Non-linear feature extraction for object re-identification in cameras networks

Charbel Chahla

Thèse
de doctorat
de l'UTT

Charbel CHAHLA

Non-linear Feature Extraction for Object Re-identification in Cameras Networks

Spécialité :
Optimisation et Sûreté des Systèmes

2017TROY0023 **Année 2017**

Thèse en cotutelle avec l'Université Libanaise - Beyrouth - Liban

Université Libanaise

utt
UNIVERSITÉ DE TECHNOLOGIE
TROYES

THESE

pour l'obtention du grade de

DOCTEUR de l'UNIVERSITE DE TECHNOLOGIE DE TROYES

Spécialité : OPTIMISATION ET SURETE DES SYSTEMES

présentée et soutenue par

Charbel CHAHLA

le 28 septembre 2017

Non-linear Feature Extraction for Object Re-identification in Cameras Networks

JURY

M. C. FRANCIS	PROFESSEUR	Président
M. F. ABDALLAH	PROFESSEUR	Directeur de thèse
M. F. DORNAIKA	PROFESSOR	Directeur de thèse
M. H. DRIRA	MAITRE DE CONFERENCES	Examinateur
M. M. A. EL YACOUBI	PROFESSEUR - HDR TELECOM SUDPARIS	Rapporteur
Mme L. OUKHELLOU	DIRECTEUR DE RECHERCHE	Rapporteur
M. H. SNOUSSI	PROFESSEUR DES UNIVERSITES	Directeur de thèse

Acknowledgments

First and foremost, I would like to gratefully and sincerely thank my advisors, Hichem SNOUSSI, Fahed ABDALLAH and Fadi DORNAIKA for their excellent guidance, patience, and most importantly, their friendship during these last three years. Without them, this thesis would not have been successfully completed. One simply could not wish for better or more devoted advisors.

Next, I would like to thank the members of my defense committee for their time and dedication. I thank the reviewers, Latifa OUKHELLOU and Mounim EL YACOUBI, who patiently read the manuscript and provided detailed comments and insightful suggestions. I also thank the examiners Clovis FRANCIS and Hassen DRIRA. Their helpful feedback greatly improved the contents of this manuscript.

I also would like to thank all the members of the LM2S Laboratory for their support and friendship. My gratitude extends to Bernadette André and Véronique Banse for their availability and all the help they provided me with. I also thank the UTT doctoral school, in particular the director Régis Lengellé, and the secretaries Pascale Denis, Isabelle Leclercq and Thérèse Kazarian.

My thanks go to all my colleagues and all the friends I made during these last three years. Words cannot express my infinite love and gratitude for all of you. Thank you for your patience and encouragement throughout this thesis!

I greatly thank my mother Mona and my father Maroun for their unfailing support and their faith in me. I am indebted to them for teaching me dedication and discipline to do whatever I undertake well. I also thank my sister Cherine for always managing to put a smile on my face.

Abstract

Replicating the human visual system that the brain uses to process the information is an area of substantial scientific interest. This field of research, known as Computer Vision, is notoriously difficult. Almost no Computer Vision problem has been satisfactorily solved. The main reason for this difficulty is that the human visual system is simply too good, and Computer Vision systems suffer by comparison. A fundamental requirement of these systems is to detect, track and re-identify a person or an object in a region of interest.

This thesis is situated in the context of a fully automated system capable of analyzing facial features when the target is near the cameras, and tracking his identity when his facial features are no more traceable. Out of plane rotation of face has long been one of the bottlenecks in the face recognition area, in the sense that these systems are very sensitive to pose variations. The first part of this thesis is devoted to face pose estimation procedures to be used in face recognition scenarios. We proposed a new label-sensitive embedding based on a sparse representation. The resulting technique is called Sparse Label sensitive Locality Preserving Projections. For enhancing the discrimination between poses, the projected data obtained by the Sparse Label Sensitive Locality Preserving Projections are fed to a Discriminant Embedding that exploits the continuous labels. The obtained results show that the sparse representation with label similarity is an efficient method for data embedding, in the sense that it is easy to adapt to different datasets and that it only needs two parameters to tune.

In a crowded and uncontrolled environment observed by cameras from an un-known distance, person re-identification relying upon conventional biometrics such as face recognition is neither feasible nor reliable. Instead, visual features based on the appearance of people determined by their clothing, can be exploited more reliably for re-identification. This problem can be divided into two categories: single-shot and multi-shot approaches. Single-shot approaches are applied when tracking information is absent. In this context, we propose a new embedding scheme for single-shot person re-identification under non overlapping target cameras. First, a new representation is given

for each feature vector by projecting to a new linear subspace. Then, a collection of prototype is utilized to provide a measure to recognize an unseen object. The robustness of the algorithm against results that are counterintuitive to a human operator is improved by proposing the Color Categorization procedure. On the contrary, when tracking information is available, the existence of multiple images for each person makes it easier to train machine learning algorithms in general, and deep neural networks in particular. In the last part of this thesis, we propose a "Siamese" architecture of two Convolutional Neural Networks (CNN), with each CNN reduced to only eleven layers. This architecture allows a machine to be fed directly with raw data and to automatically discover the representations needed for classification. A comparison between learned features and hand crafted is provided, showing the superiority of the first one.

Contents

List of Figures

List of Tables

Chapter 1

General Introduction

Contents

1.1 Motivation

The human visual system effortlessly detects and classifies objects from among tens of thousands of possibilities. It is very effective at transforming the vast quantities of complex data into useful information, allowing us to perceive and model the environment. From an evolutionary perspective, our recognition abilities are not surprising, we draw on many years of implicit training and experience at processing the world around us. Our survival, depends on our precise and quick extraction of object features from the patterns of photons on our retina. A huge amount of effort has been put into the development of systems that replicate some of our visual recognition abilities using visual sensors and machine learning skills.

Computer vision is a rapidly growing field due to ceaseless advances in both camera technology and the computational power of modern computers. Due to the recent global security context, computer vision is gaining more popularity and installations of camera networks nowadays are widespread. The number of cases investigated by law enforcement, which have left some image related traces, is sharply increasing. The recent attack in Boston [1] (2013) demonstrated the effectiveness of video surveillance, where the criminals were identified by officials looking through camera footage. Nowadays, video surveillance systems have become ubiquitious in all domains ranging from the protection of small home in private areas, to securing administrative buildings, airports, public transportations and so on.

