposed by in [1], this descriptor uses a two-level gaussian distribution to model an individual. This descriptor densely sample the image and model each hierarchical structure with gaussian distribution. The combination of GOG and cross view quadratic analysis(XQDA) has outperformed many other works. Firstly it divides the image into a few overlapping horizontal slides, and in each slide, dense sampling patches are made with certain size. So there is a two-level structure in this image, small patches and slides. Then by model each level with gaussian model we can get a robust representation of the individual.

3.1. Single pixel modelling

In this hierarchical model, it is very important to have a full representation for every single pixel. To fully characterize single pixel, a d dimensional vector is used to represent it. In this vector, there could be any predefined properties like coordinates, color values, texture and filter response. Suppose the original image is in RGB color space, the gaussian of gaussian descriptor uses a 8-dimensional vector **f**, and

$$\boldsymbol{f}_i = (y, M_0, M_{90}, M_{180}, M_{270}, R, G, B)$$

. The y component is the y coordinate of pixel, and $M_{\{\theta\in 0^o, 90^o, 180^o, 270^o\}}$ is the quantized gradient information in 4 directions. The last three component is the color value is specified color space.

In all the benchmark dataset, all the images are cropped with a bounding box well suited the individual, and the pedestrian in an image can be at left or right of center, while in the vertical direction the head and feet of pedestrian is very close the image edge. So for each pixel, the y coordinate is more correlated than x coordinate.

Then the M is to characterize the texture with the gradient histogram. Different M values is the magnitude of gradient in every direction. Firstly the gradient intensity is computed as $G(x,y) = \{I_x, I_y\}$, and the orientation is $O(x,y) = \arctan(y/x)$. The magnitude values are quantized into four directions by a soft voting algorithm[GOG15]. For every gradient magnitude value with its orientation , the corresponding weights of all predefined directions are computed, and the direction with the biggest weight is chosen as the quantized direction for this pixel.

To model the patch with a multi-variate gaussian distribution, we have to estimate its mean value and the covariance matrix. A multi-variate gaussian model has the form

$$G(\boldsymbol{f}_i; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{\exp^{(\frac{1}{2}(\boldsymbol{f}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{f}-\boldsymbol{\mu}))}}{(2\pi)^{d/2}|\boldsymbol{\Sigma}|} \tag{1}$$

where $\boldsymbol{\mu}$ is the estimated mean value, and $\boldsymbol{\Sigma}$ is the estimated covariance matrix.

To estimate the parameters for this gaussian model, the maximal likelihood estimate is used. According MLE algorithm, we have the following estimated parameters

$$\begin{aligned} \boldsymbol{\mu} &= \frac{1}{n}\sum \boldsymbol{f}_i \\ \boldsymbol{\Sigma} &= \frac{1}{n}(\boldsymbol{f}_i - \boldsymbol{\mu})(\boldsymbol{f}_i - \boldsymbol{\mu})^T \end{aligned} \tag{2}$$

When the gaussian model is computed, the next step is to model all the patch gaussians. But it's a complex problem to directly model those gaussians. So some transformation will be operated on estimated parameters.

With the Gaussian parameters extracted in each region, the same transformation is operated on them. Then all horizontal slides' descriptor are concatenated to get the whole descriptor for the whole image.

3.2. Riemannian manifold based SPD transformation

As described before this hierarchical gaussian descriptor is a stochastic feature, so operations like computing mean and covariance need to be operated on previous summarized gaussian distributions. Mean and covariance operation in Euclidean space can not be directly operated on estimated mean vector and covariance matrix. A transformation is needed to make stochastic summarization feasible. In fact, the multivariate gaussian model is a Riemannian manifold and can be embedded into a semi positive definite matrix(SPD) space. The gaussian function is mapped into a vector space with two steps mapping. A d dimensional multivariate gaussian function can be mapped into a $d+1$ dimensional SPD_+ space. According to [GOG25], the mapping can be denoted as

$$G(\boldsymbol{x}_i; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \sim \boldsymbol{P}_i = |\boldsymbol{\Sigma}_i|^{1/(d+1)} \begin{bmatrix} \boldsymbol{\Sigma}_i + \boldsymbol{\mu}_i\boldsymbol{\mu}^T & \boldsymbol{\mu}_i \\ \boldsymbol{\mu}_i^T & 1 \end{bmatrix} \tag{3}$$

The covariance matrix $\boldsymbol{\Sigma}_i$ can be singular for small number of pixels within the patch, to avoid this problem a regular factor λ is added to $\boldsymbol{\Sigma}_i$ so that $\boldsymbol{\Sigma}_i = \boldsymbol{\Sigma}_i + \lambda \boldsymbol{I}$.

