# Person Re-identification Based on Kernel Local Fisher Discriminant Analysis and Mahalanobis Distance Learning

By Qiangsen He

*A thesis submitted to the Faculty of Graduate and Postdoctoral Studies in fulfillment*
*of the requirements for the degree of Master of Applied Science*
*in the*
*School of Electrical Engineering and Computer Science*
*Faculty of Engineering*
*University of Ottawa*

April, 2017

University of Ottawa

# *Abstract*

Faculty of Engineering
School of Electrical Engineering and Computer Science

Master of Applied Science

**Person Re-identification Based on Kernel Local Fisher Discriminant Analysis and Mahalanobis Distance Learning**

by Qiangsen He

Person re-identification (Re-ID) has become an intense research area in recent years. The main goal of this topic is to recognize and match individuals over time at the same or different locations. This task is challenging due to the variation of illumination, viewpoints, pedestrians' appearance and partial occlusion. Previous works mainly focus on finding robust features and metric learning. Many metric learning methods convert the Re-ID problem to a matrix decomposition problem by Fisher discriminant analysis (FDA). Mahalanobis distance metric learning is a popular method to measure similarity; however, since directly extracted descriptors usually have high dimensionality, it's intractable to learn a high-dimensional semi-positive definite (SPD) matrix. Dimensionality reduction is used to project high-dimensional descriptors to a lower-dimensional space while preserving those discriminative information. In this paper, the kernel Fisher discriminant analysis (KLFDA) [38] is used to reduce dimensionality given that kernelization method can greatly improve Re-ID performance for nonlinearity. Inspired by [47], an SPD matrix is then learned on lower-dimensional descriptors based on the limitation that the maximum intraclass distance is at least one unit smaller than the minimum interclass distance. This method is proved to have excellent performance compared with other advanced metric learning.

# *Acknowledgements*

First, I would like to express my sincere gratitude to my supervisor, Professor Robert Laganiere, for the continuous support of my master study and research and for his immense knowledge, patience and enlightenment. His guidance helped me throughout my research and writing of this thesis. I would also like to thank Professor Jiying Zhao and Professor Andy Adler for being my thesis examiners.

My sincere thanks also goes to my friends, Di Pang, Dongfeng Gu, Xuelu Wang, Binhao Wang, Muye Jiang and other members of the VIVA lab, for all the fun we have had in the last two years, and for your kind help and encouragement. I would like to thank Lu Sun and Andres Solis Montero, the PhD candidates in the VIVA lab, for giving me suggestions and tips in image segmentation. I would also like to thank Yong Wang, the post doctoral fellow in our lab, for his kind revision suggestions and encouragement.

Last but not the least, I would like to thank my parents and sister for supporting me throughout the study in Canada and the writing of this thesis.

# Contents

Abstract ii

Acknowledgements iii

1 Introduction 1
1.1 Basic concepts . . . . . . . . . . . . . . . . . . . . . . . . . . 3
1.2 Challenges . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6
1.3 My work in this thesis . . . . . . . . . . . . . . . . . . . . . . . 7
1.4 Contributions . . . . . . . . . . . . . . . . . . . . . . . . . . . 9
1.5 Thesis organization . . . . . . . . . . . . . . . . . . . . . . . . 10

2 Related work 11
2.1 Appearance descriptors . . . . . . . . . . . . . . . . . . . . . . 11
2.2 Metric learning . . . . . . . . . . . . . . . . . . . . . . . . . . 15
2.3 Other methods for Re-ID . . . . . . . . . . . . . . . . . . . . . 18
2.4 Some state-of-the-art works . . . . . . . . . . . . . . . . . . . . 18
2.5 Performance measuring . . . . . . . . . . . . . . . . . . . . . . 20

3 Extraction of hierarchical Gaussian descriptors 23
3.1 Basic color and textural features . . . . . . . . . . . . . . . . . 23
3.1.1 Color histogram descriptors on different color space . . . . 23
3.1.2 Local binary pattern (LBP) . . . . . . . . . . . . . . . . . 25
3.1.3 Histogram of oriented gradients (HOG) . . . . . . . . . . . 25
3.2 Influence of background segmentation on different basic descriptors 27
3.3 The hierarchical Gaussian descriptor . . . . . . . . . . . . . . . 30
3.3.1 Handling the background . . . . . . . . . . . . . . . . . . 31
3.3.2 Single pixel modelling . . . . . . . . . . . . . . . . . . . 31
3.3.3 Riemannian manifold based SPD transformation . . . . . . 33
3.3.4 Integral image for fast region covariance computation . . . . 34

3.3.5 Some variants of Hierarchical Gaussian descriptor . . . . . 36
3.3.6 Dimensionality and superiority analysis of hierarchical Gaussian descriptor . . . . . . . . . . . . . . . . . . . . . . . . 38

4 **Dimension reduction and Mahalanobis distance learning** **40**
4.1 Kernel local Fisher discriminant analysis . . . . . . . . . . . . . . . 40
4.1.1 Fisher discriminant analysis (FDA) . . . . . . . . . . . . . . 41
4.1.2 Locality preserving projection (LPP) . . . . . . . . . . . . . . 42
4.1.3 Local Fisher discriminant analysis (LFDA) . . . . . . . . . . 43
4.1.4 Kernel local Fisher discriminant analysis (KLFDA) . . . . . 44
4.2 Mahalanobis distance . . . . . . . . . . . . . . . . . . . . . . . . . 48
4.3 Gradient descent optimization . . . . . . . . . . . . . . . . . . . . 48
4.4 Metric learning based on sample pairs distance comparison . . . . . 49

5 **Experiment settings** **52**
5.1 Datasets and evaluation settings . . . . . . . . . . . . . . . . . . . 53
5.2 The influence of mean removal and $L_2$ normalization . . . . . . . . 55
5.3 Parameters setting . . . . . . . . . . . . . . . . . . . . . . . . . . 57
5.4 Results and experiment analysis . . . . . . . . . . . . . . . . . . . 59
5.5 Comparison with other methods . . . . . . . . . . . . . . . . . . . 64
5.6 The Image size and time cost . . . . . . . . . . . . . . . . . . . . . 64

6 **Conclusion** **66**
6.1 Contributions . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 66
6.2 Future work . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 67
6.2.1 Improve the hierarchical Gaussian descriptor . . . . . . . . 67
6.2.2 Improve foreground segmentation algorithm . . . . . . . . . 67
6.2.3 Decrease the computational cost of gradient descent method 67

**Bibliography** **69**

# List of Figures

1.1 Re-ID workflow . . . 2

1.2 A typical single-shot Re-ID workflow . . . 2

1.3 The VIPeR dataset . . . 5

1.4 Samples from the prid_2011 dataset . . . 5

1.5 VIPeR foreground . . . 6

1.6 The workflow of proposed work; the left part is training and the right part is testing . . . 8

1.7 Samples from the prid_2011 dataset . . . 9

2.1 A CMC plot of gallery size is $M = 10$ and probe size is $N = 12$ . . . 22

3.2 A CMC comparison of color histogram on different color spaces . . . 24

3.1 RGB and HSV visual comparison; the first row is RGB and second row is HSV for the same views . . . 24

3.3 A comparison of two patches with the same entropy but different color distribution . . . 25

3.4 LBP: by thresholding the neighbour pixels, the pixels are transformed into a binary number . . . 25

3.5 One LBP example . . . 26

3.6 A demo of the HOG feature with a cell size of six pixels . . . 27

3.7 Foreground segmentation of individuals from VIPeR . . . 28

3.8 A CMC comparison of foreground segmentation on HSV histogram descriptor tested on VIPeR . . . 29

3.9 A CMC comparison of foreground segmentation on LBP feature tested on VIPeR . . . 29

3.10 A CMC comparison of foreground segmentation on HOG feature tested on VIPeR . . . 30

3.11 Integral image . . . 35

3.12 A SLIC superpixel segmentation example . . . . . . . . . . . . . . . 37

4.1 Example of five clusters that belong to three classes . . . . . . . . . 46
4.2 1D distribution of dimension-reduced data with Gaussian kernel . . 47
4.3 1D distribution of dimension-reduced data with linear kernel . . . . 47
4.4 Steepest gradient descent . . . . . . . . . . . . . . . . . . . . . . 49

5.1 Pedestrians in prid_450 dataset . . . . . . . . . . . . . . . . . . . . 53
5.2 Pedestrians in prid_450s dataset . . . . . . . . . . . . . . . . . . . 54
5.3 Pedestrians in GRID dataset . . . . . . . . . . . . . . . . . . . . . 54
5.4 Rank 1 scores with respect to $\alpha$ on VIPeR . . . . . . . . . . . . . . 58
5.5 Rank 5 scores with respect to $\alpha$ on VIPeR . . . . . . . . . . . . . . 58
5.6 CMCs on VIPeR comparing different metric learning . . . . . . . . 60
5.7 CMCs on CUHK1 comparing different metric learning . . . . . . . 61
5.8 CMCs on prid_2011 comparing different metric learning . . . . . . 62
5.9 CMCs on prid_450s comparing different metric learning . . . . . . 63
5.10 CMCs on GRID comparing different metric learning . . . . . . . . 63

# List of Tables

2.1 A CMC example of gallery size is $M = 10$ and probe size is $N = 12$ 22

3.1 A rank score comparison of GOG$_{\text{fusion}}$ Variants . . . . . . . . . . . 36
3.2 A performance comparison between Gaussian of Gaussian descriptor and its variants on the VIPeR dataset . . . . . . . . . . . . . . . . . 38

4.1 Optimization algorithm of Mahalanobis distance matrix learning . . 51

5.1 Testing setting for different datasets . . . . . . . . . . . . . . . . . 55
5.2 Original image size of pedestrians from different datasets . . . . . . 55
5.3 The influence of data preprocessing on VIPeR . . . . . . . . . . . . 56
5.4 The influence of data preprocessing on CUHK1 . . . . . . . . . . . 56
5.5 The influence of data preprocessing on prid_2011 . . . . . . . . . . 56
5.6 The influence of data preprocessing on prid_450s . . . . . . . . . . 56
5.7 The influence of data preprocessing on GRID . . . . . . . . . . . . 57
5.8 Parameters setting . . . . . . . . . . . . . . . . . . . . . . . . . . . 58
5.9 Performance of different metrics on VIPeR . . . . . . . . . . . . . . 59
5.10 Performance of different metrics on CUHK1 . . . . . . . . . . . . . 60
5.11 Performance of different metrics on prid_2011 . . . . . . . . . . . . 61
5.12 Performance of different metrics on prid_450s . . . . . . . . . . . . 62
5.13 Performance of different metrics on GRID . . . . . . . . . . . . . . 63

# Chapter 1

# Introduction

People re-identification (Re-ID) has been an intense research topic in recent years, the main goal of which is to match an image of a given person with other images of the same person. Person Re-ID has great potential in video surveillance, target detection and tracking, and forensic search. In an intelligent surveillance system, suppose there are many cameras and these cameras may not have overlapping field of view, the system has to automatically identify individuals that appear in different cameras. It is quite challenging since the matching accuracy is greatly influenced by many factors like occlusion, illumination variation, camera settings and color response. Surveillance system like this can be applied to forensic search, domotics applications and distributed multi-camera surveillance system to track individuals across different cameras without overlapping field of view over time.

In Re-ID, those images with known labels are called gallery images, and the image used to determine its label is called a probe image. The probe image and gallery images can be from the same or different camera views, so the viewpoint and illumination between the probe and gallery image can be quite different. Also, because of the different color response of different cameras, the images of the same person may look different in different cameras. Occlusions between the camera and target can also bring about much difficulty. In a word, images of the same person may look different while images of different persons may look quite the same. In Re-ID, the main goal is to improve the matching accuracy.

Given a sequence or video of individuals, there are three steps to match a person. A simple workflow is shown in Figure 1. However, since most Re-ID datasets are already cropped manually or by an automatic detector, most Re-ID work will only focus on robust descriptors design and efficient matching algorithm aimed at those well-cropped images.

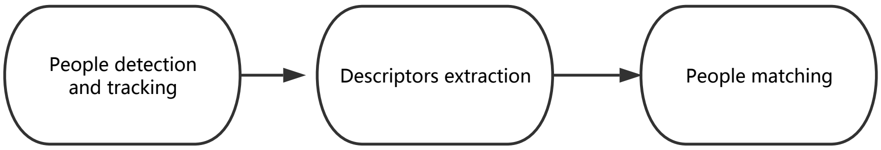

FIGURE 1.1: Re-ID workflow

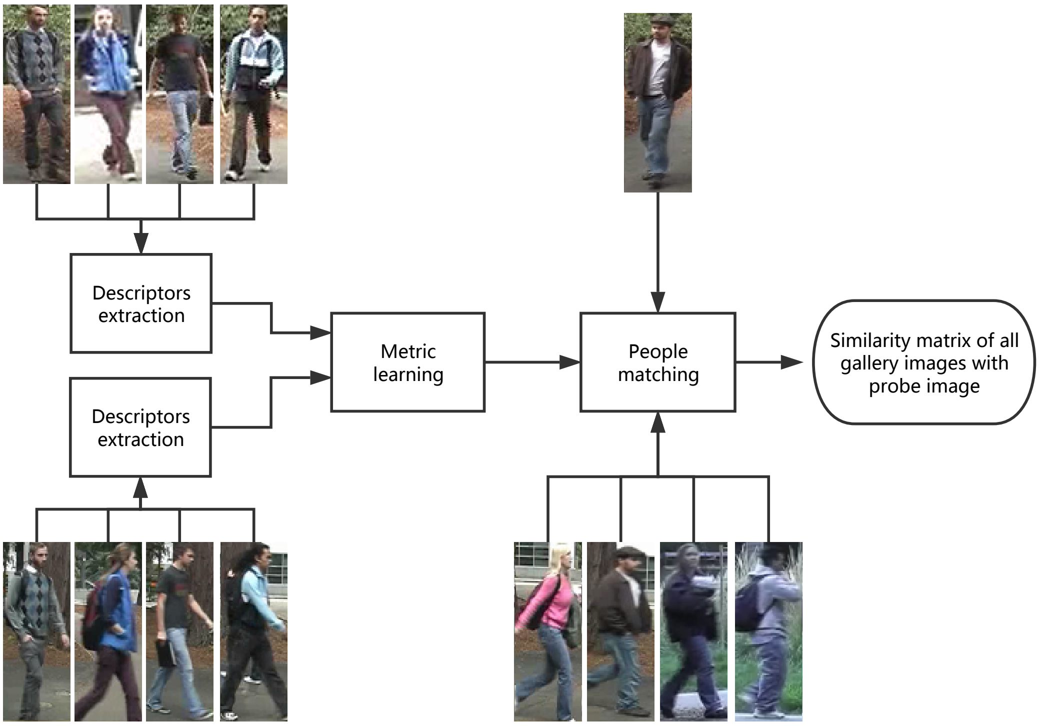

FIGURE 1.2: A typical single-shot Re-ID workflow

The first task in Re-ID is to design a robust descriptor to represent images. The descriptor is supposed to contain the key information for each captured person. Basically, the descriptors are supposed to be robust and discriminative. One straightforward way is to extract the color, textural information of images and then use descriptors to compute the similarity scores. But this method turns out to be not robust to the illumination variation, camera color response difference and camera angle settings. Therefore, many other advanced descriptors take into account the correlation of color, texture and position together to improve performance.

The second is to design the similarity computing methodology (i.e., compare the similarity of two descriptors), which is also called distance metric learning. Metric learning can be divided into supervised and unsupervised distance metric learning [45]. The supervised distance metric requires the labels of class while the unsupervised one doesn't. Many previous supervised metric learning methods try to utilize different distance constraints to learn a Mahalanobis metric which has a form as $D(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{x} - \boldsymbol{y})^T \boldsymbol{M}(\boldsymbol{x} - \boldsymbol{y})$. For example, in [48] the constraint is to project all descriptors of the same class to the same point while projecting descriptors of different classes to different points in a kernel space. In [47] the constraint is the maximal intraclass distance should be at least one unit smaller than the minimal interclass distance. In [51] the constraint is that the distance of negative class pair should be smaller than the positive class pairs. Different distance constraints leads to different semi-positive definite (SPD) matrix $\boldsymbol{M}$ to be learned so that $\boldsymbol{M}$ satisfies predefined constraints.

**Person detection, person re-identification and face recognition** There are a few differences between the three concepts. Person detection or pedestrian detection is to detect the presence of human body in an area of space. Person re-identification is to judge if two individuals detected in the same or different cameras are the same person. In person re-identification the person detection work must have been finished before attempting people re-identification. Face recognition is to identify a person by comparing its facial feature with a database of facial features. Face recognition is very similar with person re-identification, they all need to compare an input feature with a database of features to identify the input. However, face recognition utilizes only facial features, while person re-identification takes advantage of the whole appearance features (facial feature may be used if the image resolution is high enough). In many surveillance systems, due to the low resolution and viewpoint change, the human faces are not visible or very blurry. In this case face recognition can not be used.

## 1.1 Basic concepts

People re-identification can be divided into a few categories according to different conditions. Some general concepts are listed below.

**Open-set and closed-set Re-ID** [6] According to the gallery size and how the gallery size evolves, Re-ID can be divided into open-set Re-ID and closed-set Re-ID. In closed-set Re-ID, no new identities will be added to the gallery set, and gallery size remains the same as time goes by. Furthermore, the probe set will be a subset of the gallery set. That means, the number of unique identities in the gallery set will be equal or greater than the probe set. In open-set Re-ID, the gallery set will evolve as time goes by. Each time a probe image is entered into the recognition system, the system will judge if it has a corresponding match in the gallery set. If the probe image doesn't match any of the gallery images, it will be regarded as a new identity and will be added to the gallery set. The probe set is not necessarily the subset of the gallery set.

**Long-term and short-term Re-ID** According to the time interval between gallery and probe images, Re-ID can be divided into long-term and short-term Re-ID. In short-term Re-ID the time interval between gallery and probe images is small, say a few minutes or several hours. Conversely, long-term Re-ID refers to the case that the time interval between gallery and probe images is a few days or even longer. One difference to consider regarding the long time interval between gallery and probe images is the variation of individuals' clothing and appearance. If the gallery images were shot a few days prior, the same individual may have changed his suit or taken off his bag, causing the appearance to change. In this case, it will be much more difficult to recognize the same identity in long-term Re-ID. In most cases, we use short-term Re-ID, which guarantees that the appearance of the same person will remain the same and we will only need to consider the differences brought by other factors such as viewpoint variation and occlusions.

**Single-shot and multi-shot Re-ID** According to the size of the sample set for each person, Re-ID can be divided into single-shot and multi-shot approaches. In single-shot Re-ID, only one image is provided for a person in a camera view. Single-shot Re-ID is challenging because only limited information can be extracted. One example is the VIPeR dataset shown in Figure 1.3. For each person in this dataset only one image is provided in each camera view, and the viewpoint of each view is different. In multi-shot Re-ID, a sequence of images is provided for a person in a camera view. Compared with single-shot Re-ID, more information, like temporal-spatial information, can be extracted from the sample set. One example of multi-shot dataset is the prid_2011 dataset shown in Figure 1.4, which provides a long sequence of images for each person in a single camera view.

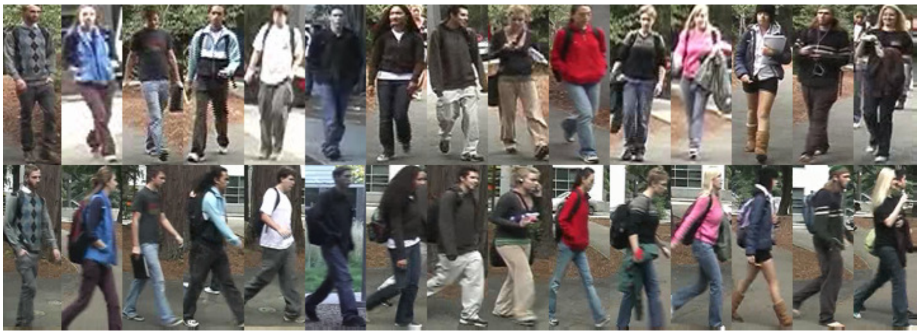

FIGURE 1.3: The VIPeR dataset

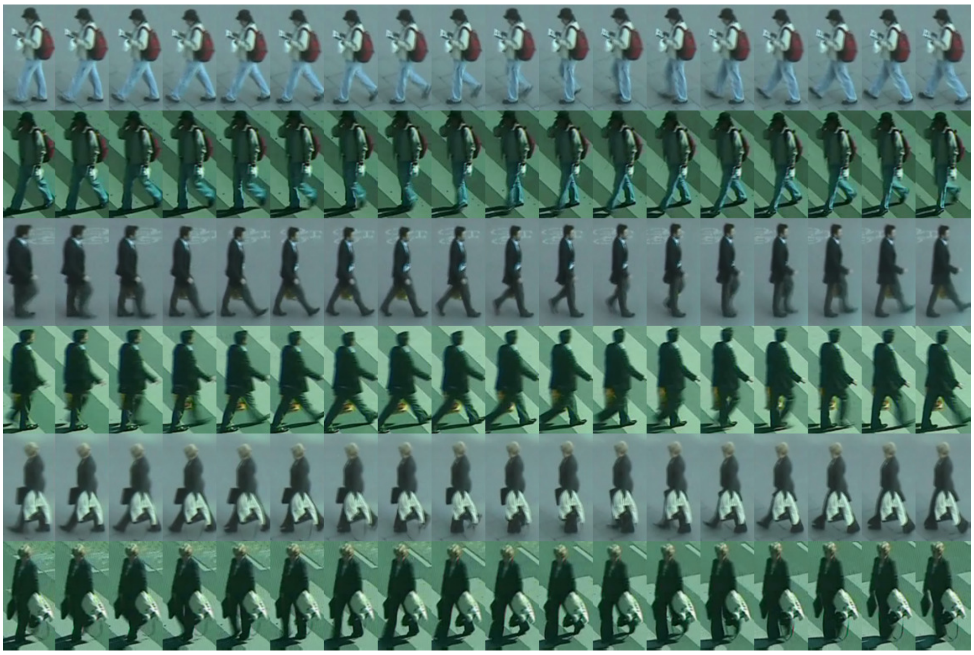

FIGURE 1.4: Samples from the prid_2011 dataset

In this thesis, we use the short-term, single-shot and close-set Re-ID. We assume the Re-ID is a short-term case so that the individuals wear the same clothes. This is vital because if individuals change their clothes, it won't be valid to identify them by their appearance features. Actually in most real-time surveillance systems, individuals do wear the same clothes over the surveillance. One example is to recognize and tracking customers in a store. The customers in a store wear the same clothes through out the time.