Searching for a given person of interest in thousands hours of footage across multiple cameras, requires to assign a large number of human operators to this task. Besides, humans are poorly equipped to perform repetitive tasks and find difficulties analyzing the massive amount of data generated by the system. For example, a 100 camera installation at 6 images/second generates 50 million images per day and 1 billion images in the database within 3 weeks. On the other hand, the video data provided as live streams require high awareness from operators. Normally, there is a huge disparity between the number of cameras and the number of operators. Operators are normally watching several displays at once, which increases the chance of missing important events. In fact, part of the problem is that sometimes one screen is dedicated to more than one camera and it displays the viewed scene of each of them periodically. Moreover, operator's attention often drops below desirable levels after a while of monitoring normal scenes where no important events occur. This is because of lapses in vigilance due to fatigue and distraction of attention.

A recent 2008 article investigated the relationship between the number of displays monitored by the operators and the precision of target detection [3]. It found that the operators missed 60% of the target events when they were monitoring 9 displays. In addition, when targets were detected, the probability of detecting another target within the same time frame was decreased. Furthermore, the miss rates were reduced to 20% when monitoring

1. Images from the real scenario can be found at: http://www.fbi.gov/news/ updates-on-investigation-into-multiple-explosions-in-boston/photos

only 4 displays. Another interesting study shows that after 20 minutes of focusing on surveillance screens, an operator will miss up to 95 percent of all activity [4]. And we are assuming that this is a motivated person being paid to do the job. In brief, in live surveillance mode the more scenes the person has to monitor, the more activity they are likely to miss because there are real limitations in people's ability to monitor several signals at once.

To overcome all these problems, efficient automated surveillance systems are required in order to filter out irrelevant information and highlight item of interest. Ideally, these systems should provide alert to the operators and point them toward suspicious events to give them the chance to investigate and take action. This research domain covers many tasks like tracking, object recognition, gesture recognition, behavior analysis and understanding. These are prominent and challenging tasks for several applications of Computer Vision, especially in surveillance systems.

1.2 Context Of Study

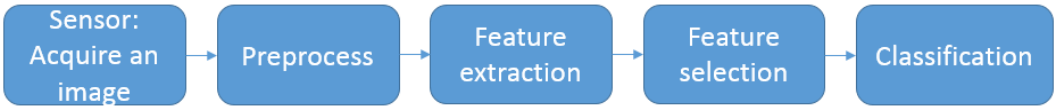


Figure 1.1: Object Recognition paradigm

Most image processing approaches used to recognize an object go through three main stages namely, feature extraction, feature selection and classification (see Figure 1.1). Prior to recognizing the object, one needs to detect the object. However prior to detecting the object one has to extract features of the object that can be compared against the features of a set of reference images stored in the database. In the hierarchy of object recognition, object detection typically precedes object recognition. Most approaches consider object detection step independently from the re-identification step and use different algorithms to achieve both tasks. In this thesis, we assume that objects have already been detected in all cameras and we do not tackle the

detection problem.

Images are a collection of pixels arranged in columns and rows. Each pixel has an intensity value representing the measured physical quantity such as solar radiance in a given wavelength band. These representations do not lend themselves well to semantic interpretation. Instead of using raw images intensities, one often extracts efficient and effective visual features and build models from them. The hand-crafted descriptors involved in extracting these features call for extensive trial-and-error experiments and substantial human expertise. In the context of visual person recognition, features extracted can be either based on biometrics (face, iris) or on non-biometric features (appearance).

Biometric features comprise an individual's unique characteristics and are therefore highly discriminative. They can be traced to 14th century in China, where Chinese merchants were stamping children's palm prints and foot prints on paper with ink to distinguish the young children from one another. Of course even biometric characteristics can change over time (the face changes with age for example). In practice however, especially in surveillance scenarios, we can rely on biometric features since they remain sufficiently invariant within the period of surveillance. Facial recognition can be considered within the scope of biometrics as it contains enough details to permit an individual to be distinctively identified. Lately, there has been notable improvements on automatic face recognition in controlled conditions. However, the efficiency in unconstrained conditions is still unsatisfactory.

Nowadays, there is an increasing need in real-world applications of visual person identification. It however comes with high demands on capability and robustness in realistic scenarios. Most of the approaches that simply count on biometric features cannot cope with challenges such as non-frontal faces and low quality images. There is always a trade-off amongst image quality and sensors price. The choice between a standard or higher resolution cameras depends on the necessity to capture more detailed images. Of course, high resolution images are fundamental to improve the performances of computer vision algorithms like face recognition and person re-identification. But most of the surveillance cameras are incapable of capturing high-resolution images due to the low resolution of low-cost cameras and the large distance between camera and human subjects.

Figure 1.2: Recognizing people is not trivial even for a human unless faces are shown in context with their clothing (images from [1])

Besides biometric cues, with the assumption that usually a person does not change his clothes with short time frames, a person's overall appearance is an alternative cue that can be used for person re-identification. This is an acceptable assumption for many scenarios such as tracking a person in a surveillance camera network. The use of that attribute in such conditions is referred to as appearance-based person re-identification. A person's overall appearance is a viable source of information, it is exploited by humans in a manner vastly superior to anything we are currently able to do with computers. Figure 1.2 shows the difficulty of using only the face to recognize people. Consider the upper row of the figure where six cropped faces from an image collection are shown. These images belong to only three different persons. Even for a human it is difficult to determine how many distinct persons are present. If however the faces are shown in context with their clothing, the task becomes almost trivial.

Person tracking/re-identification and face recognition systems should ideally integrate information to minimize identity switches across a network of cameras. A fully automated system should be capable of analyzing the face information when the target is near the cameras and tracking his identity in a video when his facial features are no more traceable. Two key issues make this challenging:

1. First, faces are often not captured frontally but from the side so that standard frontal face recognition cannot be applied directly.

2. Second, the appearance of an individual can vary extremely across a network of cameras and may cause the failure of the tracker. Mainly it is due to viewpoint and illumination changes, occlusions and cluttered background.

To address the first issue we intend to solve the face pose estimation problem using machine learning techniques. Based on the related work in the literature, a real-time system will be proposed. For the second issue, under the assumption that people do not change their clothes between different sighting in a network, we intend to design stronger human signatures which are then matched between individual camera views.