After this mapping, the $n+1$ dimensional SPD matrix needs to be transformed as a vector. The matrix logarithm is used to transform it to tangent space. A $d+1$ dimensional SPD matrix can be mapped as a $d*(d+3)/2+1$ vector, which can be denoted as $SPD_i^+ \sim \boldsymbol{p}_i = vec(log(\boldsymbol{P}_i))$. Since $\boldsymbol{P}_i$ is a positive symmetric matrix and it can be compressed by half that only the upper triangular elements are preserved. To ensure the sum of norm-1 remain the same after compression, the magnitude of off-diagonal elements in $\boldsymbol{P}_i$ are timed by $\sqrt{2}$. Let $\boldsymbol{Q} = \log \boldsymbol{P}_i$, we have

$$\begin{gathered}\boldsymbol{p}_i = [\boldsymbol{Q}_{1,1}, \sqrt{2}\boldsymbol{Q}_{1,2}, \sqrt{2}\boldsymbol{Q}_{1,3}, \cdots, \sqrt{2}\boldsymbol{Q}_{1,d+1}, \\ \boldsymbol{Q}_{2,2}, \sqrt{2}\boldsymbol{Q}_{2,3,}, \cdots, \sqrt{2}\boldsymbol{Q}_{2,d+1,}, \cdots, \boldsymbol{Q}_{d+1,d+1,}]\end{gathered} \tag{4}$$

Dimension analysis It has been shown in [ ] combination of descriptors of different color space can greatly improve re-ID performance. In this project, the hierarchical gaussian descriptor in RGB color space is the base descriptor. Descriptors in three more color space {HSV, Lab, nRGB}is extracted. The nRGB color space is calculated as $nR = \frac{R}{R+G+B}, nG = \frac{G}{R+G+B}, nR = \frac{B}{R+G+B}$, since nB can be calculated with nR and nG, in this color space only the first two channel values are used to reduce redundancy. Therefore, for color space {RGB,HSV,Lab,nRGB }, the corresponding dimension of pixel feature is {8,8,8,7}. After the matrix to vector transformation, the dimension of patch gaussian vector of each channel is {45,45,45,36}. Again after the patch gaussian to region gaussian transformation, the dimension of each channel is {1081,1081,1081,703}. Suppose there are 7 horizontal slides in each image, the dimension of concatenated descriptor of each channel is {7567,7567,7567,4921}. If four color space are all used, the dimension is the sum of each channel as 27622.

4. Dimension reduction based on kernel local fisher discriminant analysis

4.1. Background of kernel LFDA

In this paper the kernel local fisher discriminant analysis is adopted. This method is a combination of Fisher discriminant analysis[ ] and and the locality preserving projection in [LPP]. Fisher discriminant analysis is a supervised dimension reduction algorithm, whose input includes the original descriptors and the class labels. Here a brief review of Fisher linear analysis and LPP is given. For a set of d-dimensional observations $\boldsymbol{x}_i$, where $i \in \{1, 2, \cdots, n\}$, the label $l_i \in \{1, 2, \cdots, l\}$. Two matrix are defined as the within class scatter matrix $\boldsymbol{S}^{(w)}$ and between class matrix $\boldsymbol{S}^{(b)}$,

$$\begin{gathered}\boldsymbol{S}^{(w)} = \sum_{i=1}^{l} \sum_{j:l_j=i} (\boldsymbol{x}_j - \boldsymbol{\mu}_i)(\boldsymbol{x}_j - \boldsymbol{\mu}_i)^T \\ \boldsymbol{S}^{(b)} = \sum_{i=1}^{l} n_i(\boldsymbol{\mu}_i - \boldsymbol{\mu})(\boldsymbol{\mu}_i - \boldsymbol{\mu})^T\end{gathered} \tag{5}$$

where the $\boldsymbol{\mu}_i$ is the mean of samples whose label is i, and $\boldsymbol{\mu}$ is the mean of all samples,

$$\begin{gathered}\boldsymbol{\mu}_i = \frac{1}{n_i} \sum \boldsymbol{x}_i \\ \boldsymbol{\mu} = \frac{1}{n} \sum \boldsymbol{x}_i\end{gathered} \tag{6}$$

The Fisher Discriminant Analysis transform matrix $\boldsymbol{T}$ can be represented as

$$\boldsymbol{T} = \arg\max \frac{\boldsymbol{T}^T \boldsymbol{S}^{(b)} \boldsymbol{T}}{\boldsymbol{T}^T \boldsymbol{S}^{(w)} \boldsymbol{T}} \tag{7}$$

Fisher discriminant analysis tries to minimize the within-class distance while maximize the between class distance. The $\boldsymbol{T}$ is computed by the eigenvalue decomposition so that the between class scatter is maximized and the within class scatter matrix is minimized. $\boldsymbol{T}$ can be represented as the set of all the corresponding eigenvectors, as $\boldsymbol{T} = (\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \cdots, \boldsymbol{\phi}_k)$.