## 1.2 Challenges

**Detection, tracking and dataset labelling for supervised learning** Though classical person re-identification focuses on robust descriptors design and matching algorithms, in real-time application, the detection and tracking has to be operated on video frames to get bounding boxes of individuals. A good detection and tracking algorithm is necessary for Re-ID. Furthermore, training the matching algorithm is a supervised process, thus we have to know the labels for those training data.

**Descriptors design** Good descriptors should be robust to the variation of people's postures, outer environment changes and camera settings. Though there have been many kinds of descriptors based on different property like color and texture, it is hard to judge which property is universally useful for different camera settings. In fact, the robustness, reliability and feasibility depends on different camera settings and viewing conditions. What's more, the pedestrian background may add many errors to descriptors, so it is important to quantify the impact of a noisy background. Many works have tried to use a segmented foreground of pedestrians, so it is important to design segmentation algorithms. The automatic foreground segmentation for a single frame is difficult because there isn't much information available compared with video background segmentation. Take the VIPeR dataset as an example. There is only one frame for each view of a certain person, thus the segmented foreground masks are imperfect, and chances are high that important body parts will be lost. A segmented foreground provided by [15] is shown in Figure 1.5.

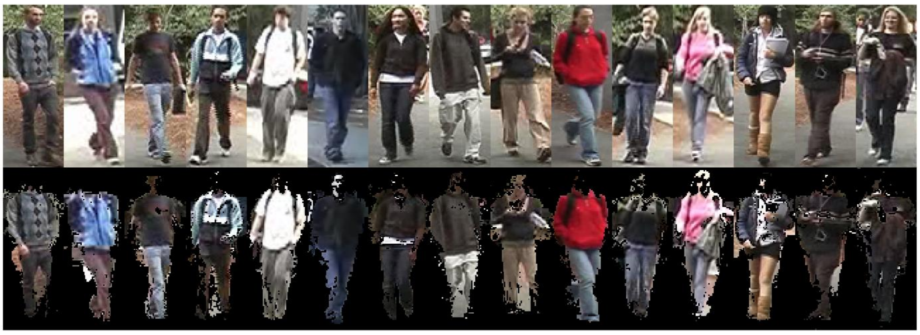

FIGURE 1.5: VIPeR foreground

**Efficient matching algorithm design** When designing machine learning algorithms to match persons, there are many limitations. One of them is the small sample size problem [48]. The extracted descriptors usually has a high dimension $d$ but only a small number of samples $n(n << d)$, and underfitting may appear because of insufficient data samples with high dimension. It is also necessary to take into consideration intraclass and interclass distance of samples. The intraclass distance means the distance of two samples with the same class label, while interclass distance is the distance of samples with different class labels.

**Feasibility, complexity and scalability** When applying those descriptors and matching algorithms, we have to consider real-time performance. The Re-ID datasets usually have small sample sizes, but in a surveillance network, more pedestrians in different cameras can be presented simultaneously. A system like this has plenty of individuals to re-identify, which requires that the processing time for a single probe be short for low latency. Because the gallery in this system evolves, it is crucial to design an algorithm that can determine if a person appearing in a current camera is new or has previously appeared in the gallery.

## 1.3 My work in this thesis

In this thesis, the programming involves C++, Matlab and Python. First the background segmentation's influence on Re-ID matching accuracy has been studied. A basic histogram descriptor is implemented with C++. By changing the color space parameters and plotting its matching accuracy curve on VIPeR dataset, the HSV color space is proved to be the best choice for histogram feature. Then by comparing the matching accuracy of HSV and other textural feature including local binary pattern (LBP) and histogram of gradient (HOG), it can be concluded that foreground segmentation (the segmentation is imperfect due to only one frame is provided for each individual) increases HSV's performance while decreases the performance of LBP and HOG. The CMC figures in chapter 3 are plotted with Python or Matlab

Second three variants of the hierarchical Gaussian descriptors have been designed and implemented with Matlab based on the source code from [25]. After comparing their performance with the original version of hierarchical Gaussian descriptor the original version has the best performance.

Third, the metric learning is designed and implemented with Matlab. In many previous works, the kernel local Fisher discriminant analysis is used as a subspace learning method, and Euclidean distance is usually used in the subspace to measure similarity. In this thesis, the KLFDA [38] method is used as a dimension-reducing method to project high-dimensional descriptors to a lower-dimensional space. Compared with other dimension reduction methods, KLFDA is a supervised method that takes into consideration intraclass and interclass information; therefore, less information is lost after dimension reduction. A Mahalanobis distance based matrix $M$ is then learned based on the limitation that the distance of people from the same class should be at least one unit smaller than the distance of people from different classes. A target function that penalizes large intraclass distance and small interclass distance is created. When the target function converges by iterative computation, the matrix $M$ is thought to be optimal. It turns out that this metric learning exhibits good performance when compared with other metric learning methods. A workflow of proposed work is in Figure 1.6.

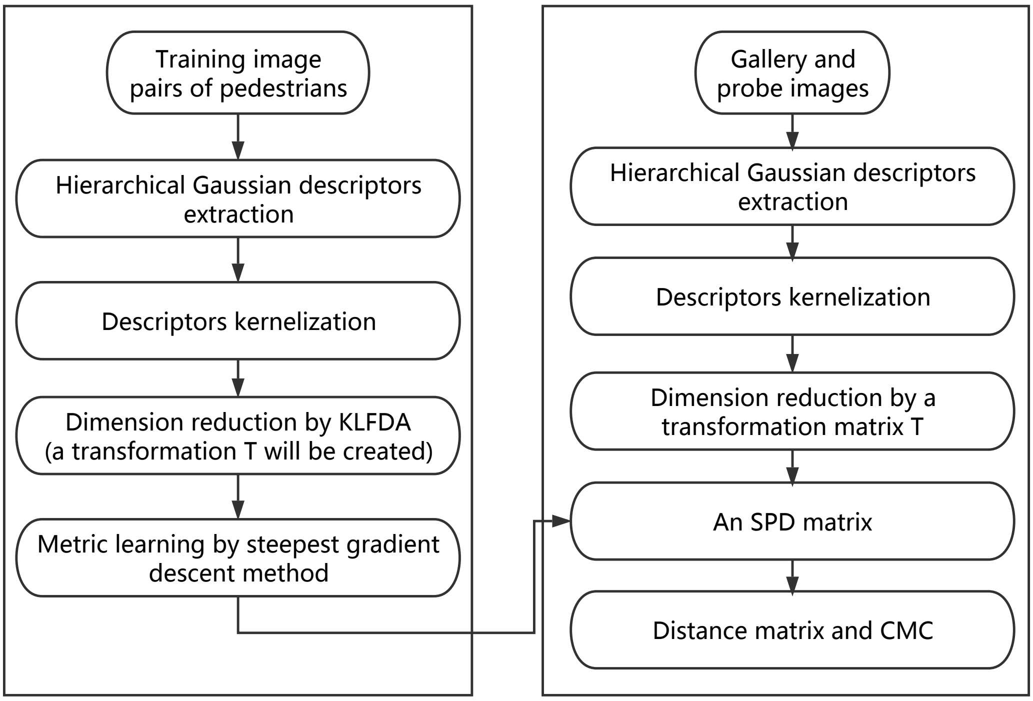
Training image pairs of pedestrians
Hierarchical Gaussian descriptors extraction
Descriptors kernelization
Dimension reduction by KLFDA (a transformation T will be created)
Metric learning by steepest gradient descent method
Gallery and probe images
Hierarchical Gaussian descriptors extraction
Descriptors kernelization
Dimension reduction by a transformation matrix T
An SPD matrix
Distance matrix and CMC

FIGURE 1.6: The workflow of proposed work; the left part is training and the right part is testing

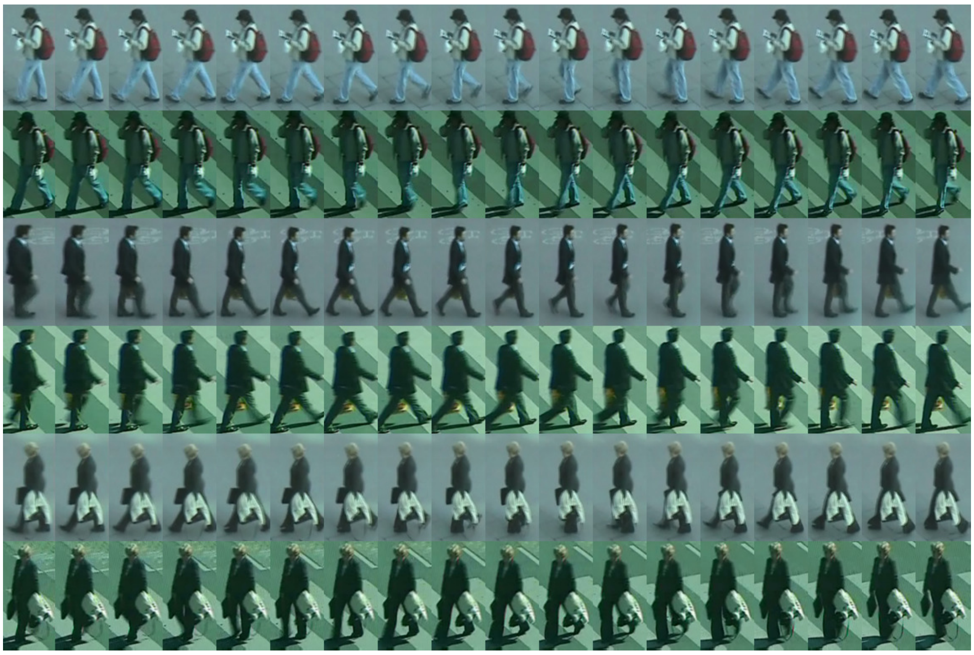

FIGURE 1.7: Samples from the prid_2011 dataset

## 1.4 Contributions

In this paper, we have three contributions. The first is that we combined the KLFDA with distance comparison metric learning. Instead of learning a subspace with KLFDA and computing Euclidean distance in the lower-dimensional space, in this thesis after dimension reduction by KLFDA a Mahalanobis metric is learned under the constraint that the maximum intraclass distance is at least one unit smaller than the minimum interclass distance. Compared with those advanced metrics including cross view quadratic analysis (XQDA) [22] and the null Foley-Sammon transfer (NFST), this proposed metric learning proves to have excellent performance on the VIPeR, CUHK1, prid_2011, prid_450s and GRID dataset.

Another contribution is we find that imperfect background subtraction increases histogram feature's performance and decreases textural feature's performance. Since in single-shot Re-ID, it's extremely hard to extract the foreground perfectly (only one frame of each individual is available to extract foreground), the extracted result may lose some parts of the foreground while contains some parts of the background. This imperfect foreground extraction can improve the performance of some descriptors, like the histogram of HSV color space, but can also decrease the performance of other descriptors, like local binary pattern (LBP) and histogram of gradient (HOG). This comparison is shown in Chapter 3. The reason for this is that imperfect background segmentation brings in textural interference. If descriptors are color based and don't handle texture information, like HSV histogram descriptor, background segmentation can greatly improve the performance. However, if the descriptor extracts texture information, background segmentation will decrease its performance since the imperfect segmentation will mask out many parts of the foreground area, which will cause important textural information variation. Because segmentation algorithms will cause different influence on various features, in this thesis, a weighted map of images is used instead of the background segmentation.

For the last contribution, some variants of hierarchical Gaussian descriptor have been tested. In one variant, LBP was used in the basic pixel feature. In another variant, superpixel segmentation was also applied to combine with hierarchical Gaussian descriptor, which implied that overlapping patch sampling is important in the hierarchical Gaussian descriptor. At last, the Gaussian mixture model (GMM) was also tested but it displayed the worst performance.

## 1.5 Thesis organization

In this thesis, Chapter 2 will give a brief introduction of previous work. Chapter 3 will explain the implementation of the hierarchical Gaussian descriptors used in this thesis. This is also the chapter where my work starts. The performance of some variants of the hierarchical Gaussian descriptor is studied in this chapter. In Chapter 4, a detailed introduction of the kernel local Fisher discriminant analysis will be presented, and a detailed explanation of the metric learning on the lower-dimensional space based on relative distance limitation learning will also be provided. In Chapter 5, the used datasets and parameters and other experiment settings will be explained, and a detailed analysis of results will be presented there. Finally, the conclusion is given in Chapter 6.

# Chapter 2

# Related work

Previous works mainly focus on finding more discriminative descriptors and better metric learning. It is known that color and texture are the most important information in Re-ID. Researchers found that features based on a single attribute (like color or texture) are not robust to various datasets. Instead, combinations of different features are exploited to improve the performance. Most descriptors capture the local or global statistic color and texture information to characterize individuals. A brief introduction of those descriptors and metrics is given in this chapter.

## 2.1 Appearance descriptors

In most descriptors, local features [36] are used to characterize the color, texture and other properties. The input image will be divided into a few subregions, features of those subregions are extracted respectively and concatenated directly or characterized by their statistic properties. According to how those subregions are divided, there can be three kinds of models, fixed part-based models, adaptive models and learned part models [36].

In fixed part models, the size of body parts is predefined. One example is in [11, 51, 34], where a silhouette is divided into a fixed number of horizontal stripes, which mainly include the head, torso and legs. In [19], the input images are divided into three horizontal stripes, and the width of each stripe is respectively 16%, 29% and 55%. The fixed models predefine parameters such as numbers of stripes and the stripe height.

In the adaptive part models, the size of each body parts may vary to fit predefined body part models. Take [15] for an instance; the silhouette of each person is divided into three parts horizontally, which include the head, torso and legs respectively.

But the height of each stripe is different for various silhouettes, and it is computed according to symmetry and asymmetry with two operators $C(i, \delta)$ and $S(i, \delta)$, where

$$\begin{aligned} C(i, \sigma) &= \sum_{B_{[i-\delta, i+\delta]}} d^2(p_i - \hat{p}_i), \\ S(i, \sigma) &= \sum \frac{1}{W\delta} |A(B_{[i,i-\delta]}) - A(B_{[i,i+\delta]})|. \end{aligned} \tag{2.1}$$

Here the $C(i, \delta)$ is called the *chromatic bilateral operator*, and it computes the Euclidean distance of two HSV blobs located symmetrically with respect to horizontal axis $y = i$, where $B_{[i-\delta,i+\delta]}$ is the blob with a height $2\delta$. $S(i, \delta)$ is called the *spatial covering operator* and it computes the difference of two foreground areas. Then the axis between the torso and legs is computed as follow:

$$i_{TL} = \arg\min_i (1 - C(i, \delta) + S(i, \delta)), \tag{2.2}$$

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

$$i_{HT} = \arg\min_i (-S(i, \delta)). \tag{2.3}$$

The axis dividing the left and right torso is

$$j_{LR} = \arg\min_j (C(j, \delta) + S(j, \delta)). \tag{2.4}$$

This method has good performance. But one shortcoming of this model is that an imperfect background segmentation causes noise and introduces errors regarding the position of axes.

Another part-based adaptive spatial-temporal model used in [7] characterizes a person's appearance using color and facial features. Few works exploit human face features. In this work, human face detection 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 a color model. This model handles multi-shot re-identification, and it also characterizes 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 (statistic classifiers) are trained with manually labelled human body parts images, exploiting features related to edges contained in the images. A pictorial structure (PS) is proposed in [16]. The 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 [2], the pictorial structure body model is made up of N parts and N corresponding part detectors.

Another example of the learned part model is in [8, 7]. 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 defines allowed arrangements of this part with respect to the bounding box. To train each model, the latent support vector machine (LSVM) is used, and four body parts are detected, namely the 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.

Features can be implemented with different methods. According to the way to extract features for a model (a whole model or a part-based model), features can be divided into two categories [36]: global and local features. Global features refer to features extracted from a whole image or region, and the size of the descriptor is usually fixed. In order to extract 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 information is exploited most by extracting the color histogram within different color spaces. Descriptors based on texture, such as the scale-invariant feature transform (SIFT), speeded up robust features (SURF) and LBP, are also widely combined to improve performance.

Global color histogram is a frequently used global feature. For an three-channel image, like an RGB image, each channel is quantized into $B$ bins separately. The final histogram could be a multi-dimensional or one-dimensional histogram. For instance, if $B = 8$, for a multi-dimensional histogram, there will be $8 \times 8 \times 8 = 512$ bins. But if we concatenate the three-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-sized blocks and computes the global feature of each block. The feature can be based on color, texture and interest points. SIFT [24] 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 key points by computing the difference of Gaussian ($DoG$) images of different scales $\sigma$ with the equation

$$D(x, y, \sigma) = (G(x, y, k_1\sigma) - G(x, y, k_2\sigma)) * I(x, y). \tag{2.5}$$

Here $G(x, y, k_1\sigma)$ is the Gaussian function with deviation $k_1\sigma$, $I(x, y)$ is the image. The $DoG$ images are compared to find their extrema as key points. With key points localization and other processing, descriptors describing key points are created as SIFT descriptors.

The maximally stable color region (MSCR) is used in [15]. The MSCR derives from the maximally stable extreme region (MSER) and detects the region with a stable color cluster. It uses an agglomerative clustering algorithm to compute color clusters, and 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 makes it easy to do scale and affine transforms.

Recurrent highly structured patches (RHSP) is also used in [15]. This feature captures patches with highly recurrent color and texture characteristics from extracted silhouette pixels. This feature is extracted from the 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 the three channels' entropy) is computed, and 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 remaining invariant to the transforms. Subsequently, the recurrence of each patch is evaluated with the local normalized cross correlation (LNCC) function. This evaluation is only performed on a small region containing the patch instead of the whole image. Then the patches with high recurrence are clustered to avoid patches with similar content. Finally, the Gaussian cluster is applied to maintain the patch nearest to the centroid of each cluster for each cluster.

In [3], a new 3D model model called SARC3D is used to characterize the individual. Compared with those 2D 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 a single shape parameter. By precise 3D mapping, this parameter can be learned and trained with even quite 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 a person is extracted to construct the 3D graphical model. The final body model is sampled to get a set of vertices from the previously learned graphic body model. Compared with other models, this model has a robust performance when dealing with partial occlusion, people posture variation and viewpoint variations since the model is based on silhouettes from three viewpoints.

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, which 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. Similar to the SDALF descriptor [15], the 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 $[H = 16, S = 16, V = 4]$. The authors define epitome to be a collage of overlapped patches. This collage is generated by the collapsing of an image or a sequence of images, and this collage indicates the essence of the texture, shape and chromatic properties of the image or sequential images.

## 2.2 Metric learning

The second step of Re-ID is to design the metric learning to match descriptors, i.e., the way to compare how similar two descriptors are. Many different metric learning methods have been proposed [20, 33, 27, 47, 51, 41, 38, 42, 44, 48, 13] to get smaller intraclass distance and larger interclass distance. Generally, for two $d \times 1$-dimensional input vectors $\boldsymbol{x}_1$, $\boldsymbol{x}_2$, any symmetric positive semi-definite matrix $M$ defines a pseudo-metric with the form of $D = (\boldsymbol{x}_1 - \boldsymbol{x}_2)^T \boldsymbol{M} (\boldsymbol{x}_1 - \boldsymbol{x}_2)$. Many widely used distance metrics exploit this rule. Previous methods includes the Euclidean distance, Bhattacharyya distance and Mahalanobis distance methods. The Euclidean distance, which is the most common used distance, is a special case of Mahalanobis distance when the $\boldsymbol{M}$ is an identity matrix. One example of metric learning is the probabilistic relative distance comparison model proposed in [51]. This model exploits all the possible positive person pairs and negative pairs so that for each person, the between-class distance is larger than the within-class distance. Compared with the other distance learning models proposed, this model solves the matrix $\boldsymbol{M}$ by an iterative optimization algorithm. Suppose $\boldsymbol{z}$ is an image descriptor of a person; the task is to identify another image descriptor $\boldsymbol{z}'$ of the same person from $\boldsymbol{z}''$ of a different person by using a distance model $f(\cdot)$, so that $f(\boldsymbol{z}, \boldsymbol{z}') < f(\boldsymbol{z}, \boldsymbol{z}'')$. The authors convert the distance learning problem to a probability comparison problem by measuring the probability of the distance between a relevant pair of images being smaller than that of a related irrelevant pair as

$$P(f(\boldsymbol{z}, \boldsymbol{z}') < f(\boldsymbol{z}, \boldsymbol{z}'')) = (1 + \exp^{(f(\boldsymbol{z}-\boldsymbol{z}') - f(\boldsymbol{z}-\boldsymbol{z}''))})^{-1}. \tag{2.6}$$

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

$$\begin{aligned} f &= \arg\min_f r(f, O), \\ r(f, O) &= -\log(\prod_{O_i} P(f(\boldsymbol{x}_i^p) < f(\boldsymbol{x}_i^n))). \end{aligned} \tag{2.7}$$

$O = \{O_i = (\boldsymbol{x}_i^p, \boldsymbol{x}_i^n)\}$ , $\boldsymbol{x}_i^p$, $\boldsymbol{x}_i^n$ are intraclass and interclass vector differences respectively, and $\boldsymbol{x}_i^p$, $\boldsymbol{x}_i^n$ are defined as

$$\begin{aligned} \boldsymbol{x}_i^p &= |\boldsymbol{z} - \boldsymbol{z}'|, \\ \boldsymbol{x}_i^n &= |\boldsymbol{z} - \boldsymbol{z}''|. \end{aligned} \tag{2.8}$$

The distance function $f(\cdot)$ here is parameterized as the Mahalanobis distance function $f = \boldsymbol{x}^T \boldsymbol{M} \boldsymbol{x}, \boldsymbol{M} \geq 0$. Here $\boldsymbol{M}$ is a semi-positive definite (SPD) matrix. In this way, the distance function learning problem is transformed to a matrix optimization problem. The author used an iteration algorithm to compute matrix $\boldsymbol{M}$. One shortcoming for this algorithm is that it is computationally expensive because for each person it compares distances between all the possible negative pairs and corresponding positive pairs.