1.3 Face Pose Estimation

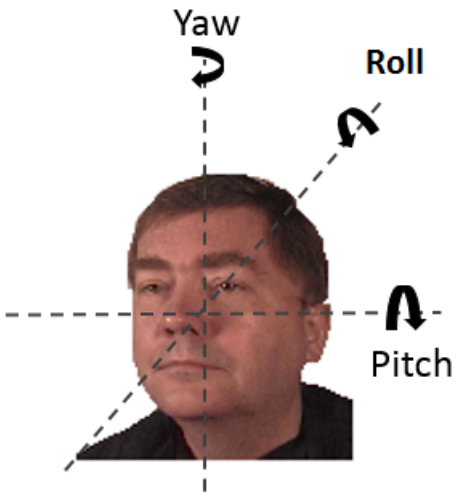


Figure 1.3: Face Pose angles

Face pose estimation is a problem related to the fields of computer vision, artificial intelligence and machine learning. The general application of these fields is to build machines which can perform intelligent behavior and solve problems for us. Head pose estimation has been a focus of research in computer vision both implicitly in tasks that use full body pose estimation, and explicitly in tasks that perform face tracking and face recognition. One of the major problems encountered by current face recognition techniques lies in the difficulties of handling varying poses. Hence, in order to make the face recognition more robust and to allow further face analysis, extensive efforts have been put to estimate the pose of the head. Figure 1.3 shows the three degrees of freedom of a human face which can be represented by the three rotation angles: alpha (yaw), beta (pitch) and gamma (roll). In this thesis we focus on the out-of-plane rotation problem, since in-plane rotation is a pure 2D problem and can be solved much more easily.

Face image based human recognition is now a relatively mature technology, especially in some controlled indoor situations. It is not surprising that face recognition systems are very sensitive to pose variation and their accuracies drop when the training and testing faces are not in the same pose. This problem was reported in the FERET and FRVT test reports [5, 6], and was stated as a principal research complication. Beside its critical role in face recognition, head pose estimation have lots of applications in the human-machine interaction domain. It is considered as an important cue of non-verbal communication since it gives information about the action and intent of a person. Indeed, humans can easily discover and understand other people's intentions by interpreting their head pose.

In video surveillance applications, the process of automatic head pose estimation is often just the first layer. It can be used as input to the final task which is typically Facial Recognition or other human-machine interaction tasks. However, in order to make a machine capable of interacting with the human's head movements, huge effort has to be done to estimate the pose from the pixel representation of facial images. In fact, the estimation of the position with respect to an observer attached to the camera, depends on the ability of the system to extract and track facial features reliably. It requires a series of processing steps to transform a pixel-based representation of a face into a high-level concept of direction. Moreover, the 2-D image measurements of facial images are usually altered by notable noise due to

diverse variations in the images like occlusion, facial variation (facial expressions, mustache, beard etc.), and other sources of variation in the image formation such as illumination. These variations make the feature extraction step very challenging and may cause large pose estimation uncertainties.

The first part of this thesis focuses on improvements to existing face pose estimation (specifically the Yaw angle) in terms of precision, computational expense, and invariance to different constraints (e.g. bad illumination, different skin colors, facial hairs, presence of glasses, etc). We try to estimate the yaw angle only due to the non-availability of databases that vary the pitch and roll angles, but the same techniques proposed in this work can also be applied to estimate the other angles. The terms "face pose estimation", "head pose estimation" and "pose estimation" have been used interchangeably in this thesis.

1.4 Person Re-Identification

Person tracking has been traditionally studied for surveillance purposes, where moving objects are detected and assigned to consistent labels as they move around a scene. To simplify the problem, researchers imposed constraints on the appearance and/or the motion. For example, almost all the tracking algorithms require a spatiotemporal continuity of objects so that they satisfy the following constraints: continuity (object's movement must be continuous) and exclusivity (an object cannot be in more than one place at the same time). A time delay may interrupt the continuity of an object's position over time, causing the failure of the tracker. Hence, when distinct images of objects are captured without enough temporal or spatial continuity, the re-identification process becomes the convenient approach to maintain the tracking. Another need for re-identification is the case of global tracking, where the object has to be identified after re-entering the field of view. Local tracking aims to track the object at a frame-to-frame level as long as it belongs to its field of view. Global tracking aims to find the corresponding label of some object in its earlier appearances.

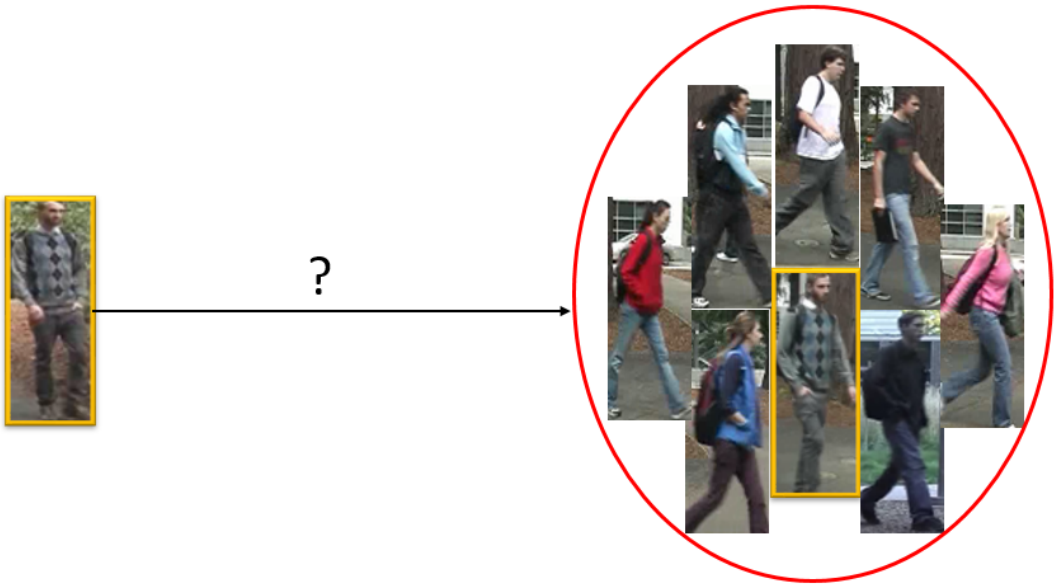


Figure 1.4: Person re-identification problem

Most tracking approaches encountered in the domain of computer vision are targeted at single-view applications and work on a frame to frame basis, which means that they are capable to maintain the label of a person only within the field of view of one camera. While a single camera might be enough to cover very small areas of interest, additional cameras quickly become inescapable as larger sites have to be monitored. If the architecture of the site allow establishing a camera network with enough overlapping fields of view, inter-camera tracking can be accomplished by spatio-temporal analysis. Nevertheless, in most realistic contexts, the needed number of cameras and related costs would be too high, so that the coverage is fairly sparse, causing "blind gaps".