FDA analysis has a form similar with signal and noise ratio, however, the FDA dimension reduction may have poor performance for it doesn't consider the locality of data. In[Hexiaofei] locality preserving projection is proposed to exploit data locality. In LPP an affinity matrix is created to record the affinity of sample $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, typically the range of elements in $\boldsymbol{A}_{i,j}$ is $[0, 1]$. There are many manners to define a $n \times n$ affinity matrix $\boldsymbol{A}$, usually the two sample points with a smaller distance measured by Euclidean or other distance has a higher affinity value than those with bigger distance value. One of them is if $\boldsymbol{x}_i$ is within k-nearest neighbours of $\boldsymbol{x}_j$ then $\boldsymbol{A}_{i,j} = 1$ otherwise $\boldsymbol{A}_{i,j} = 0$. Another diagonal matrix D can be defined that each diagonal element is the sum of corresponding column in $\boldsymbol{A}$,

$$\boldsymbol{D}_{i,i} = \sum_{j=1}^{n} \boldsymbol{A}_{i,j} \tag{8}$$

then the LPP transform matrix is defined as follow,

$$\boldsymbol{T}_{LPP} = \mathop{\arg\min}_{\boldsymbol{T} \in \boldsymbol{R}^{d \times m}} \frac{1}{2} \sum_{i,j=1}^{n} \boldsymbol{A}_{i,j} ||\boldsymbol{T}^T \boldsymbol{x}_i - \boldsymbol{T}^T \boldsymbol{x}_j|| \tag{9}$$

so that $\boldsymbol{T}^T \boldsymbol{X} \boldsymbol{D} \boldsymbol{X}^T \boldsymbol{T} = \boldsymbol{I}$. Suppose the subspace has a dimension of m, then LPP transform matrix T can be represented as

$$\boldsymbol{T}_{LPP} = \{\boldsymbol{\phi}_{d-m+1} | \boldsymbol{\phi}_{d-m+1} | \cdots \boldsymbol{\phi}_d\}$$

And each $\boldsymbol{\phi}$ in T is the eigenvector of following fomula,

$$\boldsymbol{X} \boldsymbol{L} \boldsymbol{X}^T \boldsymbol{\phi} = \gamma \boldsymbol{X} \boldsymbol{D} \boldsymbol{X}^T \tag{10}$$

where γ is corresponding eigenvalue of $\boldsymbol{\phi}$, and $L = D - A$. But the LPP dimension reduction is still not discriminant enough, LFDA combines FDA and LPP and have a more strong performance. The key in LFDA is it assigns weights to elements in $\boldsymbol{A}^{(w)}$ and $\boldsymbol{A}^{(b)}$, so that,

$$\begin{gathered}\boldsymbol{S}^{(w)} = \frac{1}{2} \sum_{i=1}^{l} \sum_{j:l_j=i} \boldsymbol{A}^{w}_{i,j} (\boldsymbol{x}_j - \boldsymbol{\mu}_i)(\boldsymbol{x}_j - \boldsymbol{\mu}_i)^T \\ \boldsymbol{S}^{(b)} = \frac{1}{2} \sum_{i=1}^{l} \boldsymbol{A}^{b}_{i,j} (\boldsymbol{\mu}_i - \boldsymbol{\mu})(\boldsymbol{\mu}_i - \boldsymbol{\mu})^T\end{gathered} \tag{11}$$

where

$$\begin{gathered}\boldsymbol{A}^{(w)}_{i,j} = \left\{ \begin{array}{rl} \boldsymbol{A}_{i,j}/n_c & y_i = y_j \\ 0 & else \end{array} \right. \\ \boldsymbol{A}^{(b)}_{i,j} = \left\{ \begin{array}{rl} (\frac{1}{n} - \frac{1}{n_c})\boldsymbol{A}_{i,j} & y_i = y_j \\ \frac{1}{n} & else \end{array} \right.\end{gathered} \tag{12}$$

where y_i is the class label of sample point $\boldsymbol{x}_i$.

When applying the LFDA to original high dimensional descriptors, one problem is the computation cost. Suppose the vector data has a dimension of d, LFDA has to solve the eigenvalue a matrix with dimension $d \times d$. In some descriptors the d could be more than 20000 and thus the cost is not trivial. It may takes a few days to compute even on a computer with good configurations. For the huge complexity of LFDA, the kernel LFDA is introduced to shorten running time.

4.2. Kernel LFDA

Kernelization is proved to greatly improve performance in [ ] since the non-linearity is exploited. In [ ] it has been demonstrated that kernelization improves the performance of many dimension reduction and metric learning. Kernelization is a projection from low dimension to hight dimension, which may make classification and clustering much more accurate. The difference of kernel version LFDA is that the between class and within class scatter matrix will be transformed into kernel space and the eigenvalue decomposition will be operated on kernel matrix. Suppose a set of sample points $\boldsymbol{x}_i, i \in \{1, 2, \cdots, n\}$, can be mapped to a implicit higher feature space by a function $\phi(\boldsymbol{x}_i)$. It has been proved that kernel function can be implicit and only the inner product of mapped vectors $\phi\boldsymbol{x}_i$ and $\phi\boldsymbol{x}_j$ need to be known. The kernel trick is proposed to solve this problem by defining a function $k(\boldsymbol{x}_i, \boldsymbol{x}_j) =< \phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_j) >$, the $< \cdot >$ is the inner product. There are many kinds of kernel like linear kernel, polynomial kernel and radial basis function(RBF) kernel. In this paper the RBF kernel is adopted. A RBF kernel is defined as $k_{RBF}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp^{(-\gamma||\boldsymbol{x}_i - \boldsymbol{x}_j||^2)}$.