Single-shot image-based person representation suffers from the small sample size problem. This is why multi-shot Re-ID has been proposed. Since there are a sequence of images for each individual, there are many more cues to exploit. In [47], the author simplified computing of the Mahananobis matrix by applying the new limitations on datasets. The author found that when using video-based person representation, the difference of interclass may be more obscure than that of still-image-based representation. Therefore, the author proposed the top-push distance learning model. For all the person pairs, the maximal intraclass distance should be one unit smaller than the minimal distance of interclass distance. Another constraint introduced the sum of all intraclass distance should be as small as possible, so the final target function is summarized as:

$$\begin{split} f(D) = (1-\alpha) \sum_{\boldsymbol{x}_i, \boldsymbol{x}_j, y_i = y_j} D(\boldsymbol{x}_i, \boldsymbol{x}_j) + \\ \alpha \sum_{\boldsymbol{x}_i, \boldsymbol{x}_j, y_i = y_j} \max\{D(\boldsymbol{x}_i, \boldsymbol{x}_j) - \min_{y_i \neq y_k} D(\boldsymbol{x}_i, \boldsymbol{x}_k) + \rho, 0\}. \end{split} \tag{2.9}$$

Here, $\boldsymbol{x}_i, \boldsymbol{x}_j and \boldsymbol{x}_k$ are feature descriptors, and $y_i, y_j and y_k$ are class labels.

Some previous works exploit the Fisher discriminant analysis and convert the problem to a matrix eigenvalue decomposition problem. In [33], the local fisher discriminant analysis (LFDA) is used, and in [44], the kernel version of several linear metrics are proposed, and it proves that kernelization improves Re-ID performance. In [22], the log ratio of Bayesian-based estimation is proposed to have advanced performance. In [48], the authors propose to make sample points of the same class collapse to the same point in the null space while different points are project to different points. In [46], the semantic representation of color names is exploited. In this thesis, an SPD matrix $\boldsymbol{M}$ is learned to meet certain intraclass and interclass distance limitations.

## 2.3 Other methods for Re-ID

Besides the descriptors and metrics mentioned above, there are some other methods for Re-ID. Convolutional neural networks (CNN) have been exploited in Re-ID. One advantage of the neural network Re-ID is that the preprocessing of images can be skipped. We can also say the preprocessing is included in convolutional layers. The input of this structure can be straightforward grey images or color images. Traditional neural networks have too many weights to train. Convolutional neural networks can avoid this problem while retaining high performance. Compared with classical neural network architecture, the convolutional neural network exploits receptive field, weights sharing and pooling to reduce weights number and thus decreases computational cost. When dealing with multi-shot and video-based re-identification, neural networks are proven to have better performance. In [26], 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 [10], the author proposes a multi-channel layer based neural network to jointly learn both local body parts and whole body information from input person images. In [43], 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.

There are many other works based on convolutional neural networks. However, person re-identification may be one of the areas where CNN performance may be poorer than regular machine learning methods for the small sample size problem (SSS). In most datasets, the sample size of each pedestrian is quite small. Especially in single-shot Re-ID only one frame is provided in each view for each person. This is why Re-ID more often relies on classical machine learning.

## 2.4 Some state-of-the-art works

Recently, many works have been proposed that improved Re-ID performance by a wide margin. In this section, those advanced descriptors and metrics are introduced.

**Cross view quadratic discriminant analysis** (XQDA) is proposed in [22]. Let's define the sample difference $\Delta = \boldsymbol{x}_i - \boldsymbol{x}_j$, where $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are two feature vectors. $\Delta$ is called intrapersonal difference when their labels satisfy $y_i = y_j$ and extrapersonal difference when $y_i \neq y_j$. The intrapersonal and interpersonal variation can be defined as $\Omega_I$ and $\Omega_E$. The authors convert the Re-ID problem to distinguishing $\Omega_I$ from $\Omega_E$. In [28], each one of intrapersonal and interpersonal class is modelled with a multivariate Gaussian distribution, and in [28], it has been proven that both $\Omega_I$ and $\Omega_E$ have zero mean. Under the zero-mean distribution, the probability of observing $\Delta$ in $\Omega_I$ and the probability of observing $\Delta$ in $\Omega_E$ can be denoted as

$$\begin{aligned} P(\Delta|\Omega_I) &= \frac{1}{(2\pi)^{d/2}|\Sigma_I|^{1/2}} \exp^{\frac{-1}{2}\Delta^T\Sigma_I^{-1}\Delta}, \\ P(\Delta|\Omega_E) &= \frac{1}{(2\pi)^{d/2}|\Sigma_E|^{1/2}} \exp^{\frac{-1}{2}\Delta^T\Sigma_E^{-1}\Delta}, \end{aligned} \tag{2.10}$$

where $\Sigma_I$ and $\Sigma_E$ are the covariance matrix of $\Omega_I$ and $\Omega_E$. Then the probability ratio between the interpersonal pairs and intrapersonal pairs can be denoted as

$$\begin{aligned} r(\Delta) = \frac{P(\Delta|\Omega_E)}{P(\Delta|\Omega_I)} &= \frac{\frac{1}{(2\pi)^{d/2}|\Sigma_E|^{1/2}} \exp^{\frac{-1}{2}\Delta^T\Sigma_E^{-1}\Delta}}{\frac{1}{(2\pi)^{d/2}|\Sigma_I|^{1/2}} \exp^{\frac{-1}{2}\Delta^T\Sigma_I^{-1}\Delta}}, \\ r(\Delta) &= C \exp^{\frac{-1}{2}\Delta^T(\Sigma_E^{-1}-\Sigma_I^{-1})\Delta}. \end{aligned} \tag{2.11}$$

Here C is a constant term, and by taking the log and deserting the constant term, we have

$$r(\Delta) = \Delta^T(\Sigma_I^{-1} - \Sigma_E^{-1})\Delta = (\boldsymbol{x}_i - \boldsymbol{x}_j)^T(\Sigma_I^{-1} - \Sigma_E^{-1})(\boldsymbol{x}_i - \boldsymbol{x}_j). \tag{2.12}$$

In [22], a subspace $W$ is learned so that

$$r(\Delta) = (\boldsymbol{x}_i - \boldsymbol{x}_j)^T W({\Sigma_I}'^{-1} - {\Sigma_E}'^{-1})W^T(\boldsymbol{x}_i - \boldsymbol{x}_j), \tag{2.13}$$

and ${\Sigma_I}' = W^T\Sigma_I W, {\Sigma_E}' = W^T\Sigma_E W$. Therefore, a subspace $M(W) = W({\Sigma_I}'^{-1} - {\Sigma_E}'^{-1})W^T$ is learned in this work.

**Null Foley-Sammon transform (NFST)** In [48], a null space is proposed so that with this space the intraclass points collapse to a same point in the null space while interclass points are projected to different points. Given the within-class scatter $\boldsymbol{S}^w$ and between-class scatter $\boldsymbol{S}^b$, an optimal projection matrix $\boldsymbol{W}$ is computed so that

$$\begin{aligned} \boldsymbol{w}_i^T \boldsymbol{S}^w \boldsymbol{w}_i &= 0, \\ \boldsymbol{w}_i^T \boldsymbol{S}^b \boldsymbol{w}_i &> 0. \end{aligned} \tag{2.14}$$

Here $\boldsymbol{w}_i$ is the $i_{th}$ column in $\boldsymbol{W}$.

It can be noted that like many other metric learnings, XQDA and NFST evolve to matrix decomposition and eigenvalue selection problems. In this paper, XQDA and NFST are used to compare with the proposed metric. In [25], it has been shown that GOG + XQDA outperforms many other combinations, including Metric ensemble [32], SCNCD [46], Semantic method [37], etc. In [48], it has been shown LOMO + NFST outperforms metrics including LMNN [41], KCCA [42], ITML [13], KLFDA [38], MFA [44], KISSME [20], Similarity learning [9], SCNCD [46], Mid-level filters [50] and Improved deep learning [14]. Based on the result that XQDA and NFST outperform other metrics, XQDA and NFST are used in this thesis to compare with our proposed metric learning.

## 2.5 Performance measuring

There are a few measures of Re-ID, such as cumulative matching curve (CMC), receiver operating characteristic curve (ROC) and synthetic recognition rate (SRR). Specifically, the CMC is used as a 1:m re-identification system and ROC is used for a 1:1 re-identification system. The SRR curve indicates the probability that any of given fixed number of matches is correct.

For the open-set Re-ID problem, the ROC [7, 8] curve is adopted. The ROC represents the relationship between Re-ID accuracy vs the false accept rate (FAR). In the ROC the x-axis is the FAR and the y-axis is the accuracy. Re-ID accuracy and the FAR are defined by equations

$$\begin{aligned} Accuracy &= \frac{TPs}{N_p}, \\ FAR &= \frac{MMs + FPs}{N_p}. \end{aligned} \tag{2.15}$$

The true positives (TPs) are the number of probe IDs that are correctly accepted. $N_p$ is the number of all positive probes. The FAR is expressed by the mismatches (MMs), false positives (FPs) and $\mathrm{N_p}$. MMs are those probe IDs that are incorrectly matched to the galley set when in fact those probe IDs exist in the gallery. The FPs are the number of probe IDs incorrectly matched to the gallery when they don't actually exist in the gallery.

For the closed-set Re-ID problem, the most commonly used performance measuring is the CMC. Closed-set Re-ID assumes that the probes are contained in the gallery set, and the performance is measured by ranking the gallery with respect to each probe. The CMC describes the probability of the right match given a list of computed similarity scores, and the first ranked ID is regarded as the matched individual.

The proportion of uncertainty removed (PUR) is proposed in [33]. PUR measures the entropy difference between before ranking and after ranking. For a certain probe before ranking, all those samples in the gallery set have equal probability, and the entropy is $\log(S)$, $S$ being the number of observations in gallery set. After ranking, the entropy is $\sum M(r)\log(M(r))$, and $M(r)$ is the probability that the right match is within the top $r$ ranked observations. With normalization, the PUR value is computed by the equation

$$PUR = \frac{\log(S) - \sum M(r)\log(M(r))}{\log(S)}. \tag{2.16}$$

In this thesis, the CMC is used to measure Re-ID performance because most datasets are tested under closed-set Re-ID settings. In this metric, the $CMC(k)$ stands for the probability that the right match is within the top k matches. Suppose a set of gallery $G = \{G_1, G_2, \cdots, G_m\}$ and a set of probe $P = \{P_1, P_2, \cdots, P_n\}$, for each identity $P_i$, there should be a right match in the gallery set. However, there could be identities that appear in the gallery set but not in the probe set. A $m \times n$ similarity matrix can be computed. Then for each probe identity $P_i$, a sorted list of gallery identities can be list as $S(P_i) = \{G_{(1)}, G_{(2)}, \cdots, G_{(m)}\}$ so that their similarity with $P_i$ descends. Suppose the right match of $P_i$ is at the position $k$ of $S(P_i)$, $k \leq m$, then $G_i$ has a rank value of $k$. Therefore, the CMC can be calculated as

$$CMC(k) = \frac{1}{n}(\#k_l \leq k), \tag{2.17}$$

where $k_l$ is the list of rank values of $P = \{P_1, P_2, \cdots, P_n\}$, and $\#k_l \leq k$ means the number of rank values that are smaller than k. Therefore, the CMC curve always ascends and converges at 1. A perfect CMC is supposed to have a hight rank 1 score and approaches 1 as fast as possible.

An example of CMC computing is as follows. Suppose the gallery size is $M = 10$ and the probe size is $N = 15$. By computing the similarity and ranking them, we have a rank score frequency table such as that shown in Table 2.1.

TABLE 2.1: A CMC example of gallery size is $M = 10$ and probe size is $N = 12$

| RankScore | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Frequency | 5 | 3 | 2 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| CumulativeSum | 5 | 8 | 10 | 11 | 12 | 12 | 12 | 12 | 12 | 12 | 12 | 12 |

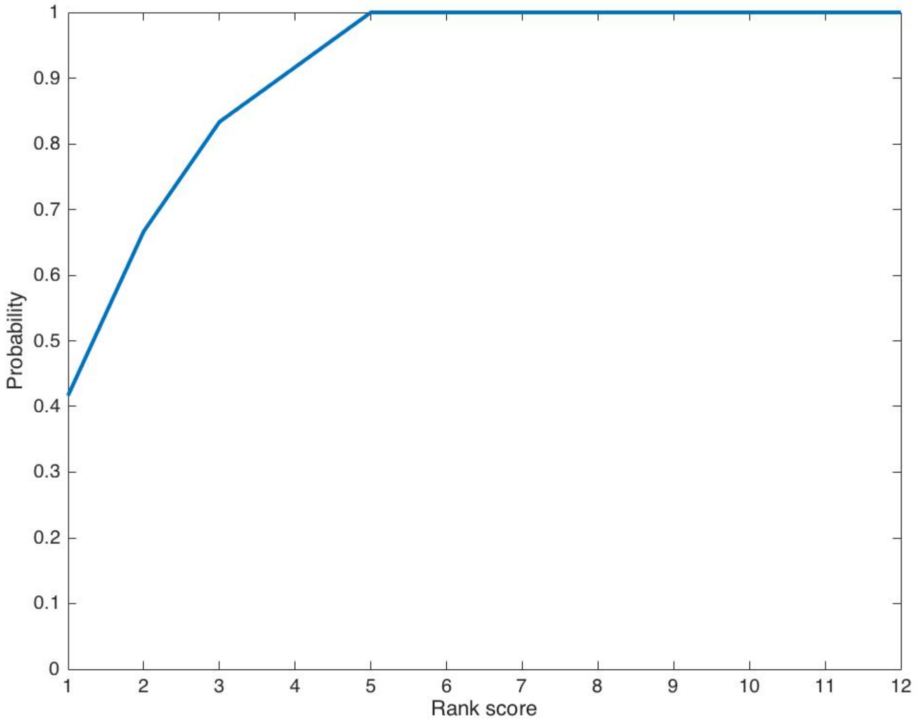

FIGURE 2.1: A CMC plot of gallery size is $M = 10$ and probe size is $N = 12$

# Chapter 3

# Extraction of hierarchical Gaussian descriptors

In person re-identification, it is very important to choose a robust descriptor to represent a person. A good descriptor should be robust to variations of illumination, viewpoint, and camera color response. Most descriptors try to capture the color and texture information. In this chapter, we will first introduce some basic descriptors and compare their performance on the VIPeR dataset. Then a detailed introduction of the hierarchical descriptor will be presented in section 3.3 of this chapter. All those figures in this chapter are original work programmed and plotted by myself.

## 3.1 Basic color and textural features

### 3.1.1 Color histogram descriptors on different color space

Histogram descriptors extract color statistics information of input images. A popular histogram extracting method is to divide input image into a few horizontal stripes and extract color histograms of each stripe, which are then concatenated to produce a histogram descriptor of the whole image. Color space selection heavily influences descriptor performance. HSV color space is commonly used in computer vision and image processing for target detection and tracking. The HSV descriptor has better performance than the RGB histogram descriptor since HSV color separates image intensity from color information. Thus, HSV color space is more robust to illumination variation. An unsupervised CMC performance comparison among different color spaces on the VIPeR dataset is given in Figure 3.2. In this comparison, camera A views are used as the probe set and camera B views are used for the gallery set.

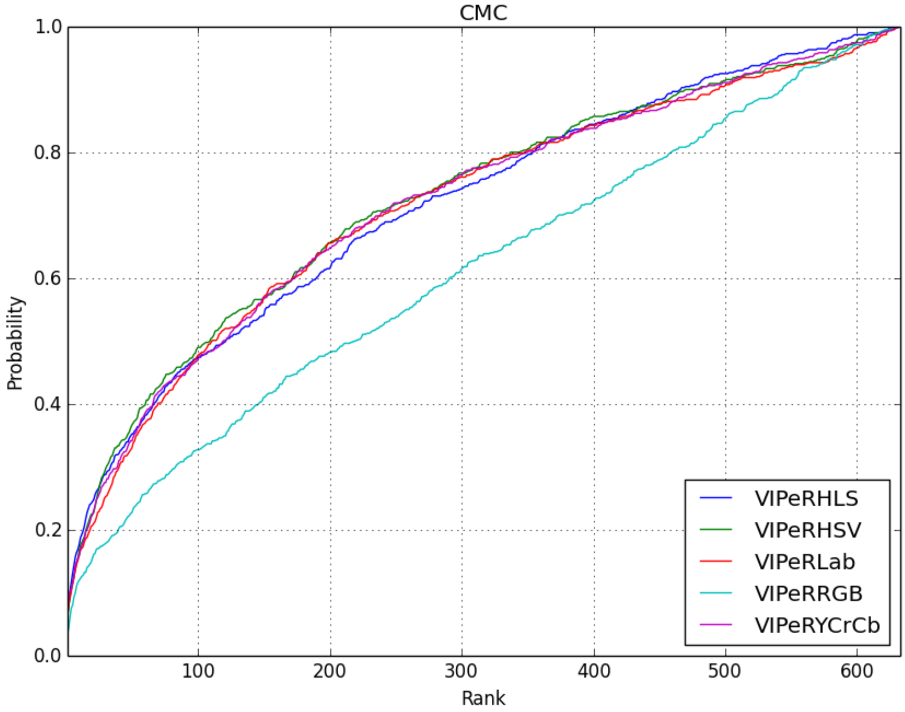

FIGURE 3.2: A CMC comparison of color histogram on different color spaces

We can find that those color spaces separating intensity information outperform RGB color space by a large margin.

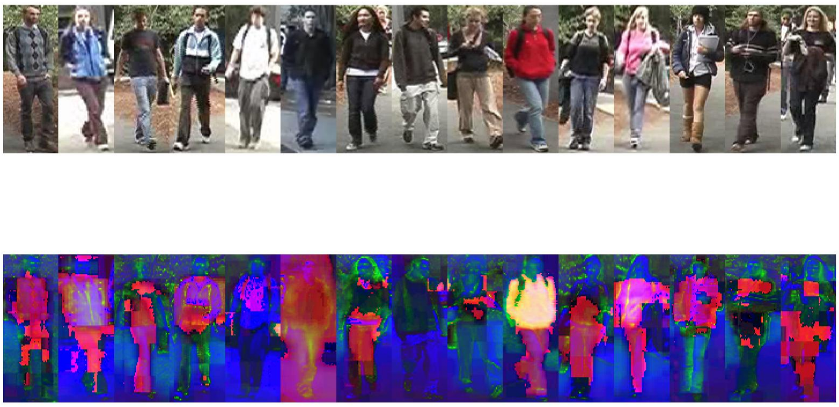

FIGURE 3.1: RGB and HSV visual comparison; the first row is RGB and second row is HSV for the same views

**Shortcoming of histogram-based descriptor** The performance of histogram descriptors suffers from ignoring the spatial information since it doesn't consider the relative distribution of color patches. Images with the same kind color patches but different distribution may have the same histogram descriptor. One example is shown in Figure 3.3.

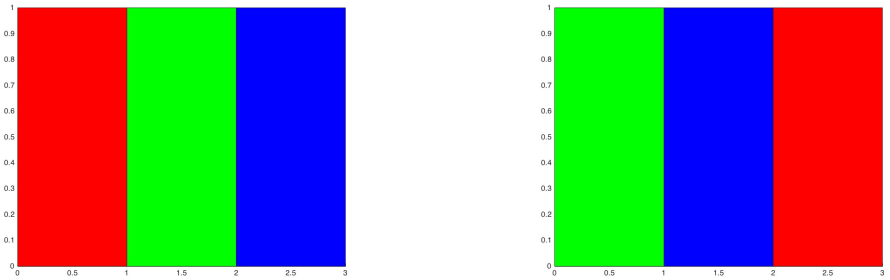

FIGURE 3.3: A comparison of two patches with the same entropy but different color distribution

### 3.1.2 Local binary pattern (LBP)

Local binary pattern [30, 29] extracts the texture information with efficient computing and has been used on people detection and recognitions. Figure 3.5 is an example of LBP. By thresholding the neighbour pixels of the center pixel (shown in Figure 3.4), the pixels are transformed into a binary integer. There are many extended LBPs like tLBP [39], VLBP [49] and OCLBP [4]. Besides, LBP is well known for its robustness to monotonic illumination variation.

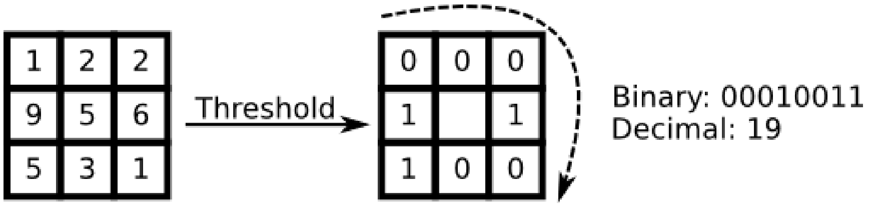

FIGURE 3.4: LBP: by thresholding the neighbour pixels, the pixels are transformed into a binary number

### 3.1.3 Histogram of oriented gradients (HOG)

The HOG [12] descriptor also extracts textural information of images by gradient computing. A brief introduction about its gradient computation is presented here, and more details can be found in [12]. The HOG feature computes the gradient of input intensity image $I(x, y)$ by equations

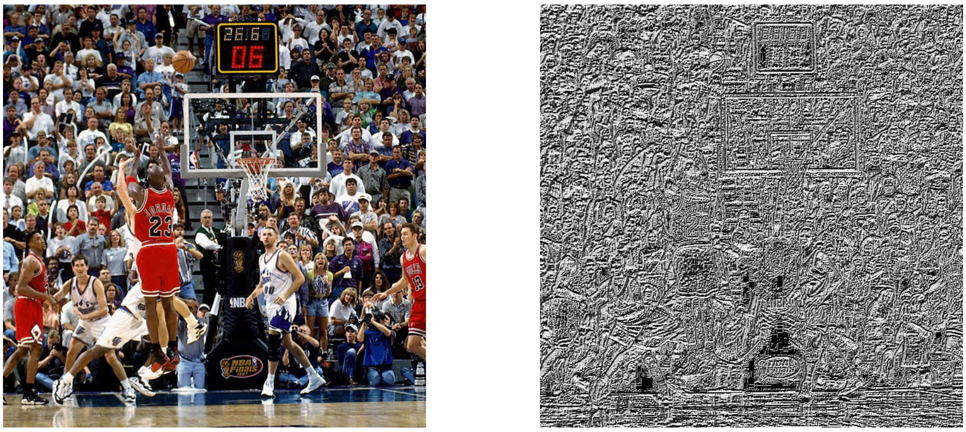

FIGURE 3.5: One LBP example

$$
\begin{aligned}
I_x &= \frac{\partial I}{\partial x}, \\
I_y &= \frac{\partial I}{\partial y}.
\end{aligned} \tag{3.1}
$$

The gradient can be computed quickly by some discrete derivative masks below, like 1-D Sobel masks:

$$
\begin{aligned}
Centered : M_c &= [-1, 0, 1], \\
Uncentered : M_{uc} &= [-1, 1].
\end{aligned} \tag{3.2}
$$

Or 2-D Sobel masks:

$$
\begin{aligned}
D_x &= \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}, \\
D_y &= \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}.
\end{aligned} \tag{3.3}
$$

Or $3 \times 3$ Sobel masks:

$$
\begin{aligned}
S_x &= \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \\
S_y &= \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}.
\end{aligned} \tag{3.4}
$$

.