Person re-identification, a central task in many surveillance scenarios, is the capability of associating a new observation of a person to others made in the past. It can be defined as recognizing a person in different locations over a network of non-overlapping cameras, enabling person tracking within the full monitored area. State-of-the-art person re-identification methods

are mostly based on global clothing and body appearance. Face recognition, that is potentially more effective, is often unpractical in video surveillance due to low resolutions of pictures, the presence of occlusions with objects and strong variations of illumination. Re-identifying persons that are in a region of interest is a high level capability that is critical in many fields beside video surveillance, like service robotics and smart environments.

In this thesis, we emphasis on person re-identification across blind gaps. Especially, we tackle the problem of choosing a target in one camera view and then recognizing the same target in another camera view without simplifying the problem by spatio-temporal analysis based on the prior knowledge of the scene. Moreover, we suppose that persons have been detected in all images as we have already mentioned. Figure 1.4 illustrates an example, where a person is selected from the images of the first camera, and the task is to automatically re-identify the same person across all the images in the other camera view. Generally, most approaches generate a ranked vector of all the persons in a gallery set (images of labeled persons) based on their resemblance with the probe image (unlabeled person). The highest similarity score in the ranked vector will assign a specific label for the probe image. Depending on the setup, we also have to differentiate between the two categories of person re-identification. The first category known as 'single-shot approach' [7], focuses on connecting pairs of images, each containing one instance of an individual. The second category uses multiple images of the same person as training data and considers short sequences for testing. It is known as 'multiple-shot approach' [8]. It is evident that the latter category offers richer information because each person is captured in several different poses. But on the other hand, such systems necessitate more sophisticated algorithms to be able to utilize the supplementary information, so that the runtime is noticeably higher than the single-shot case.

1.5 Contributions

The main contributions of this thesis are mainly related to improve some major components of a vision-based security system, namely, re-identification of a tracked suspect through a network of non-overlapping video cameras and uncontrolled face recognition. Both problems are difficult as the target is not

cooperating in a high-challenging uncontrolled environment. For instance, a varying face pose could deteriorate even high performance face recognition algorithms. In the following, we briefly sketch our main contributions concerning face pose estimation (as input to face-recognition algorithms) and appearance-based person re-identification through a set of non-overlapping cameras.

1.5.1 Face Pose Estimation

In Chapter 2, we propose a new embedding scheme for image-based continuous 3D face pose estimation. An adaptation and an extension to the existing state-of-the art approach for face pose estimation are presented.

First, we show that the concept of label sensitive Locality Preserving Projections, proposed for age estimation, can be used for modeless face pose estimation. An adapted version is proposed by building an affinity matrix taking advantage of two relationships. The first one is a spatial relationship consisting of the euclidean distance between the data points in the high dimensional feature space. The second one is a label relationship consisting of the euclidean distance between the poses of the data points.

Second, we provide a linear embedding by exploiting the connections between facial features and pose labels via a sparse coding scheme. In fact, we constructed a graph similarity matrix via a weighted sparse coding that integrates label sensitivity. The resulting technique is called Sparse Label Sensitive Locality Preserving Projections (Sp-LsLPP). Our method has less parameters compared to related works, making the adaptation process on different dataset easier and more practical.

Third, inspired by the framework of Linear Discriminant Embedding, the projections obtained by the Sparse Label sensitive Locality Preserving Projections are fed to a Discriminant Embedding that exploit the continuous labels. This latter enhance the discrimination between poses by simultaneously maximizing the local margin between heterogeneous samples and pushing the homogeneous samples closer to each other.

Finally, we demonstrated the benefits of our proposed approach on a number of publicly available datasets. It was conveniently compared with other

linear and non-linear techniques, confirming that out proposed frameworks can outperform, in general, the existing ones. This work has been published in [9] and [10].

1.5.2 Person Re-Identification

The key contribution here is a new method to perform person re-identification in a visual surveillance context. Under reasonable assumptions holding in many real contexts, two approaches are proposed to re-identify people under non-overlapping target cameras. The first approach is based on hand-crafted feature extraction and a second approach based on a data-driven deep-learned representations. Both approaches have been compared and suggestions have been given according to the application context.

Hand-crafted feature based approach

First, we explore the adaptation of the Prototype Formation in the person re-identification problem. It was proposed in psychology and cognition field [11], and tested on Face Recognition problem [12]. It suggests that human being categorizes the objects based on hierarchical prototypes, and people differentiate the world using this critical skill for category learning. Psychological experiments revealed that human brain recognizes and differentiates objects using prototypes. It means that prototypes provide a measure to recognize or classify an unseen object. Based on that, we propose an approach for person re-identification where each person is described as a vector of kernel similarities to a collection of prototype person images.

Second, we propose an additional Color Categorization step to overcome one common weak point in previous approaches: mistakenly positioning two persons who wear different colors of clothes above the true match. In fact, lighting or background changes have a huge influence on a person's appearance. These changes can make different persons appear more alike than the same person across different camera views. Experimental results demonstrated the benefits of this method by achieving results that are closer to what humans consider intuitive.

Third, to ensure that features extracted have favorable discriminative ca-

pability, we propose a novel discriminant method and we show the discrimination that can be provided using the Quaternionic Local Binary Pattern (QLBP) as a feature vector rather than traditional Local Binary Pattern (LBP) which neglects the relation between color channels. After extracting a feature vector representing each image, a linear subspace is learned, promoting the separability between different persons and minimizing the distances between two representations belonging to the same person. This work has been published in [13] and [14]

Deep Learning based approach

Finally, the concept of end-to-end learning or learning from pixel level to person re-identification has been presented in Chapter 4. A siamese Convolutional Neural Network has been trained from scratch generating a similarity metric. A loss function is defined based on the similarity of the learned features. Unlike hand crafted features, this deep architecture is more practical since all sophisticated features are learned at once without the need of multiple methods combination to achieve the task. A comparison between learning based features with the handcrafted ones is made in a single shot scenario. The study shows the superiority in terms of performance for the learned features. However, approaches based on handcrafted representation can still be important in scenarios where the learning database is not sufficient to train deep networks.