5. Mahalanobis metric learning on vectors with lower dimension

The Mahalanobis distance based metric learning has received much attention in similarity computing. The Mahalanobis distance of two observations $\boldsymbol{x}$ and $\boldsymbol{y}$ is defined as

$$D(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{x} - \boldsymbol{y})^T \boldsymbol{M} (\boldsymbol{x} - \boldsymbol{y}), \tag{13}$$

where $\boldsymbol{x}$ and $\boldsymbol{y}$are $d \times 1$ observation vectors, $\boldsymbol{M}$ is a positive-semidefinite matrix. Since $\boldsymbol{M}$ is positive-semidefinite, $\boldsymbol{M}$ can be decomposed as $\boldsymbol{M} = \boldsymbol{W}^T \boldsymbol{W}$, and Mahalanobis distance can also be written as

$$D(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{x} - \boldsymbol{y})^T \boldsymbol{W}^T \boldsymbol{W} (\boldsymbol{x} - \boldsymbol{y}) = ||\boldsymbol{W}(\boldsymbol{x} - \boldsymbol{y})|| \tag{14}$$

Therefore, Mahanalobis distance can be regarded as a variant of Euclidean distance. There are many methods proposed for metric learning[ ]. Inspired by [], in this paper, a similar metric learning based on iteration computation is used. For a sample descriptor $\boldsymbol{x}_i$, its positive pairwise set is defined as $\{\boldsymbol{x}_i, \boldsymbol{x}_j\}$, where class ID $y_i = y_j$. Also the negative pairwise set can be defined as $\{\boldsymbol{x}_i, \boldsymbol{x}_j\}$, where $y_i \neq y_j$. Similar with [PRDC], this method is also based on similarity comparison. The difference is in [PRDC], for all possible positive and negative pairs, the distance between positive pairs must be smaller than the distance between negative pairs. Since it has to compare possible positive and negative pairs, computation complexity will be quite huge. To decrease complexity, a simplified version is proposed as the top-push distance metric learning[ ]. Since re-identification is a problem of ranking, it is desired that the rank-1 descriptor should be the right match. Given a Mahanalobis matrix $\boldsymbol{M}$, for samples $\boldsymbol{x}_i, i = 1, 2, 3, \cdots, n$, n is the number of all samples, the requirement is distance between positive pair should be smaller than the minimum of all negative distance. This can be denoted as

$$D(\boldsymbol{x}_i, \boldsymbol{x}_j) + \rho < \min D(\boldsymbol{x}_i, \boldsymbol{x}_k), y_i = y_j, y_i \neq y_k. \tag{15}$$

ρ is a slack variable and $\rho \in [0, 1]$. This equation can be transformed into a optimization problem with respect to descriptor $\boldsymbol{x}_i$ as

$$\min \sum_{y_i = y_j} \max\{D(\boldsymbol{x}_i, \boldsymbol{x}_j) - \min_{y_i \neq y_k} D(\boldsymbol{x}_i, \boldsymbol{x}_k) + \rho\}. \tag{16}$$

However, the equation above only penalize the interclass distance. Another term is needed to penalize intra class distance. That is, to make the sum of intraclass distance as small as possible. This term is denoted as

$$\min \sum D(\boldsymbol{x}_i, \boldsymbol{x}_j), y_i = y_j. \tag{17}$$

To combine equations above, a ratio factor α is assigned to equation [] so that the target function can be denote as

$$\begin{aligned} f(\boldsymbol{M}) = (1 - \alpha) \sum_{\boldsymbol{x}_i, x_j, \boldsymbol{y}_i = y_j} D(\boldsymbol{x}_i, \boldsymbol{x}_j) + \\ \alpha \sum_{\boldsymbol{x}_i, \boldsymbol{x}_j, y_i = y_j} \max\{D(\boldsymbol{x}_i, \boldsymbol{x}_j) - \min_{y_i \neq y_k} D(\boldsymbol{x}_i, \boldsymbol{x}_k) + \rho, 0\} \end{aligned} \tag{18}$$

In this way the problem is transformed to an optimization problem. Notice that equation 16 can be denoted as

$$D(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{x} - \boldsymbol{y})^T \boldsymbol{M} (\boldsymbol{x} - \boldsymbol{y}) = Tr(\boldsymbol{M}\boldsymbol{X}_{i,j}) \tag{19}$$

where $\boldsymbol{X}_{i,j} = \boldsymbol{x}_i * \boldsymbol{x}_j^T$, and Tr is to compute matrix trace. Therefore, equation 21 can be transformed as follow,

$$\begin{aligned} \boldsymbol{G} = f(\boldsymbol{M}) = (1 - \alpha) \sum_{y_i = y_j} Tr(\boldsymbol{M}\boldsymbol{X}_{i,j}) \\ + \alpha \sum_{y_i = y_j, y_i \neq y_k} \max\{Tr(\boldsymbol{M}\boldsymbol{X}_{i,j}) - Tr(\boldsymbol{M}\boldsymbol{X}_{i,k}) + \rho, 0\} \end{aligned} \tag{20}$$