Using different masks will lead to different performance. Furthermore, Gaussian smoothing is often performed before gradient computing. It has been shown that using 1-D Sobel without Gaussian smoothing has the best performance. A HOG feature demo is shown in Figure 3.6.

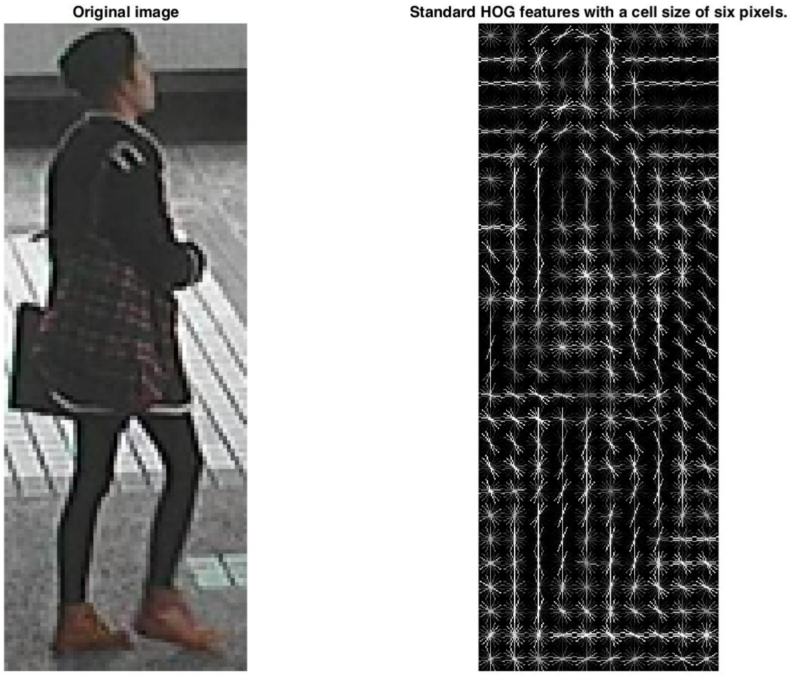

FIGURE 3.6: A demo of the HOG feature with a cell size of six pixels

## 3.2 Influence of background segmentation on different basic descriptors

Many works try to minimize the impact of background noise of pedestrians' images. It is easier to automatically segment the foreground from sequential frames or video than a single frame. In [15], the author provides foreground masks for all images following the algorithm in [18]. Authors in [18] propose structure element (stel) component analysis to model spatial correlations in image class structure by using probabilistic index maps (PIMs). A structure element is an area of an image with the same assigned index $s(s = 1, 2, \cdots, n)$, where $n$ is the total number of stels in an image. In PIMs, the indices assignment can be denoted probabilistically as a map $q(s^t_{\boldsymbol{i}} = s)$ over an image. In this equation, the location is $\boldsymbol{i} = (i, j)$, and $q$ is the probability of pixel at $\boldsymbol{i}$ of the $t$-th image (suppose there are many images for each class) belonging to the $s$-th stel. The authors propose that pixels in a structure element of an image class follow a shared distribution which can be modelled with local measurements like pixel intensity value. The number of stels $n$ is set to 2 to achieve background and foreground segmentation. That is, pixels of foreground stel share a distribution, while pixels of background stel share another distribution. Both the foreground and background stels are modelled by a mixture of $C$ Gaussians. Some of those segmented foregrounds are shown in Figure 3.7, and it is obvious that certain body parts, like head and feet, are lost. To compare the impact of loss on color and textural descriptors, a comparison of foreground segmentation on HSV color histogram descriptors, LBP and HOG is given in Figure 3.8, Figure 3.9 and Figure 3.10.

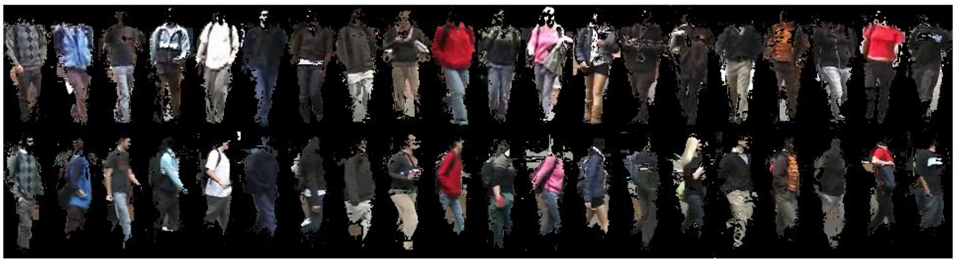

FIGURE 3.7: Foreground segmentation of individuals from VIPeR

We can find that imperfect foreground segmentation decreases the LBP and the HOG's performance but increases the HSV color histogram's performance on the VIPeR dataset greatly. The foreground segmentation is imperfect due to that segmentation algorithm is based on the Gaussian mixture model. The foreground model estimates the probability of being the foreground of each pixel. Since only one frame is provided, the model may not be quite accurate. Imperfect foreground segmentation causes body parts (like the head and the feet) to be lost and masks out many parts in the torso and legs. Besides, in images of some individuals, part of the background scene is regarded as foreground. Since HSV color histogram doesn't handle spatial distribution but only color entropy, foreground segmentation improves its performance greatly. But since LBP and HOG handle texture for each sample patch, their performances suffer from those body parts loss and little black patches from the background. What's more, we can infer that imperfect foreground segmentation will also decrease other textural features' performances.

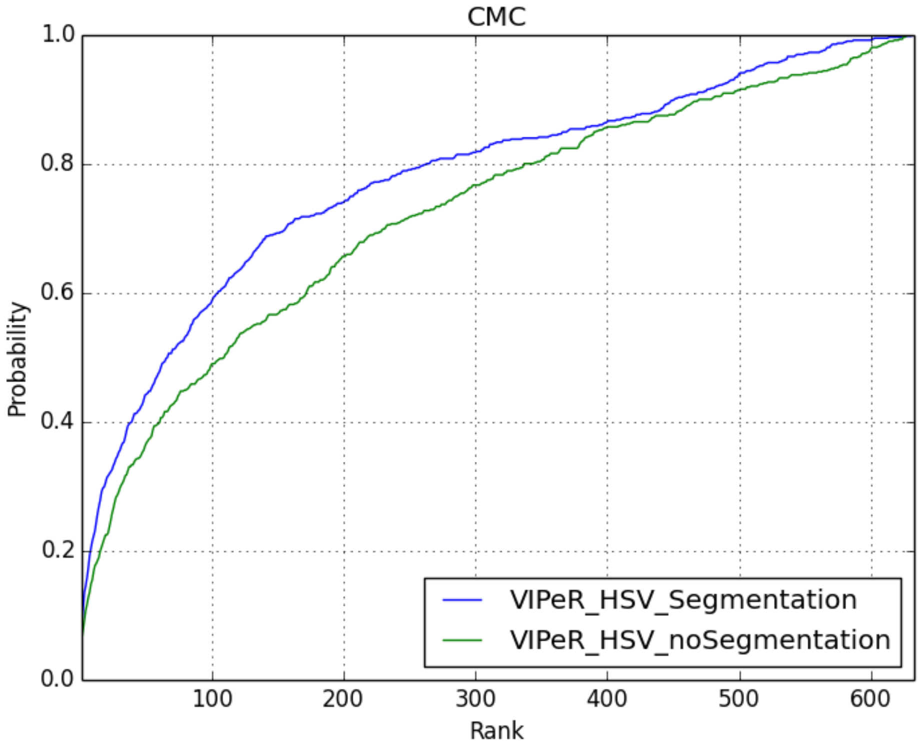

FIGURE 3.8: A CMC comparison of foreground segmentation on HSV histogram descriptor tested on VIPeR

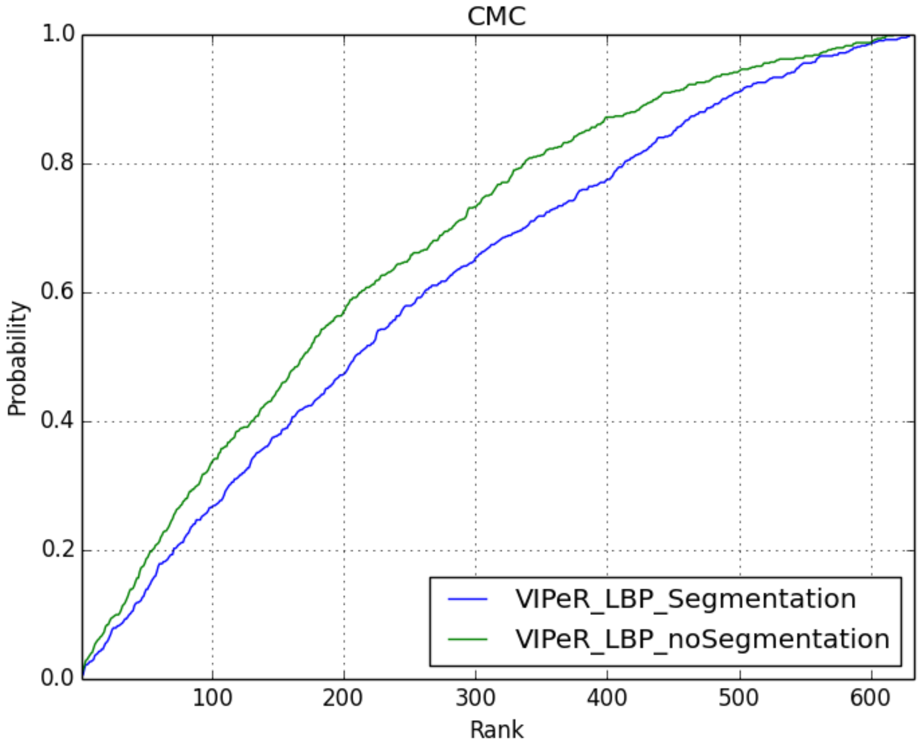

FIGURE 3.9: A CMC comparison of foreground segmentation on LBP feature tested on VIPeR

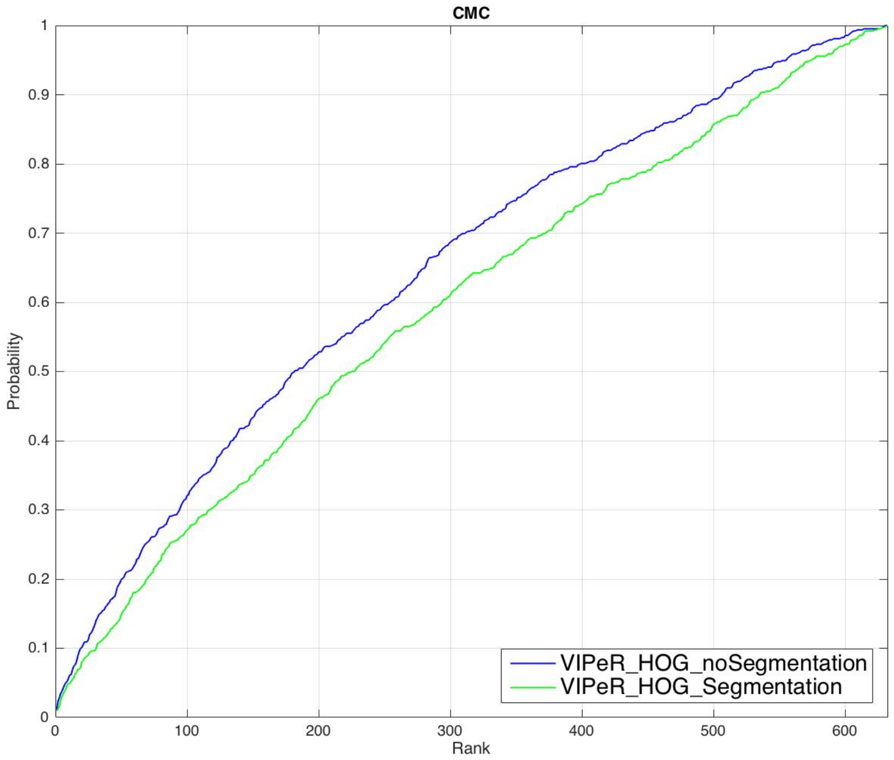

FIGURE 3.10: A CMC comparison of foreground segmentation on HOG feature tested on VIPeR

## 3.3 The hierarchical Gaussian descriptor

The hierarchical Gaussian descriptor is proposed by in [25]. This descriptor uses a two-level Gaussian distribution, i.e., a Gaussian of Gaussian (GOG) structure, to model an individual. This descriptor densely samples the image and models samples (including the square patches and the slice made up of those overlapping patches) in each level respectively with a Gaussian distribution to outperform many other works. In this thesis, all input images are sized to $128 \times 64$. First, the image is divided into 7 overlapping horizontal slices. Each slide has a size of $32 \times 64$ and slides are overlapping by $16$ pixels vertically. In each slide, densely sampled square patches have a size of $s \times s$ pixels ($s$ = 5 in this thesis), and small patches overlap each other by $2$ pixels. So there is a two-level structure in this image, small patches and slides. The small patches are first modelled with a multivariate Gaussian distribution. Then the slide with those small patch Gaussian distributions is modelled with another multivariate Gaussian distribution.

A certain horizontal slice for each small patch is modelled by a multivariate Gaussian distribution $G_p(\boldsymbol{f}_i; \boldsymbol{\mu}_p, \Sigma_p)$, and $G_p$ can be transformed to a vector $\boldsymbol{p}$ by SPD mapping. Again, when all small patches are modelled and vectorized, the same process is repeated. Each slide is characterized by a multivariate Gaussian distribution $\boldsymbol{G}_r(\boldsymbol{p}; \boldsymbol{\mu}_r, \Sigma_r)$.

After the slide is modelled with $\boldsymbol{G}_r(\boldsymbol{p}; \boldsymbol{\mu}_r, \Sigma_r)$, the same transformation will be operated on $\boldsymbol{G}_r$ so that it is vectorized as a vector $\boldsymbol{v}$. At last, all computed $\boldsymbol{v}$ are concatenated to consist of the descriptor of the current image.

### 3.3.1 Handling the background

In the previous section, the impact of background subtraction on different features' performances have been studied. We've concluded that imperfect foreground segmentation decreases textural feature's performance. Besides, in the hierarchical Gaussian descriptor, there is no histogram-based feature computing. So in this thesis, when computing the pixel basic feature $\boldsymbol{f}_i$, the foreground segmentation is not adopted. But when modelling the region Gaussian $\boldsymbol{G}_r(\boldsymbol{p}; \boldsymbol{\mu}_r, \Sigma_r)$, a weighted map is computed for each patch with equation

$$N(x; \mu_0, \sigma_0) = \frac{1}{\sigma_0\sqrt{2\pi}} \exp^{\frac{(x-\mu_0)^2}{\sigma_0^2}}, \tag{3.5}$$

here $\mu_0 = \frac{W_0}{2}, \sigma_0 = \frac{W_0}{4}$, $W_0$ is the number of patches in horizontal direction.

### 3.3.2 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 an 8-dimensional vector $\boldsymbol{f}_i$, 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 components are the color values of the specified color space.

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

The $M$ is to characterize the texture with the gradient histogram. Different $M$ values are the magnitudes of the gradient in every direction. First, the gradients in x and y direction are computed by two gradient filters $\mathrm{h_x}$ and $\mathrm{h_y}$, and we have

$$\begin{aligned} h_x &= [-1, 0, 1], \\ h_y &= -h'_x. \end{aligned} \tag{3.6}$$

Then, by convolving those two filters with the intensity image $I$, the horizontal and vertical gradient $I_x, I_y$ can be computed, so the orientation and magnitude can be computed by the following equations:

$$\begin{aligned} O(i,j) &= (\arctan(\frac{I_y(i,j)}{I_x(i,j)} + \pi) * 180/\pi, \\ M(i,j) &= \sqrt{(I_x(i,j)^2 + I_y(i,j))^2}. \end{aligned} \tag{3.7}$$

The orientations are quantized into four bins by a soft voting algorithm [31]. For each pixel, its corresponding gradient orientation is decided by its nearest bin's direction. To make the descriptor focus on the gradient components with high values, the gradient and orientation are multiplied as follows:

$$M_\theta = MO_\theta. \tag{3.8}$$

To model the patch with a multivariate Gaussian distribution, we have to estimate its mean value and the covariance matrix. A multivariate Gaussian model has the form

$$G_p(\boldsymbol{f}_i; \boldsymbol{\mu}_p, \boldsymbol{\Sigma}_p) = \frac{\exp^{(\frac{1}{2}(\boldsymbol{f}_i - \boldsymbol{\mu}_p)^T \boldsymbol{\Sigma}_{\boldsymbol{p}}^{-1} (\boldsymbol{f}_i - \boldsymbol{\mu}_p))}}{(2\pi)^{d/2} |\boldsymbol{\Sigma}_p|}, \tag{3.9}$$

where $\boldsymbol{\mu}_p$ is the estimated mean value, and $\boldsymbol{\Sigma}_p$ is the estimated covariance matrix of the current small patch.

To estimate the parameters for this Gaussian model based on basic pixel features, the maximal likelihood estimate (MLE) is used. According the MLE algorithm, we have the following estimated parameters:

$$\boldsymbol{\mu}_p = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{f}_i, \tag{3.10}$$

$$\boldsymbol{\Sigma}_p = \frac{1}{n-1}\sum_{i=1}^{n}(\boldsymbol{f}_i - \boldsymbol{\mu})(\boldsymbol{f}_i - \boldsymbol{\mu})^T, \tag{3.11}$$

where $n$ is the number of pixels in the current patch. When the Gaussian model is computed, the next step is to model all the patch Gaussians. But it is a complex problem to directly model those multivariate Gaussian functions. So some transformation will be operated on the estimated parameters $\boldsymbol{\mu}_p$ and $\boldsymbol{\Sigma}_p$.

### 3.3.3 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 cannot be directly finished on previous estimated Gaussian functions. A transformation is needed to make stochastic summarization feasible on patch Gaussian 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 by Equation 3.12. A $d$-dimensional multivariate Gaussian function can be mapped into a $d+1$-dimensional $SPD_+$ space. According to [23], the mapping can be denoted as

$$G(\boldsymbol{x}_i; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \sim \boldsymbol{P}_i = |\boldsymbol{\Sigma}_i|^{1/(d+1)} \begin{bmatrix} \boldsymbol{\Sigma}_i + \boldsymbol{\mu}_i\boldsymbol{\mu}^T & \boldsymbol{\mu}_i \\ \boldsymbol{\mu}_i^T & 1 \end{bmatrix}. \tag{3.12}$$

The covariance matrix $\boldsymbol{\Sigma}_i$ can be singular due to the small number of pixels within the patch. To avoid this problem, a regular factor $\lambda$ 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 into 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, it can be compressed by half, i.e., only the upper triangular elements are preserved. To ensure its norm-1 remains 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

$$\boldsymbol{p}_i = [\boldsymbol{Q}_{1,1}, \sqrt{2}\boldsymbol{Q}_{1,2}, \sqrt{2}\boldsymbol{Q}_{1,3}, \cdots, \sqrt{2}\boldsymbol{Q}_{1,d+1}, \tag{3.13}$$

$$\boldsymbol{Q}_{2,2}, \sqrt{2}\boldsymbol{Q}_{2,3,}, \cdots, \sqrt{2}\boldsymbol{Q}_{2,d+1,}, \cdots, \boldsymbol{Q}_{d+1,d+1,}]. \tag{3.14}$$

Again, we model the slide with a multivariate Gaussian distribution $\boldsymbol{G}_r$ by the equation

$$G_p(\boldsymbol{p}_j; \boldsymbol{\mu}_r, \boldsymbol{\Sigma}_r) = \frac{\exp^{(\frac{1}{2}(\boldsymbol{p}_j-\boldsymbol{\mu}_r)^T \boldsymbol{\Sigma}_r^{-1}(\boldsymbol{p}_j-\boldsymbol{\mu}_r))}}{(2\pi)^{d/2}|\boldsymbol{\Sigma}_r|}. \tag{3.15}$$

We have known that all patches inside this slide are represented as vector $\boldsymbol{p}_j$, thus the mean $\boldsymbol{\mu}_r$ and covariance matrix $\boldsymbol{\Sigma}_r$ of $\boldsymbol{G}_r$ can be computed by the MLE with the equations

$$\boldsymbol{\mu}_r = \frac{1}{m}\sum_{j=1}^{m} \boldsymbol{p}_j, \tag{3.16}$$

$$\boldsymbol{\Sigma}_r = \frac{1}{m-1}\sum_{j=1}^{m} (\boldsymbol{p}_j - \boldsymbol{\mu}_r)(\boldsymbol{p}_j - \boldsymbol{\mu}_r)^T, \tag{3.17}$$

where m is the total number pf patches inside the current slide. Again, $\boldsymbol{\mu}_r$ and $\boldsymbol{\Sigma}_r$ are mapped to an SPD space by the following equation

$$G(\boldsymbol{p}; \boldsymbol{\mu}_r, \boldsymbol{\Sigma}_r) \sim \boldsymbol{P}_r = |\boldsymbol{\Sigma}_r|^{1/(d+1)} \begin{bmatrix} \boldsymbol{\Sigma}_r + \boldsymbol{\mu}_r\boldsymbol{\mu}_r^T & \boldsymbol{\mu}_r \\ \boldsymbol{\mu}_r^T & 1 \end{bmatrix}, \tag{3.18}$$

and $\boldsymbol{P}_r$ is vectorized by Equation 3.13.

