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 has received increasing attention in recent years. The task of re-ID is to judge if two individuals in two images from the same or different cameras are the same person. This problem is very challenging caused by many factors like low image resolution, occlusion, background noise and different camera color response, etc. According to how many images are provides for each individual, 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 multi-shots case, there might exist quite much difference even in intra class for different pose and illuminations. Therefore, good descriptors are supposed to be robust to illumination change and occlusions.

There are two main directions for re-ID, descriptors extraction and metric learning. Most of previous literature try to improve performance from those two aspects. It's important to select a proper descriptor for images. The most used descriptors are color based descriptors. According to the extraction range descriptors can be classified as local and global descriptors[ ]. One simple descriptor is the histogram descriptor. 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 and has low performance for it doesn't consider the texture and pixel spatial distribution.

Since only using color information is not sufficient to distinguish person with similar color, most of previous descriptors consist of combination of color and texture models within local or global range, like local binary pattern, gradient patterns, one order or two order derivative in horizontal or vertical direction.

In [], the covariance descriptor is used to combine color and texture information. In each patch of interest, the covariance of each pixel's property like pixel values and gradient information is computed and the the average of all pixels' covariance is computed to cancel the effect of noise and misalignment.

After the descriptors are extracted, the next question is how to match those descriptors. Matching descriptors is to compute the similarity between descriptors. There are many straightforward distance computing methods like Euclidean distance, Bhattacharya distance and Mahanalobis distance. Those methods are simple but have low performance. So many methods dealing with the descriptors are proposed. Since the many descriptors have very high dimensionality, dimensionality reducing are used to reduce computation complexity.PCA is a very popular preprocessing method to reduce the dimensionality which uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components.

Linear discriminant analysis is also used to reduce vector data dimensionality. 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. However, it might be insufficient only considering the linearity of data. Kernel method is proved to improve the re-ID performance since it considers the nonlinearity of vector data. The kernel linear discriminant analysis(KLFDA) is proposed for dimension reduction.

Besides those methods above, 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 $D(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{x} - \boldsymbol{y})^T \boldsymbol{M}(\boldsymbol{x} - \boldsymbol{y})$. There are many proposed methods to compute the Mahalanobis matrix, one of them is the gradient descent. Also, there are methods like XQDA, and LMNN.

Besides, in [prdc], the author proposed to learn the Mahanalobis matrix with all the possible positive and negative sample pairs so that the distance between the positive pairs are smaller than those distance of negative pairs. Since the matrix $\boldsymbol{M}$ is a semi-positive defined(SPD) matrix, that is, $\boldsymbol{M} = W^T W$, the learning of M can be transfered to learning W. In [TDL] the author uses a very concise but effective model to learn the M. This model requires that

In [] the author proposes to use a nullspace transformation to transform descriptors so that after the transformation the intra-class data are mapped to a single point while those inter-class vectors are mapped to different points. This method is has achieve the highest performance in VIPeR dataset so far. In [] the convolutional neural network is used to improve the performance. One problem of convolutional neural network based re-ID is the small sample size problem. Many datasets used in re-ID , like VIPeR dataset, have small image size for model training and thus the model is poorly trained.

There are also many papers dealing with hierachical models. In [hierachical Covariance papers] the hierachical covariance model In [GOG paper], the hierachical gaussian model was proposed to represent people and has outperformed many other paper with the metric learning of XQDA.

Besides, since there have been few work handle the gaussian mixture model based descriptor, thus few metric learning methods dealing with GMM model have been proposed. This

In this paper, a variant of the hierarchical gaussian of gaussian descriptors[ ] was proposed. There are two biggest difference between our new descriptor and the original GOG descriptor.

There are two levels in the original GOG, firstly it extract the pixel feature in the basic rectangular patches with gaussian distribution. Secondly, it model the overlapping patches with another gaussian distribution. But in our descriptor, firstly the image is segmented into many superpixels, and we model model,

2. Related work

2.1. Appearance descriptors

Previous work focus on find more discriminative descriptors and better metric learning. A good descriptor is robust to problems like illumination, low resolution and viewpoint, etc. To model the complex human kinematics, the part-based models are most adopted since human body is non-rigid body. Previous literature mainly contains three kinds of models [1], fixed part models, adaptive part models and the learned part models. The fixed part models are used in [2,3,4], where a silhouette is divided into a fixed number of horizontal and equal stripes, which mainly include head, torso, legs. In [9] the width of each stripe are respectively 16%,29% and 55%. The fixed models predefine the parameters like numbers of stripes and the stripe width.