To minimize equation 23, the gradient descent method is used. The gradient respect to $\boldsymbol{M}$ is computed as

$$\frac{\partial f}{\partial \boldsymbol{M}} = (1 - \alpha) \sum_{y_i = y_j} \boldsymbol{X}_{i,j} + \alpha \sum_{y_i = y_j, y_i \neq y_k} (\boldsymbol{X}_{i,j} - \boldsymbol{X}_{i,k}) \tag{21}$$

The iteration process can be summarized as following

TABLE 1. Optimization algorithm on dimension reduced vectors

```
Gradient optimization algorithm for target function
Input Descriptors of training person pairs
Output A SPD matrix
Initialization
Initialize $\boldsymbol{M}$ with eye matrix $\boldsymbol{I}$;
Compute the initial target function value $f_0$ with $\boldsymbol{M}_0$;
Iteration count $t = 0$;
while(not converge)
Update $t = t + 1$;
Update gradient $\boldsymbol{G}_{t+1}$ with equation 24;
Update $\boldsymbol{M}$ with equation : $\boldsymbol{M}_{t+1} = \boldsymbol{M}_t - \lambda \boldsymbol{G}_t$
Project $\boldsymbol{M}_{t+1}$ to the positive semi-definite space
by $\boldsymbol{M}_{t+1} = \boldsymbol{V}_{t+1} \boldsymbol{S}_{t+1} \boldsymbol{V}_{t+1}^T$;
Update the target value $f|_{\boldsymbol{M}=\boldsymbol{M}_{t+1}}$;
end while
return $\boldsymbol{M}$
```

6. Experiment

6.1. Datasets and evaluation settings

VIPeR VIPeR dataset is the most used dataset in person re-identification. In this dataset there are 632 different individuals and for each person there are two outdoor images from different viewpoints. All the images are scaled into 48×128. In this experiment the we randomly select 316 individuals from cam a and cam b as the training set, the rest images in cam a are used as probe images and those in cam b as gallery images. This process is repeated 10 times to reduce error.

CUHK1 CUHK01 dataset contains 971 identities from two disjoint camera views. The cameras are static in each pair of view and images are listed in the same order. For each individual, there are two images in each view. All images are scaled into 60×160. In this paper, we randomly select 485 image pairs as training data and the rest person pairs are used for test data.

Prid_2011 The dataset consists of images extracted from multiple person trajectories recorded from two different, static surveillance cameras. Images from these cameras contain a viewpoint change and a stark difference in illumination, background and camera characteristics.Camera view A shows 385 persons, camera view B shows 749 persons. The first 200 persons appear in both camera views, The remaining persons in each camera view complete the gallery set of the corresponding view. Hence, a typical evaluation consists of searching the 200 first persons of one camera view in all persons of the other view. This means that there are two possible evaluation procedures, either the probe set is drawn from view A and the gallery set is drawn from view B. In this paper, we randomly select 100 persons that appeared in both camera views as training pairs, and the remaining 100 persons of the 200 person pairs from camera a is used as probe set while the 649 remaining persons from camera B are used for gallery images.

Prid_450s The PRID 450S dataset contains 450 image pairs recorded from two different, static surveillance cameras. Additionally, the dataset also provides an automatically generated, motion based foreground/background segmentation as well as a manual segmentation of parts of a person. The images are stored in two folders that represent the two camera views. Besides the original images , the folders also contain binary masks obtained from motion segmentation, and manually segmented masks. In this test, we randomly select 225 persons from each of two camera views as the training set, and the remaining persons are left as gallery and probe images.

GRID There are two camera views in this dataset. Folder probe contains 250 probe images captured in one view (file names starts from 0001 to 0250). Folder gallery contains 250 true match images of the probes (file names starts from 0001 to 0250). Besides, in gallery folder there are a total of 775 additional images that do not belong to any of the probes (file name starts with 0000). These extra images should be treated as a fixed portion in the testing set during cross validation. In this paper, we randomly select 125 persons from those 250 persons appeared in both camera views as training pairs, and the remaining persons in probe folder is used as probe images while the remaining 125 persons and those 775 additional persons from gallery folder are used as gallery images.

TABLE 2. Testing setting for different datasets

Dataset	training	probe	gallery	cam_a	cam_b
VIPeR	316	316	316	632	632
CUHK1	485	486	486	971	971
PRID_2011	100	100	649	385	749
PRID_450s	225	225	225	450	450
GRID	125	125	900	250	1025

6.2. The influence of mean removal and L_2 normalization

In [GOG], mean removal and L_2 normalization is found to improve performance by 5.1%. The reason for this is mean removal and normalization can reduce the impact of extremas of descriptors. When testing proposed metric learning, we find the mean removal can slightly improve performance. A comparison between performance of original descriptors and preprocessed descriptors is shown in Tables [2,3,4,5,6], all those datasets are tested by proposed metric. The original GOG means no mean removal and normalization. It shows that the mean removal and normalization has a slight improvement around 0.5% on the performance on all five datasets. Since preprocessing are required to test XQDA, the mean removal and normalization are operated on descriptors in this experiment.