When the descriptors of all horizontal slides are computed they are concatenated to form the descriptor for the whole image.

### 3.3.4 Integral image for fast region covariance computation

To compute estimated parameters for all those overlapping small patches, the time complexity of computing each patch one by one is high because there are many repeating computations. To compute the estimated covariance matrix $\boldsymbol{\Sigma}$ for every small patch with a size of $W \times H$, the integral image is used to reduce time complexity. The integral image [40] is an intermediate representation used to quickly compute the rectangle area sum in an image. Each pixel value in the integral image is the sum of all the pixels inside the rectangle bounded by the current pixel and the upper left pixel. That is, the integral image S(x, y) for image I(x, y) is

$$S(x', y') = \sum_{x<x', y<y'} I(x, y). \tag{3.19}$$

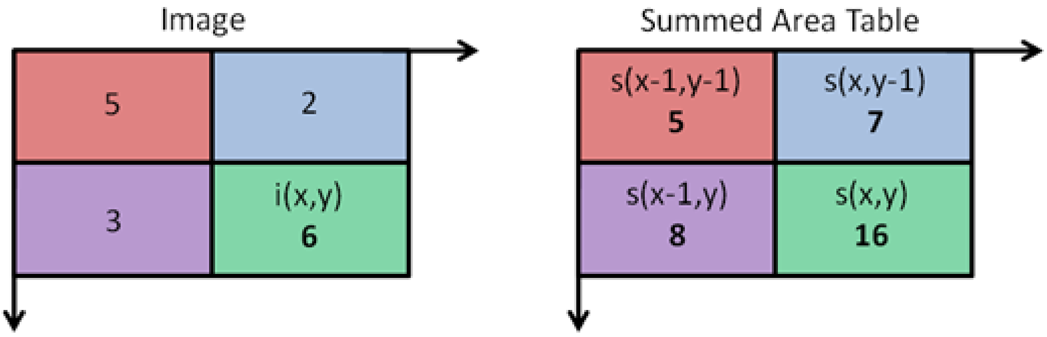

FIGURE 3.11: Integral image

By using the integral image, any rectangular region sum can be computed in constant time.

Suppose $\boldsymbol{I}_F$ is a $W \times H \times d$ feature tensor of integral images of $F$, to compute the covariance matrix of a certain rectangle area in $F$, we have the equation:

$$\boldsymbol{I}_F(x', y', i) = \sum_{x<x', y<y'} F(x, y, i), i = 1 \ldots d, \tag{3.20}$$

Suppose the $\boldsymbol{C}(x', y', i, j)$ is the $W \times H \times d \times d$ tensor of second order integral images, and we use the following equation:

$$\boldsymbol{C}(x', y', i, j) = \sum_{x<x', y<y'} F(x, y, i)F(x, y, j), i, j = 1 \ldots d. \tag{3.21}$$

Let $\boldsymbol{I}_{x,y}$ be the $d$-dimensional vector in $\boldsymbol{I}_F$, and $\boldsymbol{C}(x, y)$ be the $d \times d$-dimensional matrix in $\boldsymbol{C}$,

$$\begin{gathered} \boldsymbol{I}_{x,y} = [\boldsymbol{I}_F(x', y', 1) \ldots \boldsymbol{I}_F(x', y', d)]^T, \\ \boldsymbol{C}_{x,y} = \begin{bmatrix} \boldsymbol{C}(x, y, i, 1) & \cdots & \boldsymbol{C}(x, y, 1, d) \\ & \ddots & \\ \boldsymbol{C}(x, y, d, 1) & \cdots & \boldsymbol{C}(x, y, d, d) \end{bmatrix}. \end{gathered} \tag{3.22}$$

Then for any rectangule regions $R(x', y'; x'', y'')$, where $(x', y')$ is the upper left coordinate and $(x'', y'')$ is the lower right coordinate, the covariance matrix can be computed as

$$\begin{aligned} \boldsymbol{C}_R(x', y'; x'', y'') = \frac{1}{n-1}[\boldsymbol{C}_{x'',y''} + \boldsymbol{C}_{x',y'} - \boldsymbol{C}_{x'',y'} - \boldsymbol{C}_{x',y''} \\ -\frac{1}{n}(\boldsymbol{I}_{x'',y''} + \boldsymbol{I}_{x'',y''} - \boldsymbol{I}_{x',y''} - \boldsymbol{I}_{x'',y'})(\boldsymbol{I}_{x'',y''} + \boldsymbol{I}_{x'',y''} - \boldsymbol{I}_{x',y''} - \boldsymbol{I}_{x'',y'})^T], \end{aligned} \tag{3.23}$$

where $n$ is the number of feature vector in $F$, and $n = (x''-x')(y''-y')$. By creating the integral image, the covariance of any rectangular area in $F$ can be computed in $O(d^2)$ time.

When all patches in a region are computed, the same process is repeated to compute the region Gaussian.

### 3.3.5 Some variants of Hierarchical Gaussian descriptor

In the hierarchical Gaussian descriptor, the basic pixel feature $\boldsymbol{f}_i$ characterizes texture, color and the $y$ coordinate of an image. The importance of those three characteristics is ordered as color values > gradient components > $y$ coordinate. A rank-1, rank-5, rank-10, and rank-20 comparison of three GOG$_{\text{fusion}}$ variants on the VIPeR dataset are listed below. We can conclude that color values have the most influence on Re-ID accuracy. Based on this conclusion, in GOG$_{\text{fusion}}$ variants studied, the color values will remain unchanged and the $y$ coordinate and M components will be replaced or removed. In one GOG$_{\text{fusion}}$ variant, we replace the M components in $\boldsymbol{f}_i$ with LBP.

TABLE 3.1: A rank score comparison of GOG$_{\text{fusion}}$ Variants

| | Rank(%) | | | |
|---|---|---|---|---|
| GOG$_{\text{fusion}}$ Variants | 1 | 5 | 10 | 20 |
| GOG$_{\text{fusion}}$ | 47.97 | 77.44 | 86.80 | 93.70 |
| GOG$_{\text{fusion}}$ without $y$ | 44.80 | 76.20 | 86.80 | 93.20 |
| GOG$_{\text{fusion}}$ without M | 41.10 | 70.50 | 81.10 | 90.30 |
| GOG$_{\text{fusion}}$ without color values | 8.40 | 21.00 | 31.70 | 45.50 |

Another variant is to combine superpixel segmentation with GOG$_{\text{fusion}}$. Superpixel algorithms cluster image pixels into perceptually atomic regions, which can be used to reduce redundancy and computation complexity. One successful superpixel algorithm is Simple Linear Iterative Clustering (SLIC) [1]. To combine superpixel segmentation and GOG$_{\text{fusion}}$, each horizontal slice of an image is first segmented by the SLIC algorithm into many nonrigid groups. An example of SLIC segmentation is shown in Figure 3.12. Each superpixel group is modelled with a multivariate Gaussian function by summarizing the basic feature $\boldsymbol{f}_i$ of pixels inside this group. With similar SPD mapping and vectorization, every group can be summarized by a vector. At last, the same process is repeated on those superpixels to get region Gaussians. Compared with the original hierarchical Gaussian descriptor, overlapping square patch sampling in the hierarchical Gaussian is changed to nonrigid and non-overlapping superpixel sampling in the variant. Hopefully, superpixel segmentation can group pixels with similar properties, which may decrease errors caused by single Gaussian modelling.

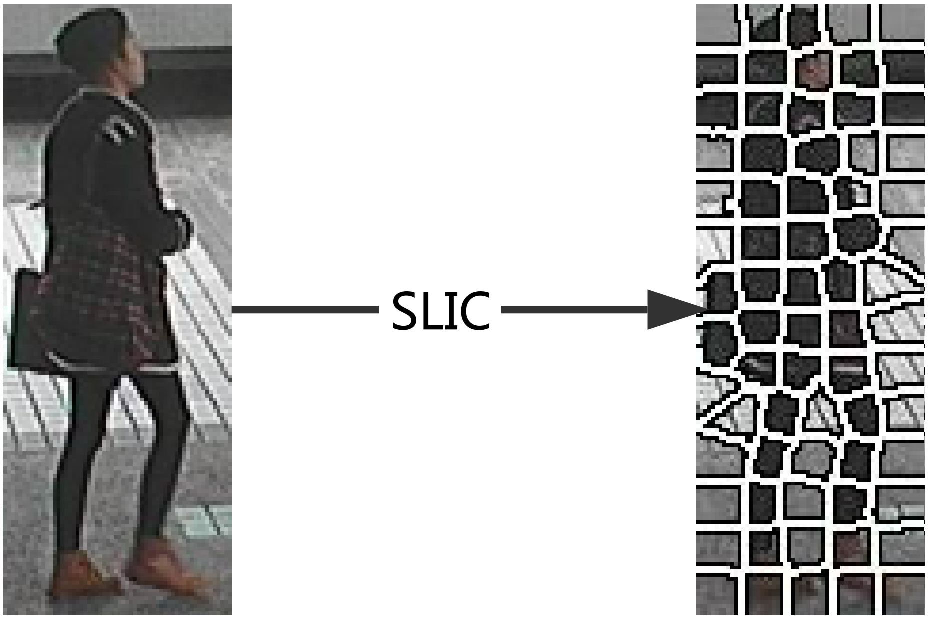

FIGURE 3.12: A SLIC superpixel segmentation example

Besides, a Gaussian mixture model (GMM) based descriptor is also tested. GMM models an image or a region with a mixture of Gaussian distributions. But it is computationally expensive to compute the similarity of GMMs. Also, it is extremely difficult to create a two-level hierarchical GMM model. A sparse representation based Earth Mover's Distance (SR-EMD) [21] is used here to measure similarity. Performances of those variants have been listed in Table 3.2. It can be concluded that

GOG$_{\text{fusion}}$ has the best performance. GOG$_{\text{fusion}}$MreplacedByLBP's performance decreases heavily because LBP components in $\boldsymbol{f}_i$ are less robust. GOG$_{\text{fusion}}$Superpixel might suffer from its non-overlapping superpixel patch sampling. The GMM + SR-EMD model has the worst performance for two reasons: (1) This model is not trained; (2) A better similarity measurement is needed to for GMMs.

TABLE 3.2: A performance comparison between Gaussian of Gaussian descriptor and its variants on the VIPeR dataset

| | Rank(%) | | | | |
|---|---|---|---|---|---|
| Variant descriptors | 1 | 5 | 10 | 15 | 20 |
| GOG$_{\text{fusion}}$+XQDA | 47.97 | 77.44 | 86.80 | 91.27 | 93.70 |
| GOG$_{\text{fusion}}$MreplacedByLBP+XQDA | 40.70 | 72.53 | 83.16 | 88.45 | 91.90 |
| GOG$_{\text{fusion}}$Superpixel+XQDA | 42.72 | 74.84 | 85.22 | 89.34 | 92.25 |
| GMM + SR-EMD(EMD-theta) | 11.40 | 21.50 | 29.70 | 37.00 | 42.40 |

### 3.3.6 Dimensionality and superiority analysis of hierarchical Gaussian descriptor

It has been known that the combination of descriptors of different color spaces 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 spaces {HSV, Lab, nRGB} are extracted. The nRGB color space is calculated as

$$\begin{aligned} nR &= \frac{R}{R+G+B}, \\ nG &= \frac{G}{R+G+B}, \\ nR &= \frac{B}{R+G+B}. \end{aligned} \tag{3.24}$$

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 spaces {RGB, HSV, Lab, nRnG}, the corresponding dimension of the pixel feature is {8, 8, 8, 7}. After the matrix to vector transformation, the dimension of the 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 would be $\{7567, 7567, 7567, 4921\}$. If 4 color spaces are all used, the dimension is the sum of each channel as 27622.

The Hierarchical Gaussian descriptor has a few advantages compared with other descriptors. First, it has a full consideration of color, texture and $y$ coordinate information. The color information is adopted by adding color values of different color space to $\boldsymbol{f}_i$. Textural information is also adopted by those four gradient components in $\boldsymbol{f}_i$. Second, by Riemannian manifold based SPD mapping, it provides a solution to summarize many multivariate Gaussian functions. The parameters $\boldsymbol{\mu}$ and $\Sigma$ of a multivariate Gaussian function can be fused into an SPD matrix. With later vectorization, this SPD matrix is transformed to a vector. In this mapping process, correlation of different components in $\boldsymbol{f}_i$ are fully taken into account, which leads to a dimension increasing from basic pixel feature vector $\boldsymbol{f}_i$ to patch Gaussian vector $\boldsymbol{p}_j$. Third, by a two-level Gaussian model, this model guarantees to extract the overall statistical information of an image while being robust to local details variation caused by factors like viewpoint changes and partial occlusion.

It is intractable to directly learn a Mahalanobis distance matrix from the high-dimensional concatenated Hierarchical Gaussian descriptor. In the next chapter, KLFDA is applied to reduce the dimensionality of the extracted descriptor from 27622 to $n - 1$, where $n$ is the number of different classes. With this supervised non-linear dimensionality reduction, discriminative information among different classes is preserved. At last, a Mahalanobis matrix is learned on the dimension-reduced descriptors by the gradient descent method.

# Chapter 4

# Dimension reduction and Mahalanobis distance learning

To deal with high-dimensional descriptors, dimension reduction is first performed by kernel local Fisher discriminant analysis (KLFDA). Then Mahalanobis distance metric learning based on limitations between interclass and intraclass distance is applied on dimension-reduced data.

## 4.1 Kernel local Fisher discriminant analysis

The extracted hierarchical Gaussian descriptors have a high dimension, and it is intractable to learn an SPD matrix with such high dimension. Dimension reduction is required to learn a subspace. Among those methods to reduce dimension, principal component analysis (PCA) is often used. However, PCA is an unsupervised dimension reduction and may have low performance for two reasons. $(1)$, PCA is to maximize the variance of dimension-reduced data, and as an unsupervised method it doesn't take into consideration the between-class and within-class relation. It is very likely that the descriptors of different classes can be mixed up after dimension reduction. $(2)$ PCA may suffer from the small sample size problem. In some Re-ID datasets, there may be one or two images for each pedestrian in each viewpoint (like VIPeR). If the dimension of the descriptor is much bigger than the sample size, information can be lost with PCA. In this thesis, the kernel local Fisher discriminant analysis (KLFDA) is used to reduce dimension.

KLFDA is the kernel version of LFDA, and LFDA is a combination of the Fisher discriminant analysis [33], the locality preserving projection [17] and the kernel method. A brief introduction of the FDA, LPP and kernel method is given below.

### 4.1.1 Fisher discriminant analysis (FDA)

FDA is a supervised dimension reduction, and its input contains the class labels. Given a set of $d$-dimensional observations $\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_i, \cdots, \boldsymbol{x}_n)$, where $i \in \{1, 2, \cdots, n\}$, the label $y_i \in \{1, 2, \cdots, l\}$, for each sample descriptor $\boldsymbol{x}_i$, a linear transformation with transformation matrix $\boldsymbol{T}$ can be defined by the equation

$$\boldsymbol{z}_i = \boldsymbol{T}^T \boldsymbol{x}_i. \tag{4.1}$$

$\boldsymbol{T}$ has a dimension of $d \times m$, and $\boldsymbol{z}_i$ is the $m(m < d)$ -dimensional vector. In FDA, two matrices are defined as the within-class scatter matrix $\boldsymbol{S}^{(w)}$ and between-class scatter matrix $\boldsymbol{S}^{(b)}$,

$$\begin{aligned} \boldsymbol{S}^{(w)} &= \sum_{i=1}^{l} \sum_{j:y_j=i} (\boldsymbol{x}_j - \boldsymbol{\mu}_i)(\boldsymbol{x}_j - \boldsymbol{\mu}_i)^T, \\ \boldsymbol{S}^{(b)} &= \sum_{i=1}^{l} n_i (\boldsymbol{\mu}_i - \boldsymbol{\mu})(\boldsymbol{\mu}_i - \boldsymbol{\mu})^T, \end{aligned} \tag{4.2}$$

where $n_i$ is the number of classes with class label $i$, $\boldsymbol{\mu}_i$ is the mean of samples whose label is $i$, and $\boldsymbol{\mu}$ is the mean of all samples,

$$\begin{aligned} \boldsymbol{\mu}_i &= \frac{1}{n_i} \sum \boldsymbol{x}_i, \\ \boldsymbol{\mu} &= \frac{1}{n} \sum \boldsymbol{x}_i. \end{aligned} \tag{4.3}$$

The Fisher discriminant analysis transform matrix $\boldsymbol{T}$ can be represented as

$$\boldsymbol{T} = \arg\max \frac{\boldsymbol{T}^T \boldsymbol{S}^{(b)} \boldsymbol{T}}{\boldsymbol{T}^T \boldsymbol{S}^{(w)} \boldsymbol{T}}. \tag{4.4}$$

This equation can be solved by the Lagrange multiplier method. We define a Lagrange function as:

$$L(\boldsymbol{t}) = \boldsymbol{t}^T \boldsymbol{S}^{(b)} \boldsymbol{t} - \lambda(\boldsymbol{t}^T \boldsymbol{S}^{(w)} \boldsymbol{t} - 1). \tag{4.5}$$

Then the differential respect to $\boldsymbol{t}$ is

$$\frac{\partial L(\boldsymbol{t})}{\partial \boldsymbol{t}} = 2\boldsymbol{S}^{(b)} \boldsymbol{t} - 2\lambda \boldsymbol{S}^{(w)} \boldsymbol{t}, \tag{4.6}$$

let

$$\frac{\partial L(\boldsymbol{t})}{\partial \boldsymbol{t}} = 0, \tag{4.7}$$

and we can get

$$\boldsymbol{S}^{(b)}\boldsymbol{t}_i = \lambda \boldsymbol{S}^{(w)}\boldsymbol{t}_i. \tag{4.8}$$

Here $\boldsymbol{t}_i$ is the $i_{th}$ column of $\boldsymbol{T}$, and the optimization problem is converted to an eigenvalue decomposition problem. $\boldsymbol{T}$ is the set of eigenvectors of $\frac{\boldsymbol{S}^{(b)}}{\boldsymbol{S}^{(w)}}$.

The Fisher discriminant analysis tries to minimize the intraclass scatter matrix while maximizing the interclass scatter matrix. $\boldsymbol{T}$ is computed by the eigenvalue decomposition. $\boldsymbol{T}$ can be represented as the set of all the corresponding eigenvectors, as $\boldsymbol{T} = (\boldsymbol{t}_1, \boldsymbol{t}_2, \cdots, \boldsymbol{t}_k)$.

FDA has a form similar to signal and noise ratio; however, the FDA dimension reduction may have poor performance because it doesn't consider the locality of data. An example of this is the multimodality [38]. Multimodality is when many clusters are formed in the same class.

### 4.1.2 Locality preserving projection (LPP)

In [17], locality preserving projection (LPP) is proposed to exploit data locality. 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 an $n \times n$ affinity matrix $\boldsymbol{A}$; usually two sample points with a smaller distance have a higher affinity value than those with a bigger distance value. One example is that 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 $\boldsymbol{D}$ can be defined so that each diagonal element is the sum of the corresponding column in $\boldsymbol{A}$:

$$\boldsymbol{D}_{i,i} = \sum_{j=1}^{n} \boldsymbol{A}_{i,j}. \tag{4.9}$$

Then the LPP transform matrix is defined as follow:

$$\boldsymbol{T}_{LPP} = \underset{\boldsymbol{T}\in\boldsymbol{R}^{d\times m}}{\arg\min} \frac{1}{2} \sum_{i,j=1}^{n} \boldsymbol{A}_{i,j} ||\boldsymbol{T}^T\boldsymbol{x}_i - \boldsymbol{T}^T\boldsymbol{x}_j||, \tag{4.10}$$

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 the LPP transform matrix $T$ can be represented as:

$$\boldsymbol{T}_{LPP} = \{\boldsymbol{\phi}_{d-m+1}|\boldsymbol{\phi}_{d-m+2}|\cdots\boldsymbol{\phi}_d\}, \tag{4.11}$$

and each $\boldsymbol{\phi}$ in $T$ is the eigenvector of the following formula,

$$\boldsymbol{X}\boldsymbol{L}\boldsymbol{X}^T\boldsymbol{\phi} = \gamma\boldsymbol{X}\boldsymbol{D}\boldsymbol{X}^T, \tag{4.12}$$

where $\gamma$ is corresponding eigenvalue of $\boldsymbol{\phi}$, and $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}$.

### 4.1.3 Local Fisher discriminant analysis (LFDA)

LFDA [33] combines FDA and LPP and has better performance. The key to LFDA is that it assigns weights to elements in $\boldsymbol{A}^{(w)}$ and $\boldsymbol{A}^{(b)}$, so that

$$\begin{aligned}\boldsymbol{S}^{(w)} &= \frac{1}{2}\sum_{i=1}^{l}\sum_{j:y_j=i}\boldsymbol{A}^{w}_{i,j}(\boldsymbol{x}_j-\boldsymbol{\mu}_i)(\boldsymbol{x}_j-\boldsymbol{\mu}_i)^T,\\ \boldsymbol{S}^{(b)} &= \frac{1}{2}\sum_{i=1}^{l}\boldsymbol{A}^{b}_{i,j}(\boldsymbol{\mu}_i-\boldsymbol{\mu})(\boldsymbol{\mu}_i-\boldsymbol{\mu})^T,\end{aligned} \tag{4.13}$$

where

$$\begin{aligned}\boldsymbol{A}^{(w)}_{i,j} &= \begin{cases}\boldsymbol{A}_{i,j}/n_c & y_i = y_j\\ 0 & y_i \neq y_j\end{cases},\\ \boldsymbol{A}^{(b)}_{i,j} &= \begin{cases}(\frac{1}{n}-\frac{1}{n_c})\boldsymbol{A}_{i,j} & y_i = y_j\\ \frac{1}{n} & y_i \neq y_j\end{cases},\end{aligned} \tag{4.14}$$

where $y_i$ is the class label of sample point $\boldsymbol{x}_i$. So the transformation matrix $T_{LFDA}$ can be computed by the equation

$$\boldsymbol{T}_{LFDA} = \arg\min_{\boldsymbol{T}}(\frac{\boldsymbol{T}^T\boldsymbol{S}^{(b)}\boldsymbol{T}}{\boldsymbol{T}^T\boldsymbol{S}^{(w)}\boldsymbol{T}}). \tag{4.15}$$

Again, this problem can be solved by eigenvalue decomposition with Equation 4.8.