In the adaptive part models, the models vary from one to one according to predefined algorithm. Take [6] for an instance, the silhouette of each person is divided into three parts horizontally, which include the head, torso and legs respectively. But the width of each stripe is different for various silhouettes, and it is computed according to the symmetry and asymmetry with two operators $C(y, \sigma)$and $S(y, \sigma)$, where

$$C(y, \sigma) = \sum d^2(p_i - \hat{p}_i)$$

$$S(y, \sigma) = \sum \frac{1}{W\delta} |A(B[y, y - \delta]) - A(B[y, y + \delta])|$$

Here the $C(y, \sigma)$ computes the asymmetry of two blobs and $S(y, \sigma)$ computes the difference of two areas. Then the axis between torso and legs are computed as follow

$$y_{TL} = \arg\min(1 - C(y, \sigma) + S(y, \sigma))$$

and the axis between head and torso is computed with following equation,

$$y_{HT} = \arg\min(-S(y, \sigma))$$

the axis divides the left and right torso is

$$j_{LR} = \arg\min(C(y, \sigma) + S(Y, \sigma))$$

With those equations above axis can be computed depend on specific image. This method has a relatively high performance.

The part-based adaptive spatial-temporal model used in [7] characterizes person?s appearance using color and facial feature. Few work exploits human face feature but in this work human face selection based on low resolution cues selects useful face images to build face models. Color features capture representative color as well as the color distribution to build color model. This model handles multi-shots re-identification and it also model the color distribution variation of many consecutive frames. Besides, the facial features of this model is conditional, that is, in the absence of good face images this model is only based on color features.

Some methods based on learned part models have been proposed. Part model detectors, that is, statistic classifiers, are trained with manually labelled human body parts images, exploiting features related with edges contained in the images. The pictorial structure is proposed in [8], and a PS model of a non-rigid body is a collection of part models with deformable configurations and connections with certain parts. The appearance of each part is separately modelled and deformable configurations are implemented with spring-like connections. This model can quantitatively describe visual appearance and model the non-rigid body. In [8] the body model is made up of N parts and N corresponding part detectors. Suppose $L = (l_0, l_1, \ldots, _{N-1})$ be the possible configurations of each part, where l_i is the state of the i-th body part and $l_0 = (x_i, y_i, \theta_i, s_i)$, x_i and y_i are the coordinates of the part center, θ_i is the absolute part orientation, and s_i is the scale size relative to the part size in the training set. Given the image evidence D, the problem is to maximize the posterior probability $P(D)$ that the part configuration is correct, and we have $P(D) \propto P(L) * P(L)$, where $P(L)$ is the likelihood of image evidence given a particular part configuration and P(L) corresponds to a kinematic prior, and those two items can be learned from a training set.

Another example of learned part model is in [12,13], the overall human body model consists of several part models, each model is made up of a spatial model and a part filter. For each part the spatial model define allowed arrangements of this part with respect to the bounding box. To train each model the Latent Support Vector Machine is used and in [12,13] four body parts are detected, namely head, left and right torso and upper legs. Compared with other models this model exploits a sequence of frames of an individual and thus captures appearance characteristics as well as the appearance variation over time.

Moreover, 3-D model is proposed to improve re-ID performance. A new 3-D model model called SARC3D [16] is used to represent the individual. Compared with those 2-D models, this model combines the texture and color information with their location information together to get a 3D model. This model starts with an approximate body model with single shape parameter. By precise 3-D mapping this parameter can be learned and trained with even few images (even one image is feasible). This model?s construction is driven by the frontal, top and side views extracted from various videos, and for each view the silhouette of people is extracted to construct the 3-D graphical model. The final body model is sampled to get a set of vertices from previously learned graphic body model. Compared with other model, this model has a robust performance when dealing with partial occlusion, people pose and viewpoint variations since the model is based on people silhouettes from three viewpoints.

As for the feature for each model (a whole model or part-based model), the feature can be implemented with different methods. The features can be divided into two categories, the global and local feature. The global feature refers to the feature extracted from a whole image or region, and the size of the descriptor is usually fixed. While to extract the local feature of a specified image or region, we first divide the whole image into many equal blocks and compute the feature of each block. Both descriptors may deal with color, texture and shape. The color is exploited most as the color histogram within different color space. descriptor based on texture, such as the SIFT, SURF and LBP are also widely combined to improve the performance.