TABLE 3. The influence of data preprocessing on VIPeR

	Rank(%)				
Terms	1	5	10	15	20
Original GOG	43.32	74.78	85.00	89.94	93.39
Preprocessed GOGrgb	43.73	74.75	85.41	90.28	93.86
Original GOGfusion	48.67	77.41	87.41	91.65	94.34
Preprocessed GOGfusion	48.10	76.90	87.59	91.90	94.40

TABLE 4. THE INFLUENCE OF DATA PREPROCESSING ON CUHK1

	Rank(%)				
Terms	1	5	10	15	20
Original GOGrgb	56.15	83.79	90.08	92.63	94.26
Preprocessed GOGrgb	55.86	84.28	90.45	93.09	94.65
Original GOGfusion	57.00	84.55	90.37	92.82	94.69
Preprocessed GOGfusion	56.69	84.40	90.53	93.27	94.90

TABLE 5. THE INFLUENCE OF DATA PREPROCESSING ON PRID_2011

	Rank(%)				
Terms	1	5	10	15	20
Original GOGrgb	24.70	51.80	63.30	69.60	72.70
Preprocessed GOGrgb	23.80	52.10	63.50	70.20	73.50
Original GOGfusion	32.40	56.80	66.80	73.10	77.70
Preprocessed GOGfusion	32.20	57.50	66.40	73.50	78.00

TABLE 6. THE INFLUENCE OF DATA PREPROCESSING ON PRID_450S

	Rank(%)				
Terms	1	5	10	15	20
Original GOGrgb	61.02	84.22	91.33	94.09	96.22
Preprocessed GOGrgb	60.44	84.44	91.33	94.00	96.13
Original GOGfusion	62.89	86.62	92.53	95.29	96.89
Preprocessed GOGfusion	62.62	86.44	92.36	95.20	96.93

TABLE 7. THE INFLUENCE OF DATA PREPROCESSING ON GRID

	Rank(%)				
Terms	1	5	10	15	20
Original GOGrgb	22.96	42.00	51.76	58.72	64.64
Preprocessed GOGrgb	22.80	43.76	52.08	59.04	65.12
Original GOGfusion	24.32	44.40	54.96	62.40	66.56
Preprocessed GOGfusion	23.84	44.64	55.04	62.24	66.24

6.3. Parameters setting of gradient descent iteration

In this experiment, there are a few parameters for the iteration computing including slack variable ρ, maximal iteration T, gradient step λ, the inter and intra class limitation factor α and the updating ratio β. Firstly the slack variable ρ is initialized as 1 to ensure the minimum inter class distance is 1 larger than intra class distance at least. The step size of gradient updating λ is initialized as 0.01. When target value f increases, λ is scaled by a factor 0.5, and λ is scaled by 1.01 when target value f decreases. To judge if target value converges, the thresh β is defined as the ratio target value change versus previous target value, that is, $\beta = \frac{(f_{t+1}-f_t)}{f_t}$. According many experiment trials, when it satisfies $\beta = 10^{-5}$, the target value converges and the iteration is stopped. The maximal iteration times is set to 100 since the target value f will converge in around 15 iterations. The last parameter for the iteration is α, to know the best value for α, we tried 11 different values ranges from 0 to 1 with a step of 0.1, and the rank-1 and rank-5 scores of responding α is shown in figure []. The best α value should have as large top rank scores as possible. By comparison, α is set as 0.7. A form of all parameters are shown in Form 7.

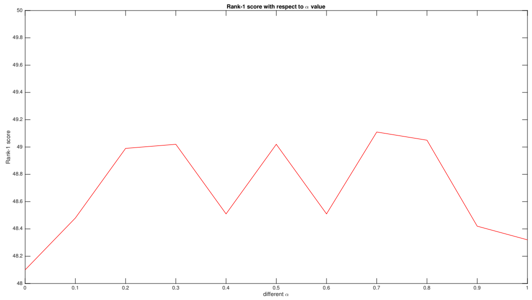

Figure 1. Rank 1 scores with respect to α on VIPeR

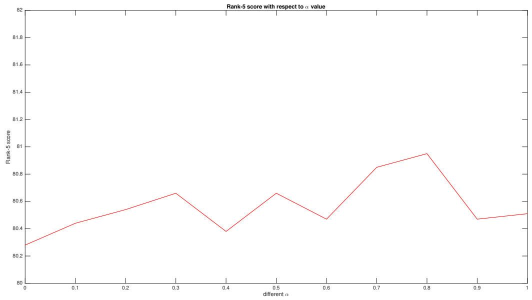

Figure 2. Rank 5 scores with respect to α on VIPeR

TABLE 8. PARAMETERS SETTING

Paramters	α	thresh	step	Max iteration	slack variable
Values	0.7	10^{-5}	0.01	100	1

Performance measuring The cumulative matching curve is used to measure the descriptor performance. The score means the probability that the right match is within the top n samples. A perfect CMC curve is expected to have a high rank-1 value and reaches 1 as fast as possible.