1.6 Outline

This PhD manuscript is organized into 5 chapters as follows:

- Chapter 2 starts by reviewing existing works in the field of face pose estimation, organizing them according to the strategy they use to tackle the task. A review of the label sensitive Locality Preserving Projections is given before presenting the proposed framework. The final part of this chapter is dedicated for experimental results and discussions.

- Chapter 3 introduces our proposed approach for appearance based Person re-identification based on hand-crafted features. A general overview of the whole framework as a complete processing chain is given, provid-

ing theoretical and conceptual background on color, feature selection and classification techniques. It outlines the potential benefit of prototype formation and color categorization by providing the corresponding experimental results.

- Chapter 4 outlines a practical implementation of a re-identification system based on a Deep Learning architecture. It starts by presenting the basic operational principles of these architectures. Then, it explains how the similarity metric has been learned by training a siamese Convolutional Neural Network. At the end, a comparison between learned and hand-crafted features is presented.

- Chapter 5 summarizes the thesis and make some concluding remarks and limitations. We also discuss about the short-term and long-term perspectives of this study.

1.7 Publications

The following journal and conference papers have been produced as parts of outcomes of this research:

1.7.1 Journal

- "C. Chahla, H. Snoussi, F. Abdallah, F. Dornaika, Discriminant quaternion local binary pattern embedding for person re-identification through prototype formation and color categorization, Engineering Applications of Artificial Intelligence, Volume 58, February 2017, Pages 27-33, ISSN 0952-1976"
- "F. Dornaika, C. Chahla, F. Khattar, F. Abdallah, H. Snoussi, Discriminant sparse label-sensitive embedding: Application to image-based face pose estimation, Engineering Applications of Artificial Intelligence, Volume 50, April 2016, Pages 168- 176, ISSN 0952-1976."

1.7.2 Conference

- "C. Chahla, H. Snoussi, F. Abdallah,F. Dornaika, Exploiting Color Strength to Improve Person Re-identification, 7th IET International

Conference on Imaging for Crime Detection and Prevention, ISBN 978-1-78561-400-2 ”

- ”C. Chahla,F. Dornaika, F. Abdallah, H. Snoussi, Sparse feature extraction for model-less robust face pose estimation, International Conference on Sensors, networks, smart and emerging technologies (SENSET2017),” (Accepted)

Chapter 2

Face Pose Estimation

Abstract

In this chapter, we describe our proposed framework to estimate the pose of the head. We start by giving a review on the common solutions of this problem. Then we describe our algorithm relying on a sparse representation. Finally, we compare its performance with other linear and non-linear techniques.

Contents

2.1 Face Pose estimation from images: A review

2.1.1 Introduction

Face image analysis has attracted increasing attention in the computer vision community. It is required for developing artificial systems able to per-

form intelligent behavior such as face recognition and annotation [15, 16, 17, 18, 19], facial landmark annotation [20], age estimation [21], or face pose estimation[22]. The pose estimation process requires a series of processing steps to transform a pixel-based representation of a face into a high-level concept of direction. 3D face pose can play an important role in many applications [23]. For instance, it can be used in the domain of face recognition either by using hierarchical models or by generating a frontal face image. The head pose estimation refers to the specific task consisting of determining the position and/or the orientation of the head in an image (e.g. a facial one). This task is a challenging problem because there are many degrees of freedom that should be estimated.

During the past years many techniques and algorithms have been proposed to estimate the pose of faces from images. Murphy-Chutorian and Trivedi have conducted a very good survey of the proposed techniques in [22]. The majority of work in 3D face pose estimation deals with tracking full rigid body motion. This requires the estimation of 6 degrees of freedom of the face/head in every video frame. This can be successful for a limited range of motion (typically $\pm 45^{\circ}$ out-of-plane) and only for relatively high resolution images [24]. Such systems typically rely on a 3D model that should be fitted to the person specific shape [25, 26]. There is a tradeoff between the complexity of the initialization process, the speed of the algorithm and the robustness and accuracy of pose estimation. Although the model-based systems can run in real-time, they rely on frame-to-frame estimation and hence are sensitive to drift and require relatively slow and non-jerky motion. These systems require initialization and failure recovery. For situations in which the subject and camera are separated by more than a few feet, full rigid body motion tracking of fine head pose is no longer practical. In this case, model-less pose estimation can be used [27, 28]. This approach can be performed on a single image at any time without any model given that some pose-classified ground truth data are previously learned [29, 30]. In the following, we will give an overview over the main categories that have been used to etimate the face pose.

2.1.2 Geometric Methods

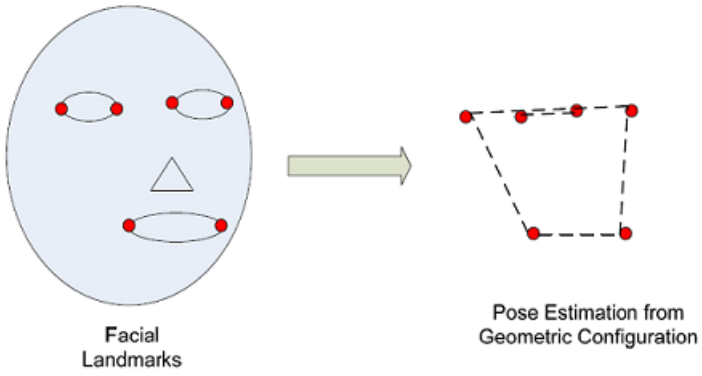


Figure 2.1: Geometric Methods

Geometric methods [31, 32] rely heavily on the estimation of facial features, such as eyes, mouth corners, nose tip, etc. and use their relative position to estimate the pose using projective geometry. For example, if the eyes and the mouth form an isosceles triangle, then the image corresponds to a frontal view. Schematic representation of geometric methods is illustrated in Figure 2.1. The major disadvantage of these methods is the requirement of detecting the face features in a very precise and accurate way. They also need to handle missing facial features in some poses. In contrast, these methods are considered simple and computationally inexpensive.