Global color histogram is a frequently used global feature. For an three-channel image, like RGB image, each channel is quantized into B bins separately. The final histogram could be a multi-dimensional or mono-dimensional histogram. For instance, if $B = 8$, for multi-dimensional histogram there will be $8 * 8 * 8 = 512$ bins, but if we concatenate the 3 dimensional bins together the dimension can be reduced to $8+8+8 = 24$ bins while the performance of this reduced descriptor doesn't decrease. This method can be applied on other color spaces like HSV and Lab, etc.

Local color histogram usually splits the specified model or region into many equal size blocks and compute the global feature of each block. The feature can be based on color, texture and interest points. SIFT[14] is a kind of local feature based on the interest points. The salient interest points (identifiable over rotating and scaling) are selected by the interest operator. This algorithm detects the scale-extrema of the function difference of Gaussian function with scale σ, that is,

$$D(x, y, \sigma) = (G(x, y, k_1\sigma) - G(x, y, k_2\sigma)) * I(x, y)$$

the Gaussian filter with standard deviation $k1 * \sigma$, $I(x, y)$ is the image, when the DoG image is computed, each point (x, y) is compared with its neighbours to get extrema.

MSCR (maximally stable color region) is also used in [6]. The MSCR is derived from MSER (maximally stable extreme region), it detects the region with stable color and uses an agglomerative clustering algorithm to compute color clusters, by looking at the successive time steps of the algorithm the extension of color is implemented. The detected color region is described with a nine dimensional vector containing the area, averaging color, centroid and second moment matrix. With this vector the color region detected is easy to do scale and affine transforms.

RHSP (Recurrent Highly-Structured Patches) is used in [6]. This feature captures patches with highly recurrent color and texture characteristics from extracted silhouette pixels. This feature is extracted with following steps, first random and probably overlapping small image patches are extracted from silhouette pixels. Then to capture those patches with informative texture the entropy of each patch (the sum of three channels? entropy) is computed, we discard those patches with entropy smaller than a specified threshold. In the next step some transforms are performed on the remaining patches to select those remain invariant to the transforms. Subsequently, the recurrence of each patch is evaluated with the LNCC(local normalized cross correlation) function. This evaluation is only performed on small region containing the patch instead of the whole image. Then the patches with high recurrence is clustered to avoid patches with similar content. Finally, the Gaussian cluster is applied to maintain the patch nearest to cluster?s centroid for each cluster.

Combined descriptors are found to have better performance. Descriptors combining color and texture are most often used in re-identification. In [5] a signature called asymmetry-based histogram plus epitome(AHPE) was proposed. This work starts with a selection of images to reduce image redundancy (redundancy is caused by correlated consecutive sequences). This descriptor combines global and local statistical descriptors of human appearance, focusing on overall chromatic content via histogram and on the recurrent local patches via epitome analysis [6]. Similar to SDALF descriptor [6], HPE descriptor consists of three components, the chromatic color histogram, the generic epitome and local epitome. The chromatic color histogram is extracted in the HSV color space, which turns to be robust to illumination changes. Here color histogram is encoded into a 36-dimensional feature space $[H = 16, S = 16, V = 4]$. Besides, the authors customize the use of epitome here by extracting generic and local epitome here.

2.2. Metric learning

The second step of re-ID is to design the similarity computing methodology to compare descriptors. That is, the way to compare how different two descriptors are. This is also call the metric learning. Generally, for two input vectors x_1, x_2, any symmetric positive demi-definite matrix W defines a pseudo-metric with the form of $D = x_1 * W * x_2$. Many widely used distance metric obey this rule. Previous methods includes the Euclidean distance, Bhattacharyya distance and Mahalanobis distance. The Euclidean distance, which is mostly used for the straightforward descriptors like color and texture descriptors, is one case of the L_p distance when $p = 2$, and also a special case of Mahalanobis distance when the covariance matrix of two input observations is an identity matrix. One example of metric learning is the probabilistic relative distance comparison model proposed in [4]. This model decreases the error caused by sometimes high intra-class variation and low inter-class variation. Compared with other distance learning models proposed, this model behaves more robust for its probability relative distance comparison model. Suppose z is an image of a person, the task is to identify another image z' of the same person from z'' of a different person by using a distance model $f(.,.)$ so that $f(z,z') < f(z,z'')$. The contribution of this paper is that the author transfers the distance learning problem into a probability comparison problem by measuring the probability of distance between a relevant pair of images being smaller than that of a related irrelevant pair as