6.4. Performance analysis

In this paper, we compare proposed metric with other state-of-the-art metrics including NFST[], XQDA[]. NFST is a metric which learn a null space for descriptors so that the the same class descriptors will be projected to a single point to minimize within class scatter matrix while different classes are projected to different points. This metric is a good solution to small sample problems in person re-identification. XQDA is quite similar with many other metrics, which learns a projection matrix W and then a Mahanalobis SPD matrix M is learned in the subspace. Those two metric are proved to have state-of-the-art performance with many other methods. The GOGrgb in all forms stands for the hierarchical gaussian descriptor in RGB color space while GOGfusion stands for the one in four different color spaces {RGB,Lab,HSV,nRnR}.

VIPeR A comparison form is given in Table 8. Some of recent results are also included in this form. We can

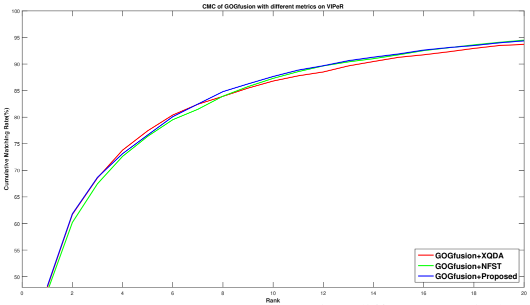

Figure 3. CMC curves on VIPeR comparing different metric learning

find that the rank scores are better than those of NFST and XQDA in terms of both GOGrgb and GOGfusion. More specifically, the rank 1, rank 5, rank 10, rank 15 and rank 20 GOGrgb scores of proposed metric learning are 0.44%, 0.72%, 1.27%, 1.3%, 1.47% higher than those of GOGrgb+XQDA, and the rank-1, rank-5, rank-10, rank-15 and rank-20 GOGfusion scores of proposed metric learning are 0.19%, -0.79%, 0.86%, 0.63%, 0.67% higher than GOGfusion + XQDA respectively. Also we can see that the proposed metric learning has a way more better performance than NFST. We can infer that the performance of KLFDA is better than XQDA, with its rank-1 score 1.6% higher.

TABLE 9. PERFORMANCE OF DIFFERENT METRICS ON VIPeR

	Rank(%)				
Methods	1	5	10	15	20
GOGrgb+NFST	43.23	73.16	83.64	89.59	92.88
GOGrgb+XQDA	43.01	73.92	83.86	89.24	92.37
GOGrgb+Proposed	43.48	74.59	85.35	90.47	93.67
GOGfusion+NFST	47.15	76.39	87.31	91.74	94.49
GOGfusion+XQDA	47.97	77.44	86.80	91.27	93.70
GOGfusion+Proposed	48.16	76.65	87.66	91.90	94.37

CUHK1 We can find that the rank 1, rank5, rank 10, rank 15, rank 20 score of GOGrgb combined with proposed metric are 5.35%, 4.22%,3.35%,2.1%,1.44% higher than XQDA, and 0.26%,1.26%,1.38%,1.11%, 1.09% than NFST. Also the rank 1, rank5, rank 10, rank 15, rank 20 score of GOGfusion combined with proposed metric are 4.59%, 2.55%, 0.72%, 1.29%, 0.89% higher than GOGfusion combined with XQDA, and 0.4%, 0.74%, 0.6%, 1.05%, 1.2% than GOGfusion combined with NFST.

TABLE 10. PERFORMANCE OF DIFFERENT METRICS ON CUHK1

	Rank(%)				
Methods	1	5	10	15	20
GOGrgb+NFST	55.60	83.02	89.07	91.98	93.56
GOGrgb+XQDA	50.51	80.06	87.10	90.99	93.21
GOGrgb+Proposed	55.86	84.28	90.45	93.09	94.65
GOGfusion+NFST	56.26	83.66	89.63	92.22	93.70
GOGfusion+XQDA	52.10	81.85	88.81	91.98	94.01
GOGfusion+Proposed	56.69	84.40	90.53	93.27	94.90