Different geometric approaches use the precise configuration of local features and the head shape in different ways to estimate face pose. In [33], authors used five feature points (the outside corners of eyes, outside corners of mouth and tip of nose) to estimate pose. These features and other elements, such as the position of the face in relation to the contour of the head, greatly affect the human perception of head pose. A symmetry line is usually drawn by joining the middle of two segments: one drawn between two eyes corners and the other drawn between mouth corners. Considering a constant ratio between these facial points and a predefined length of the nose, the facial angle is estimated from 3D angle of the nose. In [34], they used different five feature points (the inner corners of the eyes instead of the outside corners of the mouth). In order to estimate the head pose, they considered that all the

feature points are co-planner so that the yaw angle is calculated from the difference in length of the two eyes. Roll and pitch angles are calculated by comparing the distance between the line joining the two eyes and the nose tip to an anthropometric model.

Another geometric method was presented in [35]. This method uses six feature points instead of four (the inner and outer corners of each eye and the corners of the mouth). The scheme is based on the observation that three lines between the outer eye corners, the inner eye corners, and the mouth are parallel. Any observed deviation from parallel in the image plane is a result of perspective distortion and can be used to estimate the face pose. The vanishing point (i.e., where these lines would intersect in the image plane) are computed using least squares to minimize the overdetermined solution for three lines. This point can be used to predict the 3D orientation of the parallel lines if the ratio of their lengths is known and it can be used to calculate the position of each feature point if the actual line length is known. The EM algorithm with a Gaussian mixture model can adapt the facial parameters for each identity to minimize the back-projection error. The downside to this approach is that they assume the three lines to be visible which makes this method applicable only when the pose is near enough to a frontal view to see all of the facial lines.

Another estimate of pose can be obtained based on the position of pupils and nostrils. Infrared LEDs capture dark and bright pupil images. Subtracting the dark image from the bright image localize the pupils. Two normal vectors are then calculated, the first one from two eyes and right nostril plane, and the other one from two eye and left nostril plane. These vectors are used as an indication to calculate the face pose [36]. Recently, researchers investigated the 3D sensing technologies for face pose estimation [37, 38]. Although this technology promises a lot in overcoming some of the problems of methods based on 2D data by using the additional depth information, it suffers of serious computational problems. It cannot handle large pose variations, it cannot run in real time and it need manual initialization. Furthermore, they are not as scalable as the 2D sensors providing 2D images. Other approaches incorporate both 2-D and 3-D information using a face normal vector. They derive face features (eyes, nose, lips) from 2-D from which the face normal vector in obtained in 3-D space. The process of estimating the face pose from this normal vector is iterated until a given accuracy is satisfied [39].

It is worth mentioning here that even very simple cues can be effective to the face pose estimation problem. Fitting an ellipse to the gradient contour of a face can give a coarse prediction of pose for one DOF [40]. With many cameras surrounding the head, yaw can be reliably estimated as the orientation with the most skin color [41] or with a skin color template [42]. For virtual reality use, various markers are positioned on the frame of the eye glasses [43]. To prevent the distraction of the user, IR illumination are used in conjunction with IR reflective. The orientation of these markers can be used to estimate the face pose. One main disadvantage of this method is that it is only applicable to their specific virtual reality environment.

Most of these geometric methods have the advantage of being fast and simple. With only a few facial features, a decent estimate of head pose can be obtained. However, one main disadvantage remains in detecting the features with high precision. Hence, the lack of accuracy or errors in localization can greatly degrade the performance of these approaches. Low resolution images, in this context, are problematic since it is impossible to accurately localize the features. Similarly, situations where facial landmarks are occluded, such as when a person wears glasses, are not practical for this kind of methods.

2.1.3 Appearance Methods

Appearance template methods use similarity algorithms and compare a given image to a set of exemplars in order to discover the most similar image [44, 45]. Figure 2.2 shows an illustration of appearance methods. In the simplest implementation, the input image is given the same pose of the template that is associated to the most similar of these templates. The use of normalized cross-correlation at multiple image resolution [46] and mean square error (MSE) over a sliding window [47] are some basic examples of appearance methods.

One of the main advantages of these methods is that they are suitable for both low and high-resolution images. Moreover, it is not hard to adapt these methods to changing conditions. This can be achieved by simply adding more templates and expanding the dataset. Unlike face detection problem, appearance methods do not need negative training examples. Training data are created by cropping head images and providing the corresponding pose

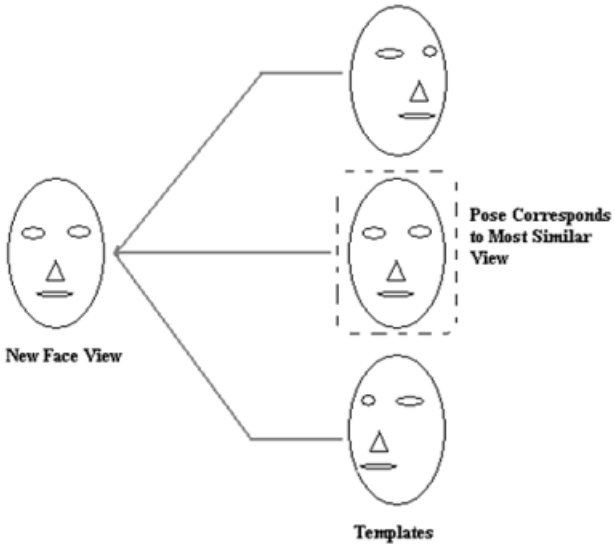


Figure 2.2: Appearance Methods

labels.

Nevertheless, even if these methods have the advantage of not requiring a feature extraction step, they may suffer from noise caused by illumination and expression changes in addition to the need of high computational power since the matching process they use is based on pair-wise similarities. Furthermore, these methods are applicable only to discrete poses and they cannot be used to the continuous pose estimation problem without the need of some interpolation method. Also, they generally consider that the head has already been localized, which means that any localization error can decrease the precision of these methods. To deal with the latter problem, the authors of [48] and [49] proposed to train a set of Support Vector Machines (SVMs) to detect and localize the face before using the support vector as appearance templates for pose estimation.