$$P(f(z,z') < f(z,z'')) = (1 + e^{(f(z-z')-f(z-z''))})^{-1} \tag{1}$$

Here the author assumes the probability of $f(z,z')$ and $f(z,z'')$ is independent, therefore, using maximal likelihood principal the optimal function can be learned as

$$\begin{aligned} f &= \arg\min_f r(f,O) \\ r(f,O) &= -log(\Pi_{O_i} P(f(z-z') - f(z-z''))) \end{aligned} \tag{2}$$

$O = \{O_i = (x_i^p - x_i^n)\}$, x_i^p,x_i^n are the pair from same person and different person respectively. The distance function $f(\cdot)$ here is parameterized as Mahalanobis distance function $f = \overrightarrow{x}^T \mathbf{M} \overrightarrow{x}, \mathbf{M} \geq 0$, here $\mathbf{M}$ is a semi-definite matrix, in this way the distance function learning problem is transformed to a matrix learning problem. The author used an iteration algorithm to compute matrix M.

Since still image based person representation suffers from factors like illumination, occlusion, viewpoint change and pose difference. The multi-shot re-ID has been proposed. Since there are a sequence of images for each indivudual, there are much more cues to exploit. In [TDL], the author simplified computing of Mahananobis matrix by applying the new limitations on datasets. The author finds that when using video based person representation the difference of inter-class may be more obscure than that of still image based representation. Therefore, the author proposed the top-push distance learning. For a person video sequence, the maximal intra-class distance should be smaller than the minimal distance of inter-class distance. One another requirement is the sum of all intra-class distance should be as small as possible, so the final target function is summarized as

$$\begin{aligned} f(D) = (1-\alpha) \sum_{x_i,x_j,y_i=y_j} D(x_i,x_j) + \\ \alpha \sum_{x_i,x_j,y_i=y_j} \max\{D(x_i,x_j) - \min_{y_i \neq y_k} D(x_i,x_k) + \rho, 0\} \end{aligned} \tag{3}$$

Cross view quadratic analysis(XQDA) is proposed in [xqda paper]. In [36 of lomo paper] Mogaddam et al. proposed to model each of two classes with a multivariate Gaussian distribution. Suppose the sample difference $\Delta = x_i - x_j$, where x_i and x_j are two feature vectors. Δ is called intrapersonal difference when their label $y_i = y_j$ and extrapersonal difference when $y_i \neq y_j$. Respectively two the intrapersonal and interpersonal variation can be defined as Ω_I and Ω_E,

Besides those aforementioned metric learning, some metric learning by neural network also draws much interest. That is, to define a neural network to compute similarity of two input descriptors or even images. Recently the deep neural network has been exploited to improve the performance. One advantage of neural network re-ID is the preprocessing of images can be skipped (We can also say the preprocess is included in convolutional layers). The input of this structure can be straight-forward grey images or color images. To deal with multi-shots and video based re-identification neural network is proven to have better performance. But for the classical neural network there are too many weights to train and the over-fitting problem can be troublesome. Convolutional neural network can avoid those problems while remains high performance. Compared with classical neural network architecture, the convolutional neural network exploits receptive field, weights sharing and pooling technology to reduce weights number and thus decreases computational cost. In [11] for the first time the author proposes a recurrent neural network layer and temporal pooling to combine all time-steps data to generate a feature vector of the video sequence. In [17] the author proposes a multi-channel layers based neural network to jointly learn both local body parts and whole body information from input person images. In [18] a convolutional neural network learning deep feature representations from multiple domains is proposed, and this work also proposes a domain guided dropout algorithm to dropout CNN weights when learning from different datasets. This method gives a solution to the problem that most CNNs are not trained enough caused from datasets with small number of images since this CNN learns from multiple datasets.