When applying the LFDA to original high-dimensional descriptors, one problem is the computational cost. Suppose the vector data has a dimension of $d$, then LFDA has to solve the eigenvalues of a matrix with a dimension $d \times d$. In some descriptors, $d$ could be more than 20000 and the computation cost is intractable.

### 4.1.4 Kernel local Fisher discriminant analysis (KLFDA)

KLFDA [38] is the nonlinear version of LFDA. Most dimensionality reduction methods, including PCA, LDA and LFDA, are linear dimensionality reduction methods. However, when descriptor data is non-linear in feature space, it's hard to capture its between-class discriminant information with linear reduction methods. One alternative method is to nonlinearly map input descriptors $\boldsymbol{x}_i$ to higher-dimensional feature space $\Phi$ by a function $\phi(\boldsymbol{x}_i)$. Again, the LFDA is performed in feature space $\Phi$. Thus the transformation matrix $\boldsymbol{T}$ can be computed by the equation

$$\boldsymbol{T} = \arg\min \frac{\boldsymbol{T}^T \boldsymbol{S}_{\phi}^{(b)} \boldsymbol{T}}{\boldsymbol{T}^T \boldsymbol{S}_{\phi}^{(w)} \boldsymbol{T}}, \tag{4.16}$$

where $\boldsymbol{S}_{\phi}^{(b)}$ and $\boldsymbol{S}_{\phi}^{(w)}$ is the between-class scatter and within-class scatter in mapped feature space $\Phi$.

Note that with the transformation matrix $\boldsymbol{T} \in \Phi$, it is computationally expensive to explicitly compute the mapping function $\phi$ and perform LFDA in feature space $\Phi$ because the dimension of $\Phi$ may be infinite. Rather than explicitly computing, the mapping function $\phi$ can be implicit, and the feature space $\Phi$ can be defined by the inner product of features in $\Phi$. The kernel trick is used here, and a kernel function can be defined as the inner product of mapped vectors $\phi(\boldsymbol{x}_i)$ and $\phi(\boldsymbol{x}_j)$ by the equation

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) =< \phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_j) >, \tag{4.17}$$

where the $< \cdot >$ is the inner product. There are many kinds of kernels like the linear kernel, the polynomial kernel and the radial basis function (RBF) kernel. In this paper, the RBF kernel is adopted. An RBF kernel is defined as

$$k_{RBF}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp^{(-\gamma ||\boldsymbol{x}_i - \boldsymbol{x}_j||^2)}. \tag{4.18}$$

Suppose $\boldsymbol{X}$ is the matrix of sample descriptors, and we have

$$\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n), \tag{4.19}$$

and the label vector is $\boldsymbol{l} = (l_1, l_2, \cdots, l_n)$. Then the kernel matrix of $\boldsymbol{X}$ can be computed as the following equation:

$$\boldsymbol{K} = \phi(\boldsymbol{X})^T \phi(\boldsymbol{X}), \tag{4.20}$$

and we have

$$\boldsymbol{K}_{i,j} = k(\boldsymbol{x}_i, \boldsymbol{x}_j) =< \phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_j) >= \exp^{(-\gamma ||\boldsymbol{x}_i - \boldsymbol{x}_j||^2)}. \tag{4.21}$$

In [38], the authors proposed fast computation of LFDA by replacing $\boldsymbol{S}^{(b)}$ with the local scatter mixture matrix $\boldsymbol{S}^{(m)}$defined by

$$\begin{aligned} \boldsymbol{S}^{(m)} &= \boldsymbol{S}^{(b)} + \boldsymbol{S}^{(w)}, \\ \boldsymbol{S}^{(m)} &= \frac{1}{2} \sum_{i,j=1} \boldsymbol{A}^{(m)}_{i,j} (\boldsymbol{x}_i - \boldsymbol{x}_j)(\boldsymbol{x}_i - \boldsymbol{x}_j)^T, \end{aligned} \tag{4.22}$$

and

$$\boldsymbol{A}^{(m)}_{i,j} = \boldsymbol{A}^{(w)}_{i,j} + \boldsymbol{A}^{(w)}_{i,j}. \tag{4.23}$$

According to the identity (cf. (Fukunaga, 1990))

$$trace((\boldsymbol{T}^T \boldsymbol{S}^{(w)} \boldsymbol{T})^{(-1)} (\boldsymbol{T}^T \boldsymbol{S}^{(m)} \boldsymbol{T}) = trace((\boldsymbol{T}^T \boldsymbol{S}^{(w)} \boldsymbol{T})^{(-1)} (\boldsymbol{T}^T \boldsymbol{S}^{(b)} \boldsymbol{T}) + m. \tag{4.24}$$

Equation 4.15 is equal to

$$\boldsymbol{T}_{LFDA} = \arg\min_{\boldsymbol{T}} (\frac{\boldsymbol{T}^T \boldsymbol{S}^{(m)} \boldsymbol{T}}{\boldsymbol{T}^T \boldsymbol{S}^{(w)} \boldsymbol{T}}), \tag{4.25}$$

and it can be transformed into a eigenvalue decomposition problem

$$\boldsymbol{S}^{(m)} \boldsymbol{t}_i = \lambda \boldsymbol{S}^{(w)} \boldsymbol{t}_i. \tag{4.26}$$

With the replacement of $\boldsymbol{S}^{(m)}$, in [38] the author summarized that

$$\boldsymbol{S}^{(m)} = \boldsymbol{X} \boldsymbol{L}^{(m)} \boldsymbol{X}^T, \tag{4.27}$$

where $\boldsymbol{L}^{(m)} = \boldsymbol{D}^{(m)} - \boldsymbol{A}^{(m)}$, and $\boldsymbol{D}^{i,i} = \sum_{j=1}^{n} \boldsymbol{A}_{i,j}^{(m)}$. Also, $\boldsymbol{S}^{(w)}$ can be represented as

$$\boldsymbol{S}^{(w)} = \boldsymbol{X}\boldsymbol{L}^{(w)}\boldsymbol{X}^T, \tag{4.28}$$

where $\boldsymbol{L}^{(w)} = \boldsymbol{D}^{(w)} - \boldsymbol{A}^{(m)}$, and $\boldsymbol{D}^{i,i} = \sum_{j=1}^{n} \boldsymbol{A}_{i,j}^{(w)}$. Therefore, Equation 4.26 can be represented as

$$\boldsymbol{X}\boldsymbol{L}^{(m)}\boldsymbol{X}^T\boldsymbol{t}_i = \lambda\boldsymbol{X}\boldsymbol{L}^{(w)}\boldsymbol{X}^T\boldsymbol{t}_i. \tag{4.29}$$

The eigenvector $\boldsymbol{t}_i$ can be represented as $\boldsymbol{t}_i = \boldsymbol{X}\boldsymbol{\gamma}$, vector $\boldsymbol{\gamma}_i \in R^n$. With this replacement, we left multiply $\boldsymbol{X}^T$ to Equation 4.29 to get

$$\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{L}^{(m)}\boldsymbol{X}^T\boldsymbol{X}\gamma_i = \lambda\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{L}^{(w)}\boldsymbol{X}^T\boldsymbol{X}\gamma_i, \tag{4.30}$$

and with the kernel trick, it's represented as

$$\boldsymbol{K}\boldsymbol{L}^{(m)}\boldsymbol{K}\gamma_i = \lambda\boldsymbol{K}\boldsymbol{L}^{(w)}\boldsymbol{K}\gamma_i. \tag{4.31}$$

One example of using KLFDA to reduce dimension and classify the nonlinear data clusters can be shown in Figures 4.1, 4.2 and 4.3. Three classes with five clusters are distributed on a 2D plane, and by KLFDA dimension reduction their 1D dimension-reduced data distribution are shown in Figure 4.2 and Figure 4.3. We can conclude that for those clusters the Gaussian kernel is better than the linear kernel because the dimension-reduced data is more separate when using Gaussian kernel function.

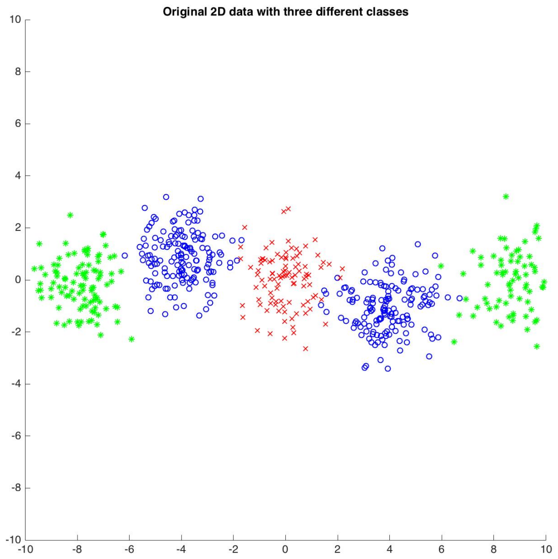

FIGURE 4.1: Example of five clusters that belong to three classes

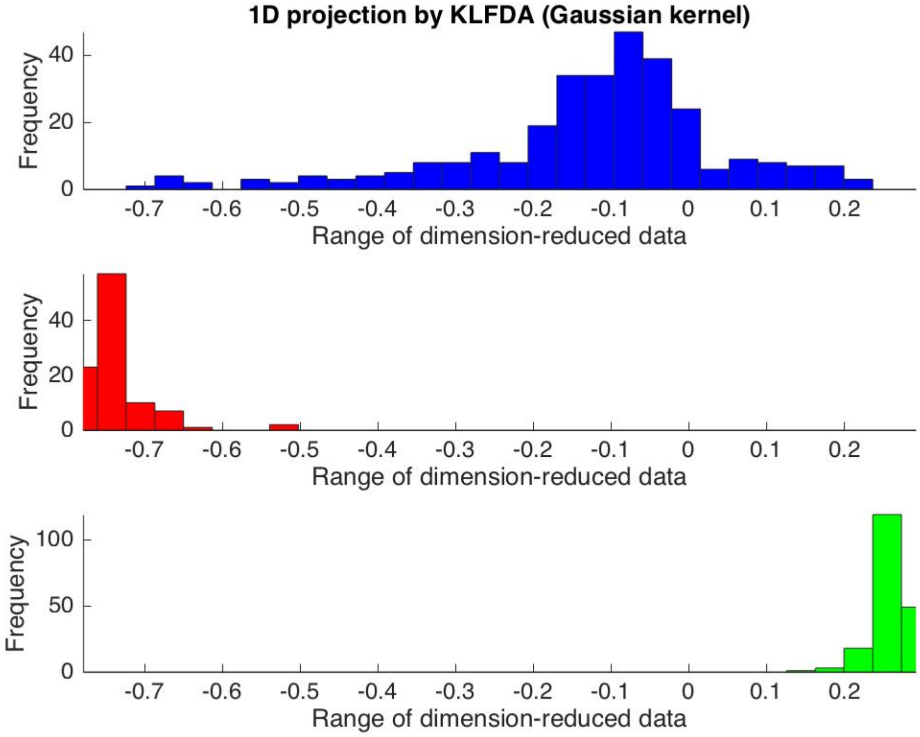

FIGURE 4.2: 1D distribution of dimension-reduced data with Gaussian kernel

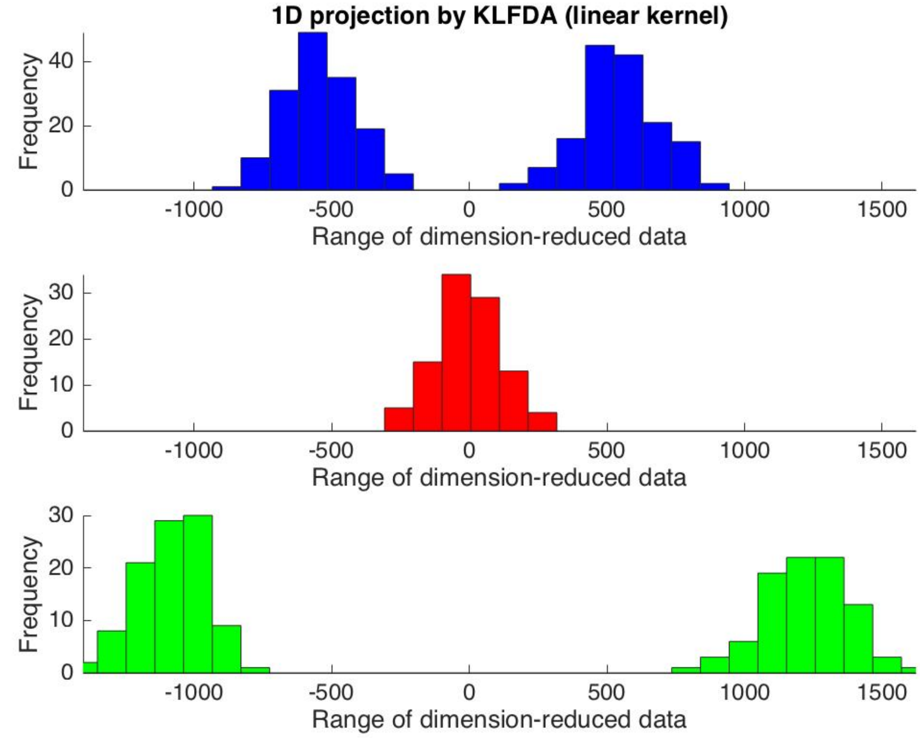

FIGURE 4.3: 1D distribution of dimension-reduced data with linear kernel

## 4.2 Mahalanobis distance

Mahalanobis distance [35] based metric learning has received much attention in similarity computing. The Mahalanobis distance of two observations $\boldsymbol{x}$ and $\boldsymbol{y}$ is defined as

$$D(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{x} - \boldsymbol{y})^T \boldsymbol{M} (\boldsymbol{x} - \boldsymbol{y}), \tag{4.32}$$

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

$$D(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{x} - \boldsymbol{y})^T \boldsymbol{W}^T \boldsymbol{W} (\boldsymbol{x} - \boldsymbol{y}) = ||\boldsymbol{W}(\boldsymbol{x} - \boldsymbol{y})||. \tag{4.33}$$

Therefore, Mahalanobis distance can be regarded as a variant of Euclidean distance.

## 4.3 Gradient descent optimization

Given a multivariate function $F(\boldsymbol{x})$, where $\boldsymbol{x}$ is a $d$-dimensional vector, if $F(\boldsymbol{x})$ is continuous and differentiable in the neighbour of point $\boldsymbol{x}$ for all $\boldsymbol{x}$, then $F(\boldsymbol{x})$ decreases fastest toward the direction of the negative gradient of $F(\boldsymbol{x})$. To compute the minimum of $F(\boldsymbol{x})$, an iterative method can be used by updating $F$ with respect to $\boldsymbol{x}$. If the updating step $\lambda$ is small enough, by updating $\boldsymbol{x}$ with

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \lambda \boldsymbol{G}_t, \tag{4.34}$$

where

$$\boldsymbol{G}_t = \frac{\partial F(\boldsymbol{x})}{\partial \boldsymbol{x}_t}, \tag{4.35}$$

we have

$$F(\boldsymbol{x}_t) \geq F(\boldsymbol{x}_t + 1). \tag{4.36}$$

$\boldsymbol{x}_t$ is updated by the gradient $\boldsymbol{G}$, this iterative process is repeated until a certain condition is met, generally when gradient $||\boldsymbol{G}|| \leq \eta$, and $\eta$ is a very small positive integer.

**Analysis of steepest gradient descent method** The advantages of gradient descent are that it is always downhill and it can avoid the saddle points. Besides, it is very efficient when the initial value of $F(\boldsymbol{x})$ is far from the minimum. However,

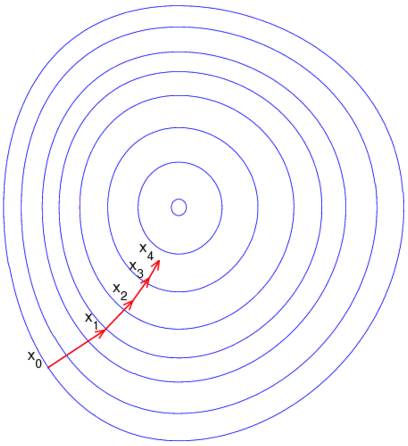

FIGURE 4.4: Steepest gradient descent

there are a few shortcomings of the gradient descent method. The first one is that the convergence value of gradient descent might be the local minima of $F(\boldsymbol{x})$ if $F(\boldsymbol{x})$ is not monotonic. In this case, the convergence value will depend on the initial value of $\boldsymbol{x}$. Another shortcoming is that the converging speed goes very slowly when approaching the minimum. One example is the case of zigzag downhill. The third shortcoming is linear search in gradient descent might cause some problems.

## 4.4 Metric learning based on sample pairs distance comparison

Inspired by the top-push video-based distance metric learning in [47], a similar metric learning based on iteration computation is used in this thesis. 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 $y_i \neq y_j$. Similarly, the method in [51] is also based on similarity comparison. In [51], 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 all possible positive and negative pairs, its computation complexity is quite expensive. To reduce the complexity, a simplified version is proposed as the top-push distance metric learning [47]. Since re-identification is a problem of ranking, it is desired that the rank-1 descriptor should be the right match. Given a Mahalanobis matrix $\boldsymbol{M}$, for samples $\boldsymbol{x}_i, i = 1, 2, 3, \cdots, n$, where $n$ is the number of all the samples, the requirement is that the maximum distance among all positive pairs should be at least one unit smaller than the minimum distance of all negative pairs. This can be denoted as

$$D(\boldsymbol{x}_i, \boldsymbol{x}_j) + \rho < \min D(\boldsymbol{x}_i, \boldsymbol{x}_k), y_i = y_j, y_i \neq y_k, \tag{4.37}$$

where $\rho$ is a slack variable and $\rho \in [0, 1]$. This equation can be transformed into an optimization problem as

$$\min \sum_{y_i = y_j} \max\{D(\boldsymbol{x}_i, \boldsymbol{x}_j) - \min_{y_i \neq y_k} D(\boldsymbol{x}_i, \boldsymbol{x}_k) + \rho, 0\}. \tag{4.38}$$

However, the equation above only penalizes the minimum interclass distance. Another term is needed to penalize large intraclass distance. That is, the sum of intraclass distance should be as small as possible. This term is denoted as

$$\min \sum_{y_i = y_j} D(\boldsymbol{x}_i, \boldsymbol{x}_j). \tag{4.39}$$

To combine the equations above, a ratio factor $\alpha$ is assigned to Equation (4.38) and (4.39) so that the target function can be denoted as

$$\begin{gathered} f(\boldsymbol{M}) = (1 - \alpha) \sum_{\boldsymbol{x}_i, \boldsymbol{x}_j, \boldsymbol{y}_i = y_j} D(\boldsymbol{x}_i, \boldsymbol{x}_j) + \\ \alpha \sum_{\boldsymbol{x}_i, \boldsymbol{x}_j, y_i = y_j} \max\{D(\boldsymbol{x}_i, \boldsymbol{x}_j) - \min_{y_i \neq y_k} D(\boldsymbol{x}_i, \boldsymbol{x}_k) + \rho, 0\}. \end{gathered} \tag{4.40}$$

In this way, the problem is transformed to an optimization problem. Notice that Equation 4.32 can be denoted as

$$D(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{x} - \boldsymbol{y})^T \boldsymbol{M} (\boldsymbol{x} - \boldsymbol{y}) = trace(\boldsymbol{M}\boldsymbol{X}_{i,j}), \tag{4.41}$$

where $\boldsymbol{X}_{i,j} = \boldsymbol{x}_i * \boldsymbol{x}_j^T$, and $trace$ is to compute the matrix trace. Therefore, Equation 4.40 can be transformed as follows:

$$\begin{gathered} f(\boldsymbol{M}) = (1 - \alpha) \sum_{y_i = y_j} trace(\boldsymbol{M}\boldsymbol{X}_{i,j}) \\ +\alpha \sum_{y_i = y_j, y_i \neq y_k} \max\{trace(\boldsymbol{M}\boldsymbol{X}_{i,j}) - trace(\boldsymbol{M}\boldsymbol{X}_{i,k}) + \rho, 0\}. \end{gathered} \tag{4.42}$$

To minimize Equation 4.42, the gradient descent method is used. The gradient respect to $\boldsymbol{M}$ is computed as

$$\boldsymbol{G} = \frac{\partial f}{\partial \boldsymbol{M}} = (1-\alpha)\sum_{y_i=y_j} \boldsymbol{X}_{i,j} + \alpha \sum_{y_i=y_j, y_i\neq y_k} (\boldsymbol{X}_{i,j} - \boldsymbol{X}_{i,k}). \tag{4.43}$$

The iteration process can be summarized in Table 4.1. In each iteration, to make sure the updated $\boldsymbol{M}$ is an SPD matrix, first a eigenvalue decomposition is performed on $\boldsymbol{M}$, and we have

$$\boldsymbol{M} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^T. \tag{4.44}$$

Here $\boldsymbol{\Lambda}$ is a diagonal matrix, and its diagonal elements are eigenvalues. Then the negative eigenvalues in $\boldsymbol{V}$ are removed, and the corresponding eigenvectors in $\boldsymbol{V}$ are also removed. Then $\boldsymbol{M}$ is restored by Equation (4.44).