Apart from those weak points, the most significant disadvantage of appearance template methods is that they assume that pair-wise similarity in image space can be equated to similarity in pose. Suppose we have two images of distinct persons but with the same pose in one case, and two images

of the same person but with different poses in another case. In this context, the identity effect can cause more dissimilarity in the image than from a change in pose which can lead to erroneous pose estimation. To overcome the effect of pair-wise similarity problem, [50] suggested to convolve images with a Laplacian-of-Gaussian filter to emphasize some of the more common facial contours while removing some of the identity specific texture variation. In the same way, [51] suggested to convolve images with a complex Gabor-wavelet to emphasize directed features like the horizontal line of the mouth. The magnitude of this convolution is invariant to shift which can greatly reduce the error produced due to variance in facial feature locations between different persons.

2.1.4 Regression Based Method

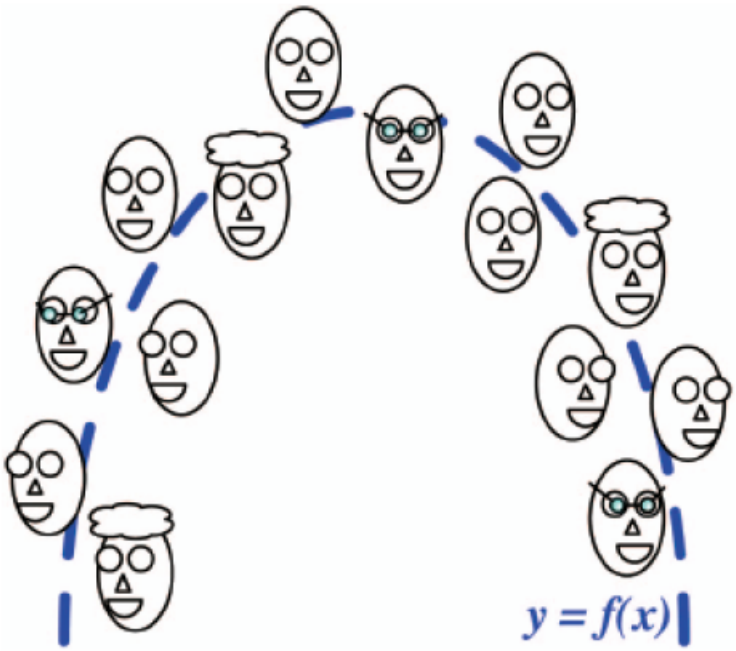


Figure 2.3: Regression Methods

Regression-based methods [52] allow to obtain continuous pose estimates. An illustration is provided in Figure 2.3. Indeed, they use regression techniques [53, 54] in order to find the relationship between the face image and its corresponding pose and to earn continuous mapping functions between the face image and the pose space. The weak point of these methods is that it is difficult to obtain an ideal mapping using regression.

The high dimensionality of the data represents an important challenge in this kind of methods because of the well-known "curse of dimensionality" problem [55]. Many solution have been proposed to reduce the dimensionality using the Principal Component Analysis (PCA) for example before using Support Vector Regressors (SVRs) [56]. Others used localized gradient orientation histograms to reduce the dimensionality [57] which gave more precision in the estimation process. On the other hand, if the location of facial features are predefined, the regression methods can be applied on relatively low dimensional feature data extracted at these points [58, 59].

One of the most popular nonlinear regression procedures used in the literature of face pose estimation are neural networks. Commonly, a multi layer perceptron (MLP) is trained with backpropagation which is a supervised training method that propagates the error through each layer to update each of the weights and biases of these layers. The latter model can be used as a feature extractor and a classifier to any new face image by giving it a discrete pose label. To take in consideration the similarity in the training images corresponding to similar poses, a Gaussian kernel can be applied [60, 61].

To obtain continuous pose estimates, a MLP is trained with one output for each DOF [62, 63]. The output's value is proportional to the assigned face pose angle. Other methods trained several MLP networks each with a single output corresponding to each DOF. The head region is detected by a color filter or a background substraction and a Bayisan filter is used to smooth each individual camera [64, 65]. Local-linear map (LLM) is another common neural network widely used for head pose estimation [66]. The input image is compared with the centroid of each of the linear maps constituting the LLM network in order to build a weight matrix. Finally, a regression technique succeeds the search of the nearest neighbors to estimate the poses. In [67], the authors proposed to boost the latter work by Gabor wavelets

decomposition.

The main advantage of these methods is that they are very fast and only need images of cropped faces with the corresponding poses. These approaches are very practical since they give high precision in estimation for both near and far field images. Just like the appearance template methods, these methods are very vulnerable to the head localization errors. To overcome this problem, the work of [68] proposed a convolution network that artificially model some shift and scale that can reduce this source of error.

2.1.5 Manifold Learning

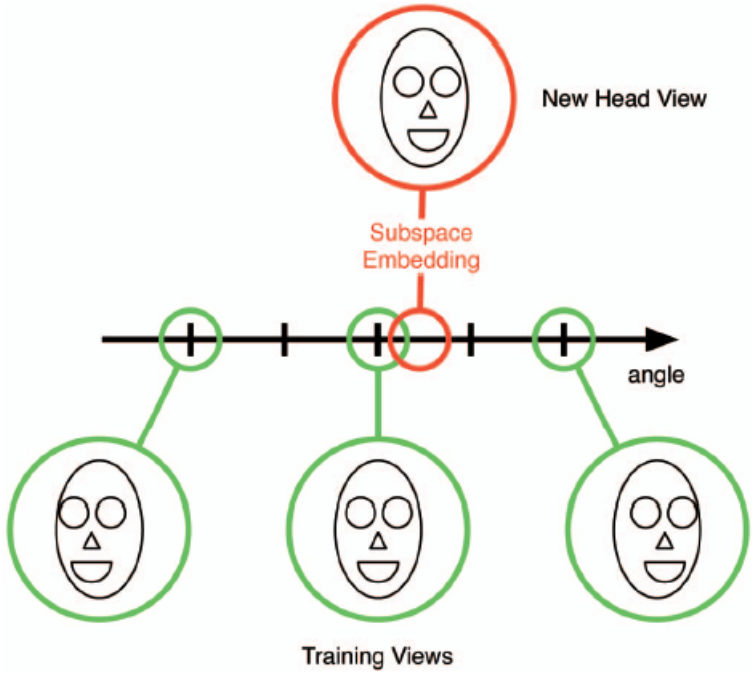


Figure 2.4: Manifold Methods

The manifold embedding methods [69] consider face images as samples of a low-dimensional manifold embedded in the high-dimensional observation space (the space of all possible images) (Figure 2.4). They try to find

a low dimensional representation that is linked to the pose. After properly modeling the manifold, an embedding technique is used to embed the test sample on this manifold. After that, classification or regression techniques are applied to discover the pose. All the dimensionality reduction methods can be considered as manifold embedding but the challenge is to design an approach that can ignore all erroneous sources in the image in order to precisely determine the pose.