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 and 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

$$G(\boldsymbol{f}_i; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{\exp^{(\frac{1}{2}(\boldsymbol{f}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{f}-\boldsymbol{\mu}))}}{(2\pi)^{d/2}|\boldsymbol{\Sigma}|} \tag{4}$$

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

$$\begin{aligned} \boldsymbol{\mu} &= \frac{1}{n}\sum \boldsymbol{f}_i \\ \boldsymbol{\Sigma} &= \frac{1}{n}(\boldsymbol{f}_i - \boldsymbol{\mu})(\boldsymbol{f}_i - \boldsymbol{\mu})^T \end{aligned} \tag{5}$$

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 finished on previous estimated gaussian functions. A transformation is needed to make stochastic summarization feasible on previous level function. 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

$$G(\boldsymbol{x}_i; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \sim \boldsymbol{P}_i = |\boldsymbol{\Sigma}_i|^{1/(d+1)} \begin{bmatrix} \boldsymbol{\Sigma}_i + \boldsymbol{\mu}_i \boldsymbol{\mu}^T & \boldsymbol{\mu}_i \\ \boldsymbol{\mu}_i^T & 1 \end{bmatrix} \tag{6}$$

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

$$\begin{aligned} \boldsymbol{p}_i = [&\boldsymbol{Q}_{1,1}, \sqrt{2}\boldsymbol{Q}_{1,2}, \sqrt{2}\boldsymbol{Q}_{1,3}, \cdots, \sqrt{2}\boldsymbol{Q}_{1,d+1}, \\ &\boldsymbol{Q}_{2,2}, \sqrt{2}\boldsymbol{Q}_{2,3,}, \cdots, \sqrt{2}\boldsymbol{Q}_{2,d+1,}, \cdots, \boldsymbol{Q}_{d+1,d+1,}] \end{aligned} \tag{7}$$

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

The original descriptor has a high dimension and dimensionality disaster will be brought about if the original high-dimensional descriptor is used to learn a metric matrix. One popular method is to reduce the high dimension then learn a metric matrix in the lower dimension data. Among those methods to reduce dimension, principal component analysis is often used. However, PCA is a unsupervised dimension reduction and may have a low performance for those reasons, (1), PCA is to maximize the variance of dimension reduced data, and as a unsupervised method it doesn't has a full consideration of the the relation of between and within classes, it is very likely that the descriptors of different classes can be mixed up in the dimension reduction; (2) PCA may suffer from the small sample size problem. In some re-ID datasets, there may be two or less images for each pedestrian in each viewpoint (like VIPeR), if the dimension of descriptor is much bigger than sample size, much information can be lost with PCA.

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 $\boldsymbol{S}^{(b)}$,

$$\begin{aligned} \boldsymbol{S}^{(w)} &= \sum_{i=1}^{l} \sum_{j:l_j=i} (\boldsymbol{x}_j - \boldsymbol{\mu}_i)(\boldsymbol{x}_j - \boldsymbol{\mu}_i)^T \\ \boldsymbol{S}^{(b)} &= \sum_{i=1}^{l} n_i (\boldsymbol{\mu}_i - \boldsymbol{\mu})(\boldsymbol{\mu}_i - \boldsymbol{\mu})^T \end{aligned} \tag{8}$$

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

$$\begin{aligned} \boldsymbol{\mu}_i &= \frac{1}{n_i} \sum \boldsymbol{x}_i \\ \boldsymbol{\mu} &= \frac{1}{n} \sum \boldsymbol{x}_i \end{aligned} \tag{9}$$

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

$$\boldsymbol{T} = \arg\max \frac{\boldsymbol{T}^T \boldsymbol{S}^{(b)} \boldsymbol{T}}{\boldsymbol{T}^T \boldsymbol{S}^{(w)} \boldsymbol{T}} \tag{10}$$

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 $\boldsymbol{A}$,

$$\boldsymbol{D}_{i,i} = \sum_{j=1}^{n} \boldsymbol{A}_{i,j} \tag{11}$$

then the LPP transform matrix is defined as follow,

$$\boldsymbol{T}_{LPP} = \underset{\boldsymbol{T} \in \boldsymbol{R}^{d \times m}}{\arg\min} \frac{1}{2} \sum_{i,j=1}^{n} \boldsymbol{A}_{i,j} ||\boldsymbol{T}^T \boldsymbol{x}_i - \boldsymbol{T}^T \boldsymbol{x}_j|| \tag{12}$$

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,

$$\boldsymbol{X} \boldsymbol{L} \boldsymbol{X}^T \boldsymbol{\phi} = \gamma \boldsymbol{X} \boldsymbol{D} \boldsymbol{X}^T \tag{13}$$