TABLE 4.1: Optimization algorithm of Mahalanobis distance matrix learning

```
**Gradient optimization algorithm for target function**
**Input** Descriptors of training person pairs
**Output** An SPD matrix
**Initialization**
Initialize $\boldsymbol{M}_0$ 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 $t = t + 1$;
    Find $\boldsymbol{x}_k$ for all sample points $\boldsymbol{x}_i$, where $y_i \neq y_k$;
    Update gradient $\boldsymbol{G}_{t+1}$ with Equation 12;
    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-positive definite space;
    Update the target value $f|_{\boldsymbol{M}=\boldsymbol{M}_{t+1}}$;
**end while**
return $\boldsymbol{M}$
```

# Chapter 5

# Experiment settings

In this chapter, we evaluate the performance of our proposed metric and compare it with XQDA and NFST. We apply the proposed metric that is based on the KLFDA and gradient descent method to the GOG descriptor by a few steps. First, due to the high dimension of extracted GOG descriptors, KLFDA is applied to GOG descriptor to reduce the dimension to $C-1$, where $C$ is the number of person pairs in the training dataset. A transformation matrix $\boldsymbol{T}$ is generated in the dimension reduction. Second, with the dimension-reduced descriptors, a $(C-1)\times(C-1)$-dimensional SPD matrix $\boldsymbol{M}$ is trained based on the relative distance comparison, which restricts that the maximum intraclass distance is at least one unit smaller than the minimum interclass distance. The gradient descent method is used to optimize this problem by minimizing the target function denoted by the Equation 4.42 in Chapter 4. At last, the matrix $\boldsymbol{T}$ is used to reduce the dimension of testing data to $(C-1)$-dimensional vectors. With the trained SPD matrix $\boldsymbol{M}$, we compute the similarity scores and CMCs of the testing data.

The XQDA and NFST are two metrics used to compare with our proposed metric in this thesis. A brief introduction of these two metrics is given in Chapter 2. These two metrics are chosen because they outperform most of the other previous metrics. The detailed comparison can be found in [25] and [48]. Also, in this thesis, XQDA and NFST are tested with the GOG descriptor to reduce possible errors that may be caused by using different descriptors.

## 5.1 Datasets and evaluation settings

**VIPeR** dataset is the most widely 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 $48 \times 128$. In this experiment, we randomly select 316 individuals from cam A and cam B as the training set. The rest of the images in cam A are used as probe images, and those in cam B are used as gallery images. This process is repeated 10 times to reduce error.
**CUHK1** dataset contains 971 identities from two disjointed camera views. The cameras are static in each view, and images are listed in the same order. For each individual, there are two images in each view. All images are scaled into $60 \times 160$. In this paper, we randomly select 485 image pairs as training data, and the remaining pairs are used for test data.

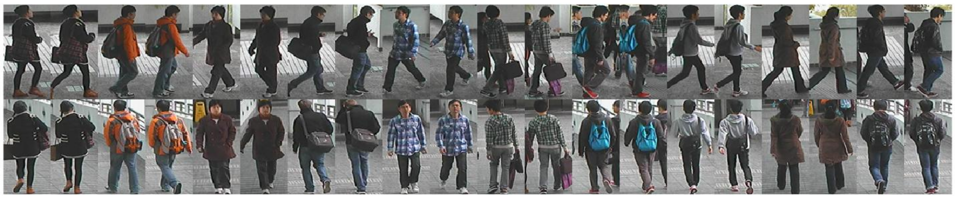

FIGURE 5.1: Pedestrians in prid_450 dataset

**Prid_2011** 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, and camera view B shows 749 persons. The first 200 persons appear in both camera views. The remaining persons in each camera view complete the set of the corresponding view. 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 are used as the probe set while the 649 remaining persons from camera B are used for the gallery set.
**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 the gallery and probe images.

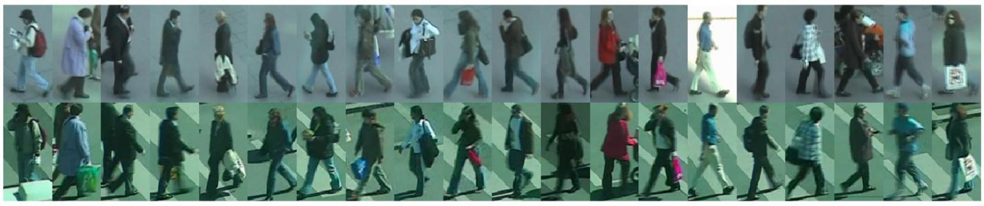

FIGURE 5.2: Pedestrians in prid_450s dataset

**GRID** has two camera views. The probe folder contains 250 probe images captured in one view (file names starts from 0001 to 0250). The gallery folder contains 250 true match images of the probes (file names starts from 0001 to 0250). Furthermore, in the 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 who appeared in both camera views as training pairs, and the remaining persons in the probe folder are used as probe images while the remaining 125 persons and those 775 additional persons from the gallery folder are used as gallery images. A brief summarization of test settings of all those datasets mentioned is listed in Table 5.1.

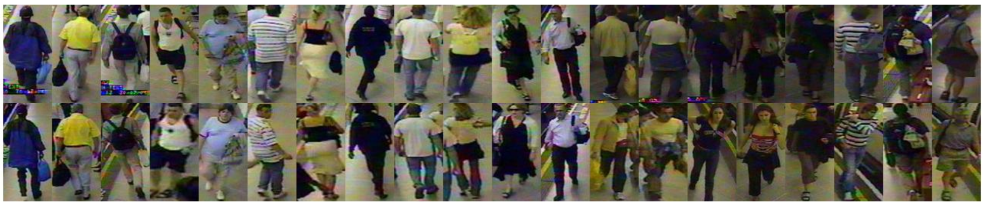

FIGURE 5.3: Pedestrians in GRID dataset

TABLE 5.1: 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 |

TABLE 5.2: Original image size of pedestrians from different datasets

| Datasets | VIPeR | CUHK1 | Prid_2011 | Prid_450s | GRID |
|---|---|---|---|---|---|
| Image size | 48*128 | 60*160 | 64*128 | variant | variant |

## 5.2 The influence of mean removal and $L_2$ normalization

In [25], mean removal and $L_2$ normalization is found to improve performance by $5.1\%$. The reason for this is that mean removal and normalization can reduce the impact of extremas of descriptors. When testing proposed metric learning, we find that mean removal can slightly improve performance. A comparison between performances of original descriptors and preprocessed descriptors is shown in Tables 5.3, 5.4, 5.5, 5.6 and 5.7, and 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 of around 0.5% on the performance on all five datasets. Since preprocessing is required to test XQDA, the mean removal and normalization are applied on descriptors in this experiment.

TABLE 5.3: The influence of data preprocessing on VIPeR

| | Re-ID matching accuracy at rank R(%) | | | | |
|---|---|---|---|---|---|
| Terms | R=1 | R=5 | R=10 | R=15 | R=20 |
| Original GOG | 43.01 | 74.91 | 84.87 | 89.81 | 93.32 |
| Preprocessed GOG$_{rgb}$ | 43.77 | 74.84 | 85.25 | 90.32 | 93.89 |
| Original GOG$_{fusion}$ | 48.77 | 77.47 | 87.41 | 91.52 | 94.27 |
| Preprocessed GOG$_{fusion}$ | 48.32 | 76.90 | 87.78 | 91.93 | 94.49 |

TABLE 5.4: The influence of data preprocessing on CUHK1

| | Re-ID matching accuracy at rank R(%) | | | | |
|---|---|---|---|---|---|
| Terms | R=1 | R=5 | R=10 | R=15 | R=20 |
| Original GOG$_{rgb}$ | 56.11 | 83.77 | 90.10 | 92.65 | 94.28 |
| Preprocessed GOG$_{rgb}$ | 55.91 | 84.24 | 90.41 | 93.15 | 94.67 |
| Original GOG$_{fusion}$ | 57.10 | 84.65 | 90.35 | 92.88 | 94.65 |
| Preprocessed GOG$_{fusion}$ | 56.67 | 84.49 | 90.51 | 93.31 | 94.84 |

TABLE 5.5: The influence of data preprocessing on prid_2011

| | Re-ID matching accuracy at rank R(%) | | | | |
|---|---|---|---|---|---|
| Terms | R=1 | R=5 | R=10 | R=15 | R=20 |
| Original GOG$_{rgb}$ | 24.80 | 52.10 | 63.20 | 69.90 | 72.90 |
| Preprocessed GOG$_{rgb}$ | 23.80 | 52.20 | 63.50 | 70.20 | 73.50 |
| Original GOG$_{fusion}$ | 32.20 | 56.60 | 67.00 | 73.10 | 77.70 |
| Preprocessed GOG$_{fusion}$ | 32.30 | 57.40 | 66.30 | 73.40 | 78.00 |

TABLE 5.6: The influence of data preprocessing on prid_450s

| | Re-ID matching accuracy at rank R(%) | | | | |
|---|---|---|---|---|---|
| Terms | R=1 | R=5 | R=10 | R=15 | R=20 |
| Original GOG$_{rgb}$ | 60.93 | 84.31 | 91.29 | 94.00 | 96.18 |
| Preprocessed GOG$_{rgb}$ | 60.71 | 84.53 | 91.29 | 94.13 | 96.27 |
| Original GOG$_{fusion}$ | 63.07 | 86.67 | 92.53 | 95.20 | 96.98 |
| Preprocessed GOG$_{fusion}$ | 62.80 | 86.58 | 92.36 | 95.29 | 96.89 |

TABLE 5.7: The influence of data preprocessing on GRID

| | Re-ID matching accuracy at rank R(%) | | | | |
|---|---|---|---|---|---|
| Terms | R=1 | R=5 | R=10 | R=15 | R=20 |
| Original $\text{GOG}_{\text{rgb}}$ | 22.96 | 41.92 | 51.68 | 58.72 | 64.64 |
| Preprocessed $\text{GOG}_{\text{rgb}}$ | 22.64 | 43.68 | 52.00 | 59.04 | 65.04 |
| Original $\text{GOG}_{\text{fusion}}$ | 24.32 | 44.56 | 54.80 | 62.40 | 66.64 |
| Preprocessed $\text{GOG}_{\text{fusion}}$ | 23.92 | 44.64 | 54.88 | 62.32 | 66.40 |

## 5.3 Parameters setting

In this experiment, there are a few parameters for the iteration computing, including slack variable $\rho$, maximal iteration $T$, gradient step $\lambda$, the interclass and intraclass limitation factor $\alpha$ and the updating ratio $\beta$. First, the slack variable $\rho$ is initialized as 1 to ensure the minimum interclass distance is 1 larger than intraclass distance at least. The step size of gradient updating $\lambda$ is initialized as 0.01. When target value $f$ increases, $\lambda$ is scaled by a factor of 0.5, and $\lambda$ is scaled by 1.01 when target value $f$ decreases. To judge if target value converges, the threshold $\beta$ is defined as the ratio target value change versus previous target value, that is, $\beta = \frac{(f_{t+1} - f_t)}{f_t}$. According to many experiment trials, when it satisfies $\beta = 10^{-5}$, the target value converges and the iteration is stopped. The maximal iteration times $t$ is set to 100 since the target value $f$ will converge in around 15 iterations. The last parameter for the iteration is $\alpha$. To know the best value for $\alpha$, we tried 11 different values ranges from 0 to 1 with a step of 0.1. We found that the rank 1 and rank 5 scores reach maxima at interval $[0.7, 0.8]$, as shown in 5.4 and 5.5. Then another ten trials were performed with alpha ranging from $[0.7, 0.8]$ with a step of 0.01. The best $\alpha$ value should have largest top-ranking scores possible. According to Figure 5.4 and 5.5, $\alpha$ is set to be 0.76. A form of all parameters is shown in Table 5.8.

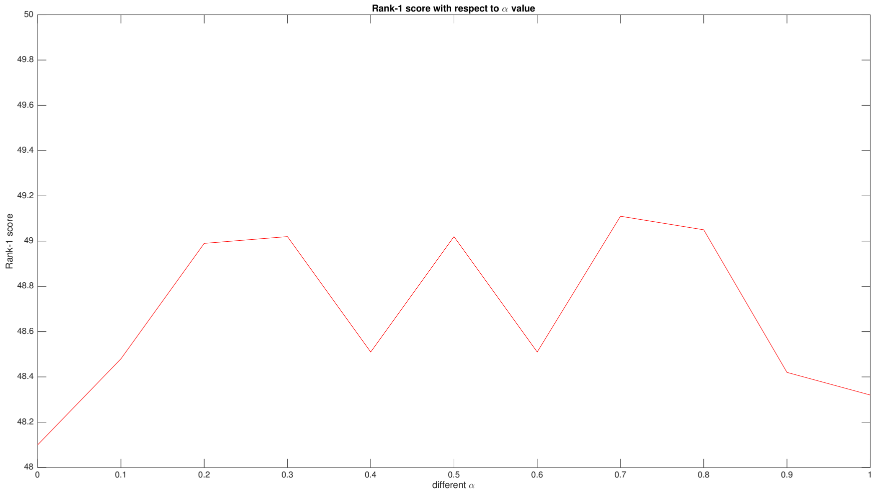

FIGURE 5.4: Rank 1 scores with respect to $\alpha$ on VIPeR

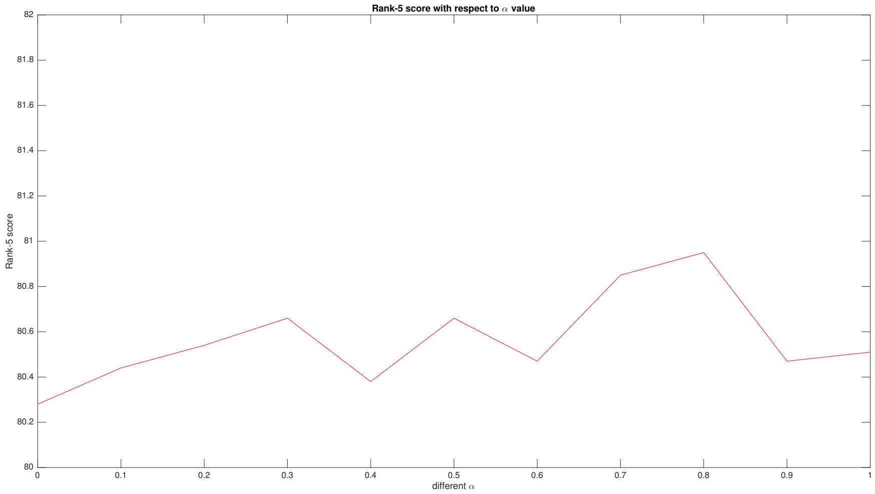

FIGURE 5.5: Rank 5 scores with respect to $\alpha$ on VIPeR

TABLE 5.8: Parameters setting

| Paramters | $\alpha$ | $\beta$ | $\lambda$ | $t$ | $\rho$ |
|---|---|---|---|---|---|
| Values | 0.76 | $10^{-5}$ | 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 better CMC is expected to have a high rank-1 value and reaches 1 as fast as possible.

## 5.4 Results and experiment analysis

In this paper, we compare proposed metric with other state-of-the-art metrics including NFST [48] and XQDA [22]. NFST is a metric which enables us to 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 to many other metrics, which learns a projection matrix $W$, and then a Mahalanobis SPD matrix $M$ is learned in the subspace. These two metrics are proven to have state-of-the-art performance with many other methods. The $\text{GOG}_{\text{rgb}}$ in all forms stands for the hierarchical Gaussian descriptor in RGB color space, while $\text{GOG}_{\text{fusion}}$ stands for the descriptor concatenated by four different color spaces {RGB, Lab, HSV, nRnG}.

**VIPeR** A comparison form is given in Table 3. 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 $\text{GOG}_{\text{rgb}}$ and $\text{GOG}_{\text{fusion}}$. More specifically, the rank 1, rank 5, rank 10, rank 15 and rank 20 scores of proposed metric learning are 0.76%, 0.92%, 1.39%, 1.08%, 1.52% higher than those of $\text{GOG}_{\text{rgb}}$ + XQDA. The rank 1, rank 5, rank 10, rank 15 and rank 20 $\text{GOG}_{\text{fusion}}$ scores of proposed metric learning are 0.35%, -0.54%, 0.98%, 0.66%, 0.79% higher than $\text{GOG}_{\text{fusion}}$ + XQDA respectively. Also, we can see that proposed metric learning has a better performance than NFST.

TABLE 5.9: Performance of different metrics on VIPeR

| | Re-ID matching accuracy at rank R(%) | | | | |
|---|---|---|---|---|---|
| Methods | R=1 | R=5 | R=10 | R=15 | R=20 |
| $\text{GOG}_{\text{rgb}}$+NFST | 43.23 | 73.16 | 83.64 | 89.59 | 92.88 |
| $\text{GOG}_{\text{rgb}}$+XQDA | 43.01 | 73.92 | 83.86 | 89.24 | 92.37 |
| $\text{GOG}_{\text{rgb}}$+Proposed | 43.77 | 74.84 | 85.25 | 90.32 | 93.89 |
| $\text{GOG}_{\text{fusion}}$+NFST | 47.15 | 76.39 | 87.31 | 91.74 | 94.49 |
| $\text{GOG}_{\text{fusion}}$+XQDA | 47.97 | 77.44 | 86.80 | 91.27 | 93.70 |
| $\text{GOG}_{\text{fusion}}$+Proposed | 48.32 | 76.90 | 87.78 | 91.93 | 94.49 |

**CUHK1** We can find that the rank 1, rank 5, rank 10, rank 15 and rank 20 scores

of GOG$_{\text{rgb}}$ combined with proposed metric are 5.4%, 4.18%, 3.31%, 2.16% and 1.46% higher than XQDA, and 0.31%, 1.22%, 1.34%, 1.17% and 1.11% higher than NFST. Also, the rank 1, rank 5, rank 10, rank 15 and rank 20 scores of GOG$_{\text{fusion}}$ combined with the proposed metric are 4.57%, 2.64%, 0.70%, 1.33% and 0.83% higher than GOG$_{\text{fusion}}$ combined with XQDA, and 0.41%, 0.83%, 0.88%, 1.09% and 1.14% than GOG$_{\text{fusion}}$ combined with NFST.

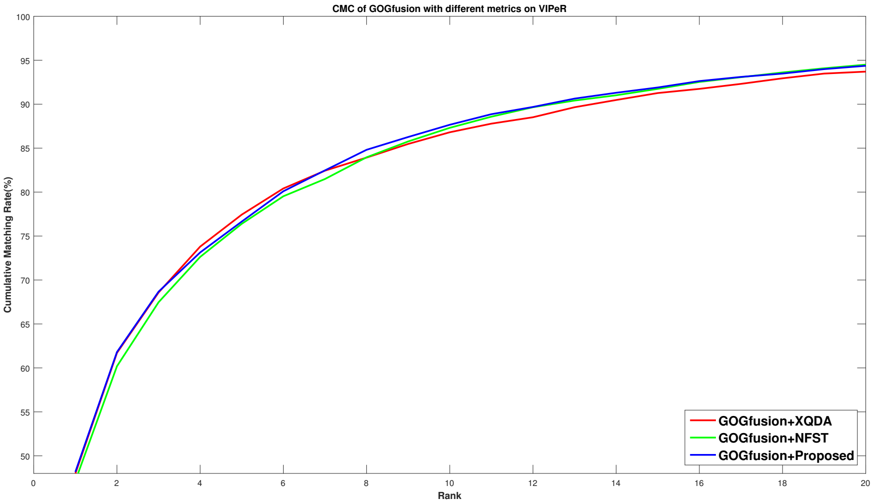

FIGURE 5.6: CMCs on VIPeR comparing different metric learning

TABLE 5.10: Performance of different metrics on CUHK1

| | Re-ID matching accuracy at rank R(%) | | | | |
|---|---|---|---|---|---|
| Methods | 1 | 5 | 10 | 15 | 20 |
| GOG$_{\text{rgb}}$+NFST | 55.60 | 83.02 | 89.07 | 91.98 | 93.56 |
| GOG$_{\text{rgb}}$+XQDA | 50.51 | 80.06 | 87.10 | 90.99 | 93.21 |
| GOG$_{\text{rgb}}$+Proposed | 55.91 | 84.24 | 90.41 | 93.15 | 94.67 |
| GOG$_{\text{fusion}}$+NFST | 56.26 | 83.66 | 89.63 | 92.22 | 93.70 |
| GOG$_{\text{fusion}}$+XQDA | 52.10 | 81.85 | 88.81 | 91.98 | 94.01 |
| GOG$_{\text{fusion}}$+Proposed | 56.67 | 84.49 | 90.51 | 93.31 | 94.84 |

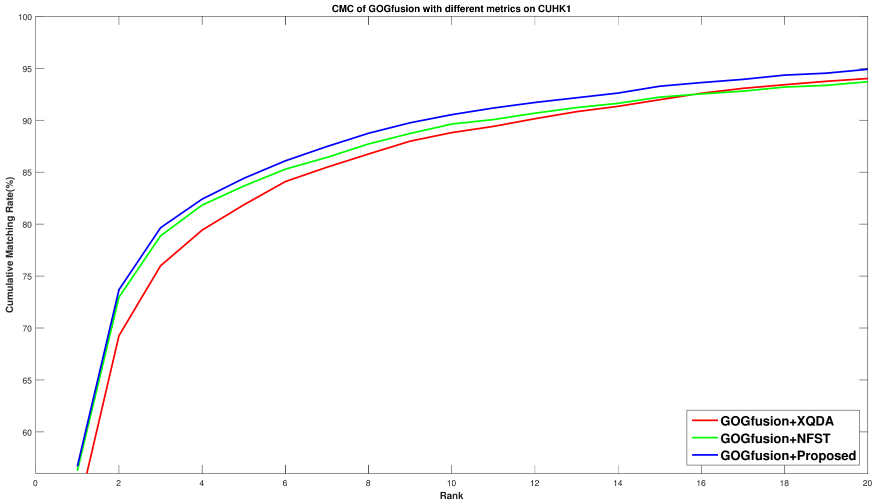

FIGURE 5.7: CMCs on CUHK1 comparing different metric learning

TABLE 5.11: Performance of different metrics on prid_2011

| | Re-ID matching accuracy at rank R(%) | | | | |
|---|---|---|---|---|---|
| Methods | R=1 | R=5 | R=10 | R=15 | R=20 |
| GOG$_{\text{rgb}}$+NFST | 26.60 | 53.80 | 62.90 | 71.30 | 75.40 |
| GOG$_{\text{rgb}}$+XQDA | 31.10 | 55.70 | 66.10 | 72.40 | 76.10 |
| GOG$_{\text{rgb}}$+Proposed | 23.80 | 52.20 | 63.50 | 70.20 | 73.50 |
| GOG$_{\text{fusion}}$+NFST | 34.10 | 58.30 | 67.60 | 73.80 | 78.30 |
| GOG$_{\text{fusion}}$+XQDA | 38.40 | 61.30 | 70.80 | 75.60 | 79.30 |
| GOG$_{\text{fusion}}$+Proposed | 32.30 | 57.40 | 66.30 | 73.40 | 78.00 |

**Prid_2011** The rank 1, rank 5, rank 10, rank 15 and rank 20 scores of GOG$_{\text{fusion}}$ combined with proposed metric are 6.1%, 3.9%, 4.5%, 2.2% and 1.3% lower than GOG$_{\text{fusion}}$ combined with XQDA. The performance of NFST is slightly better than proposed metric. Also, in terms of GOG$_{\text{rgb}}$, XQDA and NFST have better performance than the proposed one. So in this dataset, the proposed metric has worse performance than XQDA and NFST.