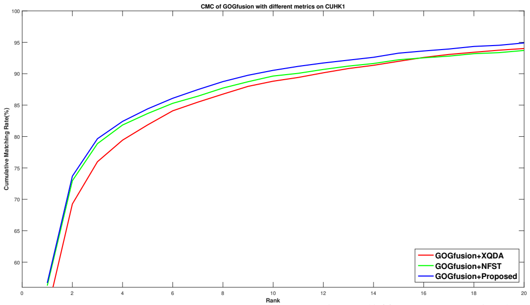

Figure 4. CMC curves on CUHK1 comparing different metric learning

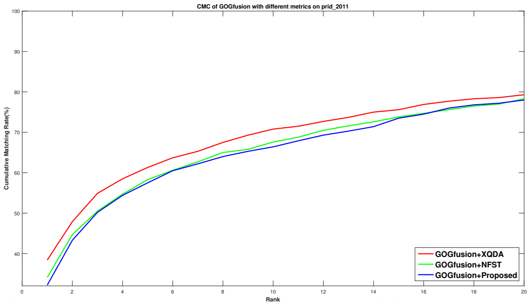

Figure 5. CMC curves on prid_2011 comparing different metric learning

TABLE 11. PERFORMANCE OF DIFFERENT METRICS ON PRID_2011

	Rank(%)				
Methods	1	5	10	15	20
GOGrgb+NFST	26.60	53.80	62.90	71.30	75.40
GOGrgb+XQDA	31.10	55.70	66.10	72.40	76.10
GOGrgb+Proposed	23.80	52.10	63.50	70.20	73.50
GOGfusion+NFST	34.10	58.30	67.60	73.80	78.30
GOGfusion+XQDA	38.40	61.30	70.80	75.60	79.30
GOGfusion+Proposed	32.20	57.50	66.40	73.50	78.00

Prid_2011 The rank 1, rank5, rank 10, rank 15, rank 20 score of GOGfusion combined with proposed metric are 6.2%, 3.8%, 4.4%, 2.1% and 1.3% lower than GOGfusion combined with XQDA. The performance of NFST is slightly better than proposed metric. Also in terms of GOGrgb XQDA and NFST has better performance than the proposed one. So in this dataset the proposed metric has worse performance than XQDA and NFST.

TABLE 12. PERFORMANCE OF DIFFERENT METRICS ON PRID_450S

	Rank(%)				
Methods	1	5	10	15	20
GOGrgb+NFST	61.96	84.98	90.53	94.09	96.09
GOGrgb+XQDA	65.29	85.02	91.13	94.76	96.49
GOGrgb+Proposed	60.44	84.44	91.33	94.00	96.13
GOGfusion+NFST	64.53	86.62	92.93	95.78	97.42
GOGfusion+XQDA	68.40	87.42	93.47	95.69	97.02
GOGfusion+Proposed	62.62	86.44	92.36	95.20	96.93

Prid_450s In this dataset, we can find the rank 1 score of XQDA and NFST is higher than proposed metric, but they have almost the same rank 5, rank 10, rank 15, and rank 20 scores with respect to both descriptors.

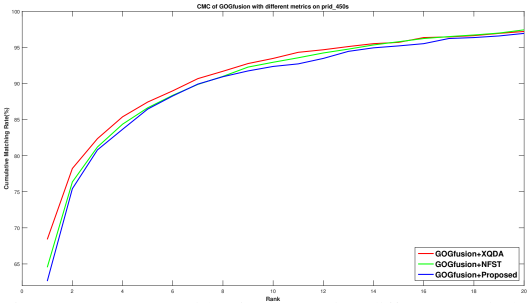

Figure 6. CMC curves on prid_450s comparing different metric learning

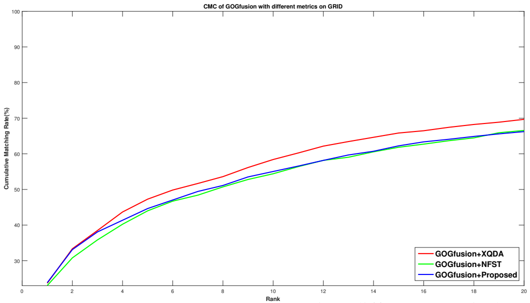

Figure 7. CMC curves on GRID comparing different metric learning

TABLE 13. PERFORMANCE OF DIFFERENT METRICS ON GRID

	Rank(%)				
Methods	1	5	10	15	20
GOGrgb+NFST	21.84	41.28	50.96	57.44	62.88
GOGrgb+XQDA	22.64	43.92	55.12	61.12	66.56
GOGrgb+Proposed	22.80	43.76	52.08	59.04	65.12
GOGfusion+NFST	23.04	44.40	54.40	61.84	66.56
GOGfusion+XQDA	23.68	47.28	58.40	65.84	69.68
GOGfusion+Proposed	23.84	44.64	55.04	62.24	66.24

GRID We can see that the rank 1 score of proposed metric are slightly higher than XQDA and 0.8% higher than NFST in terms of GOGfusion, but XQDA outperforms proposed metric on rank 5, rank 10, rank 15 and rank 20 scores. But proposed metric outperforms NFST on rank 5, rank 10, rank 15 and rank 20 scores.

7. Conclusion

In this paper a SPD matrix is learned on the lower dimension space after dimension reduction by kernel local fisher discriminative analysis. By analysis we can find the proposed metric has better performance than NFST and XQDA on VIPeR and CUHK1 datasets, but XQDA and NFST outperforms the proposed metric learning on Prid_2011 and Prid_450s, and the proposed metric learning has better rank 1 score than NFST and XQDA on GRID dataset.

References

[1] “Hierarchical Gaussian Descriptor for Person Re-Identification,” pp. 1–10, Dec. 2016.