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,

$$\begin{aligned}\boldsymbol{S}^{(w)} &= \frac{1}{2}\sum_{i=1}^{l}\sum_{j:l_j=i}\boldsymbol{A}^{w}_{i,j}(\boldsymbol{x}_j-\boldsymbol{\mu}_i)(\boldsymbol{x}_j-\boldsymbol{\mu}_i)^T \\ \boldsymbol{S}^{(b)} &= \frac{1}{2}\sum_{i=1}^{l}\boldsymbol{A}^{b}_{i,j}(\boldsymbol{\mu}_i-\boldsymbol{\mu})(\boldsymbol{\mu}_i-\boldsymbol{\mu})^T\end{aligned} \tag{14}$$

where

$$\begin{aligned}\boldsymbol{A}^{(w)}_{i,j} &= \begin{cases}\boldsymbol{A}_{i,j}/n_c & y_i=y_j\\ 0 & else\end{cases}\\ \boldsymbol{A}^{(b)}_{i,j} &= \begin{cases}(\frac{1}{n}-\frac{1}{n_c})\boldsymbol{A}_{i,j} & y_i=y_j\\ \frac{1}{n} & else\end{cases}\end{aligned} \tag{15}$$

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

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

$$D(\boldsymbol{x},\boldsymbol{y}) = (\boldsymbol{x}-\boldsymbol{y})^T\boldsymbol{M}(\boldsymbol{x}-\boldsymbol{y}), \tag{16}$$

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 Mahanalobis distance can also be written as

$$D(\boldsymbol{x},\boldsymbol{y}) = (\boldsymbol{x}-\boldsymbol{y})^T\boldsymbol{W}^T\boldsymbol{W}(\boldsymbol{x}-\boldsymbol{y}) = ||\boldsymbol{W}(\boldsymbol{x}-\boldsymbol{y})|| \tag{17}$$

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

$$D(\boldsymbol{x}_i,\boldsymbol{x}_j) + \rho < \min D(\boldsymbol{x}_i,\boldsymbol{x}_k), y_i = y_j, y_i \neq y_k. \tag{18}$$

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

$$\min \sum_{y_i=y_j} \max\{D(\boldsymbol{x}_i,\boldsymbol{x}_j) - \min_{y_i\neq y_k} D(\boldsymbol{x}_i,\boldsymbol{x}_k) + \rho\}. \tag{19}$$

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

$$\min \sum D(\boldsymbol{x}_i,\boldsymbol{x}_j), y_i = y_j. \tag{20}$$

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

$$\begin{aligned} f(\boldsymbol{M}) = (1-\alpha) \sum_{\boldsymbol{x}_i,x_j,\boldsymbol{y}_i=y_j} D(\boldsymbol{x}_i,\boldsymbol{x}_j) + \\ \alpha \sum_{\boldsymbol{x}_i,\boldsymbol{x}_j,y_i=y_j} \max\{D(\boldsymbol{x}_i,\boldsymbol{x}_j) - \min_{y_i\neq y_k} D(\boldsymbol{x}_i,\boldsymbol{x}_k) + \rho, 0\}\end{aligned} \tag{21}$$

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

$$D(\boldsymbol{x},\boldsymbol{y}) = (\boldsymbol{x}-\boldsymbol{y})^T\boldsymbol{M}(\boldsymbol{x}-\boldsymbol{y}) = trace(\boldsymbol{M}\boldsymbol{X}_{i,j}) \tag{22}$$

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

$$\boldsymbol{G} = f(\boldsymbol{M}) = (1-\alpha)\sum_{y_i=y_j} trace(\boldsymbol{M}\boldsymbol{X}_{i,j}) + \alpha \sum_{y_i=y_j, y_i\neq y_k} \max\{trace(\boldsymbol{M}\boldsymbol{X}_{i,j}) - trace(\boldsymbol{M}\boldsymbol{X}_{i,k}) + \rho, 0\} \tag{23}$$

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

$$\frac{\partial f}{\partial \boldsymbol{M}} = (1-\alpha)\sum_{y_i=y_j} \boldsymbol{X}_{i,j} + \alpha \sum_{y_i=y_j, y_i\neq y_k} (\boldsymbol{X}_{i,j} - \boldsymbol{X}_{i,k}) \tag{24}$$