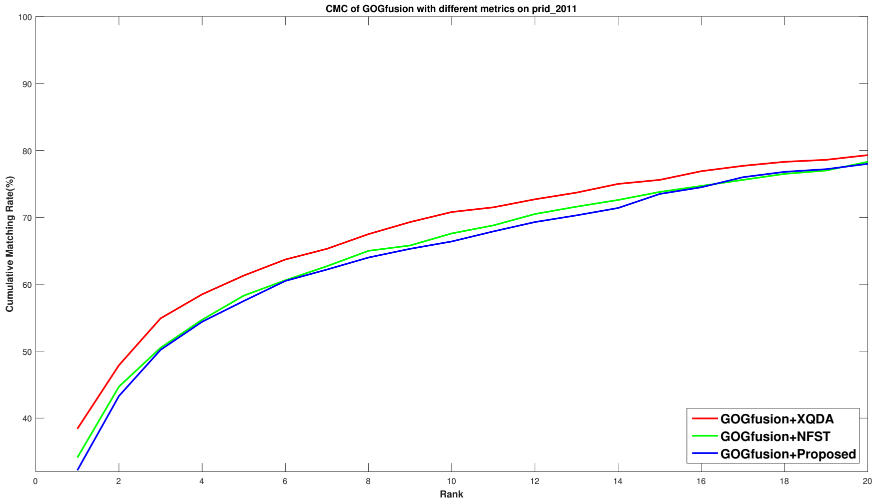

FIGURE 5.8: CMCs on prid_2011 comparing different metric learning

TABLE 5.12: Performance of different metrics on prid_450s

| | Re-ID matching accuracy at rank R(%) | | | | |
|---|---|---|---|---|---|
| Methods | R=1 | R=5 | R=10 | R=15 | R=20 |
| GOG$_{\text{rgb}}$+NFST | 61.96 | 84.98 | 90.53 | 94.09 | 96.09 |
| GOG$_{\text{rgb}}$+XQDA | 65.29 | 85.02 | 91.13 | 94.76 | 96.49 |
| GOG$_{\text{rgb}}$+Proposed | 60.71 | 84.53 | 91.29 | 94.13 | 96.27 |
| GOG$_{\text{fusion}}$+NFST | 64.53 | 86.62 | 92.93 | 95.78 | 97.42 |
| GOG$_{\text{fusion}}$+XQDA | 68.40 | 87.42 | 93.47 | 95.69 | 97.02 |
| GOG$_{\text{fusion}}$+Proposed | 62.80 | 86.58 | 92.36 | 95.29 | 96.89 |

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

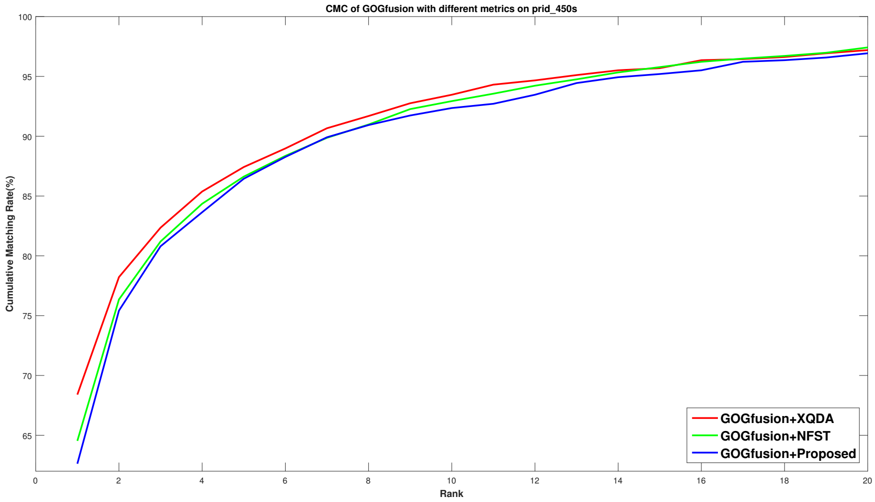

FIGURE 5.9: CMCs on prid_450s comparing different metric learning

TABLE 5.13: Performance of different metrics on GRID

| | Re-ID matching accuracy at rank R(%) | | | | |
|---|---|---|---|---|---|
| Methods | R=1 | R=5 | R=10 | R=15 | R=20 |
| GOG$_{\text{rgb}}$+NFST | 21.84 | 41.28 | 50.96 | 57.44 | 62.88 |
| GOG$_{\text{rgb}}$+XQDA | 22.64 | 43.92 | 55.12 | 61.12 | 66.56 |
| GOG$_{\text{rgb}}$+Proposed | 22.64 | 43.68 | 52.00 | 59.04 | 65.04 |
| GOG$_{\text{fusion}}$+NFST | 23.04 | 44.40 | 54.40 | 61.84 | 66.56 |
| GOG$_{\text{fusion}}$+XQDA | 23.68 | 47.28 | 58.40 | 65.84 | 69.68 |
| GOG$_{\text{fusion}}$+Proposed | 23.92 | 44.64 | 54.88 | 62.32 | 66.40 |

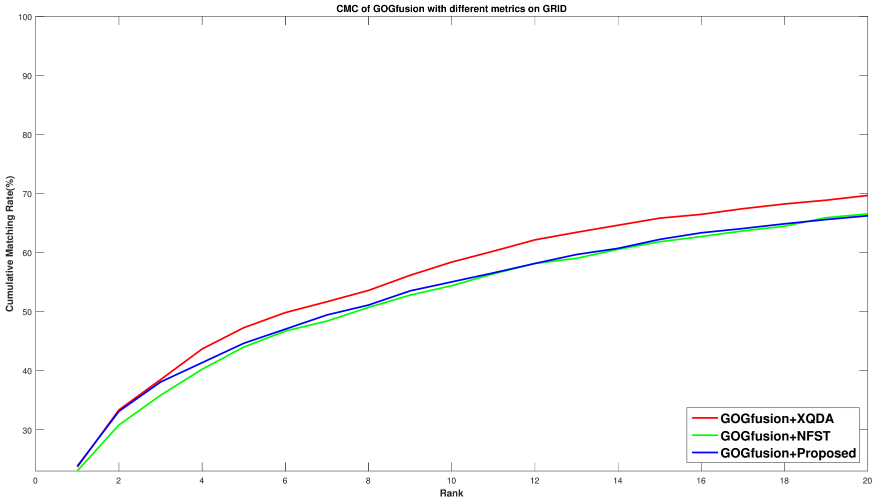

FIGURE 5.10: CMCs on GRID comparing different metric learning

**GRID** We can see that the rank 1 score of proposed metric is 0.24% higher than XQDA and 0.88% higher than NFST in terms of $\text{GOG}_{\text{fusion}}$, but XQDA outperforms proposed metric on rank 5, rank 10, rank 15 and rank 20 scores. Besides, proposed metric outperforms NFST on rank 5, rank 10 and rank 15 scores.

In summary, our proposed metric improved the Re-ID accuracy in VIPeR and CUHK1 datasets, and has almost the same performance with NFST and XQDA in the prid_450s dataset. Specifically, the proposed metric learning has the best rank 1 score in the GRID dataset and its performance is only second to XQDA. The proposed metric has superior performance for the following reasons: (1) dimension reduction by KLFDA exploits the nonlinearity, and the loss of discriminant information between classes is minimized; (2) the simplified relative distance limitation optimization helps to confine the Mahalanobis distance matrix $\boldsymbol{M}$ to discriminate different classes.

## 5.5 Comparison with other methods

As mentioned in Chapter 2.4, in [25], GOG + XQDA outperforms many other combinations, including Metric ensemble [32], SCNCD [46], Semantic method [37], etc. In [48], LOMO + NFST outperforms metrics including LMNN [41], KCCA [42], ITML [13], KLFDA [38], MFA [44], KISSME [20], Similarity learning [9], SCNCD [46], Mid-level filters [50] and Improved deep learning [14]. Based on the result that XQDA and NFST outperform other metrics, and the proposed method in this thesis has better or the same performance with XQDA and NFST. It can be concluded that the proposed method outperforms those other methods in [25] and [48].

## 5.6 The Image size and time cost

The image size of different datasets are different. A brief introduction of image size of those datasets used is given in Table 5.2. All those images have already been well cropped which only contain the area inside the person bounding box. Before extracting descriptors, all those images must be resized to 48*128.

The training time increases when the dataset size increases. For the CUHK1 dataset which has most training pairs, it takes half an hour to train the metric on a desktop PC with a 16GB RAM, Intel i5 processor. For datasets with fewer training pairs like Prid_2011 and GRID, it takes no more than five minutes.

With resized images, the Re-ID can be performed in real-time on the same PC. Compared with XQDA and NFST, this method requires extra time to train the metric. However, after training the Re-ID can be performed instantaneously upon the new pedestrian is inputted. The accuracy of proposed method is at least the same or better than XQDA and NFST on tested datasets.

# Chapter 6

# Conclusion

In this thesis, first some variants of hierarchical Gaussian descriptor has been tested. Three kinds variants are proposed by changing the basic pixel feature vector, changing the way to construct small patches by using superpixel segmentation and replacing the single Gaussian model with Gaussian mixture model. However, those three variants have worse performance than the original version of hierarchical Gaussian descriptor. Therefore, the original version is used in this thesis. The original version of hierarchical Gaussian descriptor characterizes the pixel by the pixel's vertical coordinate, texture and color. Then a hierarchical structure is created by modelling from pixels to patches and from patches to regions. The final dimension of this descriptor is $1 * 27622$.

Furthermore, KLFDA was used to reduce the dimension of the hierarchical Gaussian descriptors, and gradient descent method was used to learn a Mahalanobis distance metric on the lower-dimensional space. By comparison, we found that the proposed metric has better performance than NFST and XQDA on the VIPeR and CUHK1 datasets, but XQDA and NFST outperformed the proposed metric learning on the Prid_2011 and Prid_450s. We also found that the proposed metric learning had better rank 1 score than NFST, and its performance was only second to XQDA on the GRID dataset.

## 6.1 Contributions

There are three contributions in this thesis. (1) The metric learning on the dimension-reduced hierarchical Gaussian descriptor by KLFDA is studied. KLFDA is used to reduce dimension in this thesis, and the gradient descent method optimizes a Mahalanobis distance metric on the lower-dimensional space. In this thesis it has been demonstrated that extra improvement can be achieved by metric learning in this lower-dimensional space. (2) By experiment it is concluded that imperfect foreground segmentation improves performance of histogram-based descriptors but decreases performance of texture-based descriptors, which is caused by the imperfect segmentation on single-image-based foreground segmentation. (3) Some variants of the hierarchical Gaussian descriptor have been tested. LBP and superpixel segmentation were combined with the hierarchical Gaussian descriptor, but those variants had worse performance than the original hierarchical Gaussian descriptor.

## 6.2 Future work

### 6.2.1 Improve the hierarchical Gaussian descriptor

It has been demonstrated that in the basic pixel feature $\boldsymbol{f}_i$, the color components are more important than the $y$ coordinate and gradient components. LBP has been proven to be a worse choice for texture representation in $\boldsymbol{f}_i$. Therefore, one strategy to improve the hierarchical Gaussian descriptor is to find a better texture representation to replace gradient components in basic pixel feature.

### 6.2.2 Improve foreground segmentation algorithm

The imperfect foreground segmentation decreases the performance of textural features. One solution is to improve the method of modelling the foreground of a single image. Another solution is to segment the foreground based on a sequence of images or a video. Video-based foreground segmentation can get better results. If the foreground can be well modelled by a video or a sequence of images so that less textural noise is created, the segmentation might improve the hierarchical Gaussian descriptor's performance.

### 6.2.3 Decrease the computational cost of gradient descent method

It is important to reduce the computational cost of the gradient descent method. It takes an average of about average 15 iterations when training the Mahalanobis distance matrix. The training time for a larger dataset, like the CUHK1 dataset, takes up to one hour on a computer with a 16GB RAM, Intel i5 processor. Therefore, other variants of gradient the descent method, like stochastic gradient method and conjugate gradient method, may be tested for lower computational cost.

# Bibliography

[1] Radhakrishna Achanta et al. “SLIC superpixels compared to state-of-the-art superpixel methods”. In: *IEEE transactions on pattern analysis and machine intelligence* 34.11 (2012), pp. 2274–2282.

[2] Mykhaylo Andriluka, Stefan Roth, and Bernt Schiele. “Pictorial Structures Revisited: People Detection and Articulated Pose Estimation”. In: *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE. Apr. 2009, pp. 1–8.

[3] Davide Baltieri, Roberto Vezzani, and Rita Cucchiara. “SARC3D: a new 3D body model for People Tracking and Re-identification”. In: *International Conference on Image Analysis and Processing*. Springer. Mar. 2011, pp. 1–10.

[4] Oren Barkan et al. “Fast high dimensional vector multiplication face recognition”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2013, pp. 1960–1967.

[5] Loris Bazzani et al. “Multiple-shot person re-identification by chromatic and epitomic analyses”. In: *Pattern Recognition Letters* 33.7 (2012), pp. 898–903.

[6] Bedagkar-Gala et al. “A Survey of Approaches and Trends in Person Re-identification”. In: *Image and Vision Computing* 32.4 (Mar. 2014), pp. 270–286.

[7] A Bedagkar-Gala and Shishir K Shah. “Part-based spatio-temporal model for multi-person re-identification”. In: *Pattern Recognition Letters* 33.14 (Oct. 2012), pp. 1908–1915.

[8] Apurva Bedagkar-Gala and Shishir K Shah. “Multiple Person Re-identification using Part based Spatio-Temporal Color Appearance Model”. In: *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*. IEEE. Oct. 2011, pp. 1–8.

[9] Dapeng Chen et al. “Similarity Learning on an Explicit Polynomial Kernel Feature Map for Person Re-Identification”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Apr. 2015, pp. 1–9.

[10] De Cheng et al. “Person Re-Identification by Multi-Channel Parts-Based CNN With Improved Triplet Loss Function”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. July 2016, pp. 1–10.

[11] Shaogang Gong Chen Change Loy Chunxiao Liu and Xinggang Lin. “Person Re-identification: What Features Are Important?” In: *European Conference on Computer Vision*. IEEE. Nov. 2015, pp. 1–11.

[12] Navneet Dalal and Bill Triggs. “Histograms of Oriented Gradients for Human Detection”. In: *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. Jan. 2005, pp. 886–893.

[13] Jason V Davis et al. “Information-Theoretic Metric Learning”. In: *Proceedings of the 24th international conference on Machine learning*. ACM. June 2007, pp. 209–216.

[14] Michael Jones Ejaz Ahmed and Tim K Marks. “An Improved Deep Learning Architecture for Person Re-Identification”. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 1–9.

[15] M. Farenzena et al. “Person Re-Identification by Symmetry-Driven Accumulation of Local Features”. In: *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE. Mar. 2016, pp. 1–8.

[16] Pedro F Felzenszwalb and Daniel P Huttenlocher. “Pictorial Structures for Object Recognition”. In: *International Journal of Computer Vision* 61.1 (Jan. 2005), pp. 1–42.

[17] Xiaofei He and Partha Niyogi. “Locality Preserving Projections”. In: *NIPS*. Nov. 2003, pp. 1–8.

[18] Nebojsa Jojic et al. “Stel component analysis: Modeling spatial correlations in image class structure”. In: *Computer Vision and Pattern Recognition*. Apr. 2009, pp. 1–8.

[19] Arif Khan, Jian Zhang, and Yang Wang. “Appearance-Based Re-identification of People in Video”. In: *2010 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*. IEEE, Nov. 2010, pp. 357–362.

[20] Martin Kostinger et al. “Large Scale Metric Learning from Equivalence Constraints”. In: *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE. Apr. 2012, pp. 2288–2295.

[21] Peihua Li, Qilong Wang, and Lei Zhang. “A Novel Earth Mover’s Distance Methodology for Image Matching with Gaussian Mixture Models”. In: *Proceedings of the IEEE International Conference on Computer Vision*. IEEE, Dec. 2013, pp. 1689–1696.

[22] Shengcai Liao et al. “Person Re-identification by Local Maximal Occurrence Representation and Metric Learning”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Apr. 2015, pp. 1–10.

[23] Miroslav Lovric, Maung Min-Oo, and Ernst A Ruh. “Multivariate Normal Distributions Parametrized as a Riemannian Symmetric Space”. In: *Journal of Multivariate Analysis* 74.1 (June 2000), pp. 1–15.

[24] David G Lowe. “Object Recognition from Local Scale-Invariant Features”. In: *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*. IEEE. June 1999, pp. 1–8.

[25] Tetsu Matsukawa et al. “Hierarchical Gaussian Descriptor for Person Re-Identification”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Dec. 2016, pp. 1363–1372.

[26] Niall McLaughlin, Jesus Martinez del Rincon, and Paul Miller. “Recurrent Convolutional Network for Video-Based Person Re-Identification”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. July 2016, pp. 1–10.

[27] Alexis Mignon and Frederic Jurie. “PCCA: A New Approach for Distance Learning from Sparse Pairwise Constraints”. In: *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE. Apr. 2012, pp. 1–7.

[28] Baback Moghaddam, Tony Jebara, and Alex Pentland. “Bayesian Face Recognition”. In: *Pattern Recognition* 33.11 (Jan. 2000), pp. 1771–1782.

[29] T. Ojala, M. Pietikäinen, and D. Harwood. “A Comparative Study of Texture Measures with Classification Based on Feature Distributions”. In: *Pattern recognition* (1996), pp. 51–59.

[30] T. Ojala, M. Pietikäinen, and D. Harwood. “Performance evaluation of texture measures with classification based on Kullback discrimination of distributions”. In: *ICPR*. IEEE. 1994, pp. 582–585.

[31] Takumi Kobayashi Otsu and Nobuyuki. “Image Feature Extraction Using Gradient Local Auto-Correlations”. In: *European conference on computer vision*. Aug. 2008, pp. 1–13.

[32] Sakrapee Paisitkriangkrai, Chunhua Shen, and Anton van den Hengel. “Learning to Rank in Person Re-Identification With Metric Ensembles”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Jan. 2015, pp. 1–10.

[33] Sateesh Pedagadi et al. “Local Fisher Discriminant Analysis for Pedestrian Re-identification”. In: *2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Jan. 2013, pp. 3318–3325.

[34] Bryan Prosser et al. “Person Re-Identification by Support Vector Ranking”. In: *British Machine Vision Conference 2010*. British Machine Vision Association, July 2010, pp. 21.1–21.11.

[35] Peter M Roth et al. “Mahalanobis Distance Learning for Person Re-Identification”. In: *Person Re-Identification*. Springer, July 2014, pp. 1–21.

[36] Riccardo Satta. “Appearance Descriptors for Person Re-identification: a Comprehensive Review”. In: *Computing Research Repository* (July 2013), pp. 1–18.

[37] Zhiyuan Shi, Timothy M Hospedales, and Tao Xiang. “Transferring a Semantic Representation for Person Re-Identification and Search”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Apr. 2015, pp. 1–10.

[38] Masashi Sugiyama. “Local Fisher Discriminant Analysis for Supervised Dimensionality Reduction”. In: *Proceedings of the 23rd international conference on Machine learning*. ACM. Dec. 2006.

[39] Jirí Trefnỳ and Jirí Matas. “Extended set of local binary patterns for rapid object detection”. In: *Computer Vision Winter Workshop*. 2010, pp. 1–7.

[40] Oncel Tuzel, Fatih Porikli, and Peter Meer. “Region Covariance: A Fast Descriptor for Detection and Classification”. In: *European conference on computer vision*. Springer. Dec. 2016, pp. 1–14.

[41] Weinberger et al. “Distance Metric Learning for Large Margin Nearest Neighbor Classification”. In: *Journal of Machine Learning Research* 10 (Feb. 2009), pp. 207–244.

[42] Max Welling. “Kernel Canonical Correlation Analysis”. In: (Mar. 2005), pp. 1–3.

[43] Tong Xiao et al. “Learning Deep Feature Representations with Domain Guided Dropout for Person Re-identification ”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Apr. 2016, pp. 1–10.

[44] Fei Xiong et al. “Person Re-Identification using Kernel-based Metric Learning Methods”. In: *European conference on computer vision*. Springer. July 2014, pp. 1–16.

[45] Liu Yang and Rong Jin. “Distance metric learning: A comprehensive survey”. In: *Michigan State Universiy* 2.2 (2006).

[46] Yang Yang et al. “Salient color names for person re-identification”. In: *European Conference on Computer Vision*. 2014.

[47] Jinjie You et al. “Top-push Video-based Person Re-identification”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Jan. 2016, pp. 1–9.

[48] Li Zhang, Tao Xiang, and Shaogang Gong. “Learning a Discriminative Null Space for Person Re-identification”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Mar. 2016, pp. 1239–1248.

[49] Guoying Zhao and Matti Pietikainen. “Dynamic texture recognition using local binary patterns with an application to facial expressions”. In: *IEEE transactions on pattern analysis and machine intelligence* 29.6 (2007).

[50] Rui Zhao, Wanli Ouyang, and Xiaogang Wang. “Learning Mid-level Filters for Person Re-identification”. In: *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, May 2014, pp. 144–151.

[51] WeiShi Zheng, Shaogang Gong, and Tao Xiang. “Person Re-identification by Probabilistic Relative Distance Comparison”. In: *Computer vision and pattern recognition (CVPR), 2011 IEEE conference on*. IEEE. Nov. 2016, pp. 1–8.