The iteration process can be summarized as following

```
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.

TABLE 1. 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

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.

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 2. 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 3. 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 4. 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 5. 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 6. 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

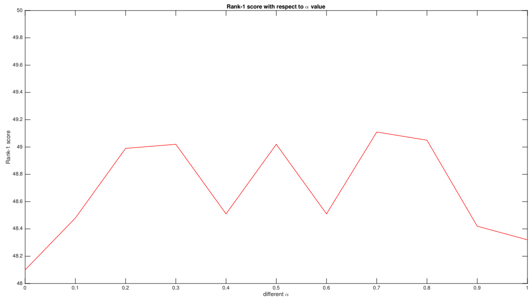

Figure 1. Rank 1 scores with respect to α on VIPeR

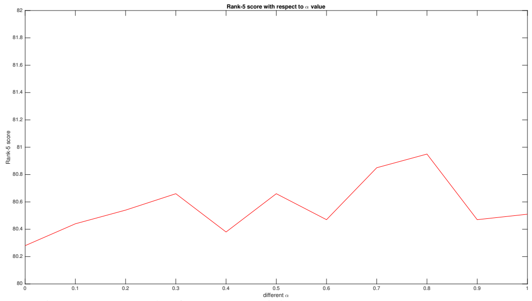

Figure 2. Rank 5 scores with respect to α on VIPeR

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.

TABLE 7. 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 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 8. 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.

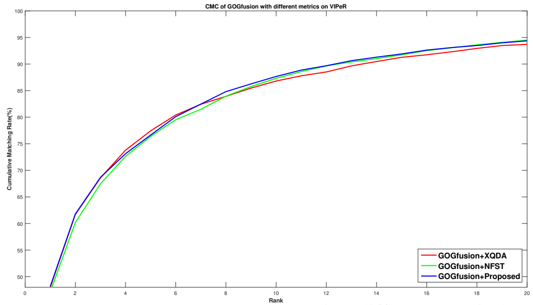

Figure 3. CMC curves on VIPeR comparing different metric learning

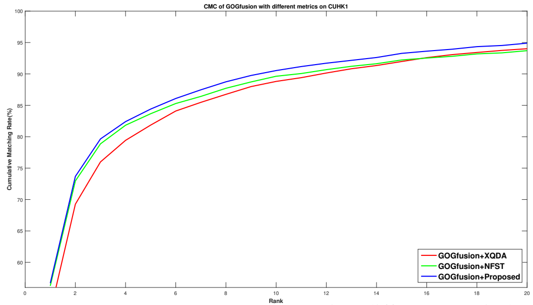

Figure 4. CMC curves on CUHK1 comparing different metric learning

TABLE 9. 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

TABLE 10. 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.

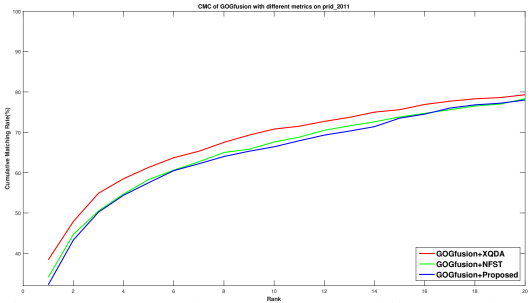

Figure 5. CMC curves on prid_2011 comparing different metric learning

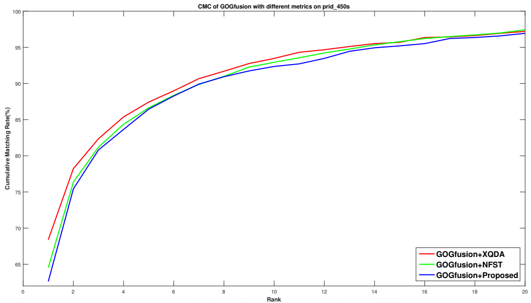

Figure 6. CMC curves on prid_450s comparing different metric learning

TABLE 11. 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.

TABLE 12. 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.

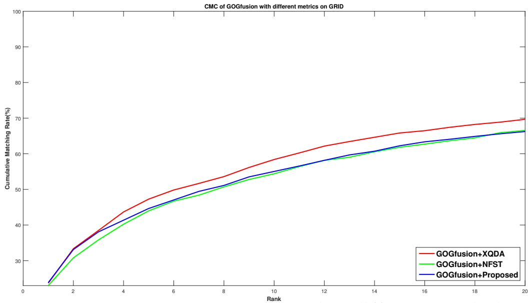

Figure 7. CMC curves on GRID comparing different metric learning

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.