The well known principal component analysis (PCA) technique has been widely used for pose estimation problems. In [70], the authors reduced the dimensionality of the face image by projecting it to a subspace using PCA, then they compared this representation with a set of reference samples to estimate the pose. The work of [71] showed that the similarity in the PCA subspace correlate more with pose similarity than appearance template matching with Gabor wavelet. The unsupervised nature of PCA and its nonlinear version Kernel PCA does not guarantee that the primary component is correlated to pose estimation rather than to appearance variation. As a solution for this problem, the work of [72] proposed to split the training data into different groups where each group shares the same pose. As a result, the appearance information is decoupled from the pose and PCA can be used to create different projection matrices for each group. Thus, the head pose is determined after projecting the image using each of the projection matrices and selecting the pose with the highest projection energy. Another method is proposed in [73], where the projected sample is used as input to Support Vector Machines (SVMs). The work of [74] showed that a better performance can be achieved by using Gabor binary patterns with a set of multi-class SVMs. Since the estimation is based on discrete measurments, pose-eigen spaces cannot be used for continuous pose estimation.

The heterogeneity of the samples is considered as one of the main challenges in real-world scenarios. Many persons are needed to train a manifold, but it is difficult to get a regular sampling poses for each person. To overcome this problem, individual submanifolds are created and used to reconstruct missing poses for each subject. This work was introduced in the Synchronized Submanifold Embedding (SSE) in [75]. This proposed algorithm is dually supervised by both identity and pose information. The submanifold of each subject is approximated as a set of simplexes constructed using neighboring

samples, and the pose label is further propagated within all the simplexes by using the generalized barycentric coordinates. Then these submanifolds are synchronized by seeking the counterpart point of each sample within the simplexes of a different subject, and consequently the synchronized submanifold embedding is formulated to minimize the distances between these aligned point pairs and at the same time maximize the intra-submanifold variance. Finally, for a new datum, a simplex is constructed using its nearest neighbors measured in the dimensionality reduced feature space, and then its pose is estimated as the propagated pose of the nearest point within the simplex.

2.1.6 Detector Arrays

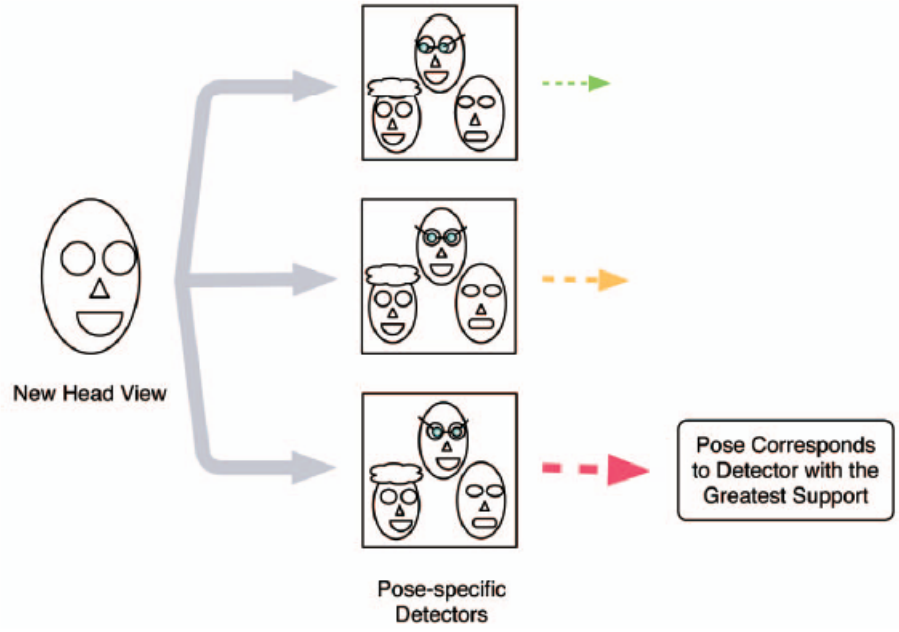


Figure 2.5: Detector Array Methods

After the notable success of face detection using frontal face images [76, 77], many methods have been suggested to extend these detectors to non frontal faces. Detector arrays method train many face detectors, one for each discrete pose. Figure 2.5 illustrates these methods. The probe image will be the input for all these detectors, and the image will be assigned to the

pose of the detector that has the maximum success. In other words, instead of comparing the input image to a set of large templates like in appearance methods, detector methods train a detector on many images with different poses. For example, the work of [78] used three support vector machines for estimating three discrete yaw angles. Since each detector can differentiate head and non-head regions, detector array methods do not require a head detection step. Unlike appearance templates, these methods can be trained to ignore the appearance changes corresponding to the identity effect and not the pose. Another advantage of these methods is that they are suited for low and high resolution images.

The main disadvantage of these methods is that they are computationally expensive since they train a large number of detectors. It is also challenging to train a detector to be used as a face detector and pose estimator in the same time since it requires negative non-face training examples, which means a larger training set. Moreover, increase in the number of detectors may lead to systematic problems. For example, when two different detectors are trained for two similar poses. The positive example for one must be negative example for the other which can cause a failure in the model. This is why in practice the detector array methods do not use more than twelve detectors. In addition, these methods are limited to discrete pose classification and are not efficient for continuous estimation because the face detectors have usually binary outputs. Finally, the computational expenses grow linearly with the number of detectors, making it impossible to design a real-time system. [79] suggested a solution for this problem by using a router classifier which choose one detector to estimate the pose. In this case, the router effectively determines the pose and the subsequent detector verifies it. It should be noted that this method has not been demonstrated for yaw or pitch changes but rather only for rotation in the camera plane, first using neural networks in [79] and later with cascaded AdaBoost [80].

2.1.7 Deformable Methods

These methods try to design a flexible model for the image in a way that it fits to the structure of the face. The deformation of the model is used as an information to estimate the pose. They are very often used in tracking the face pose and some facial actions. Unlike appearance methods, these approaches rely on tuning the model such that it conforms to the facial

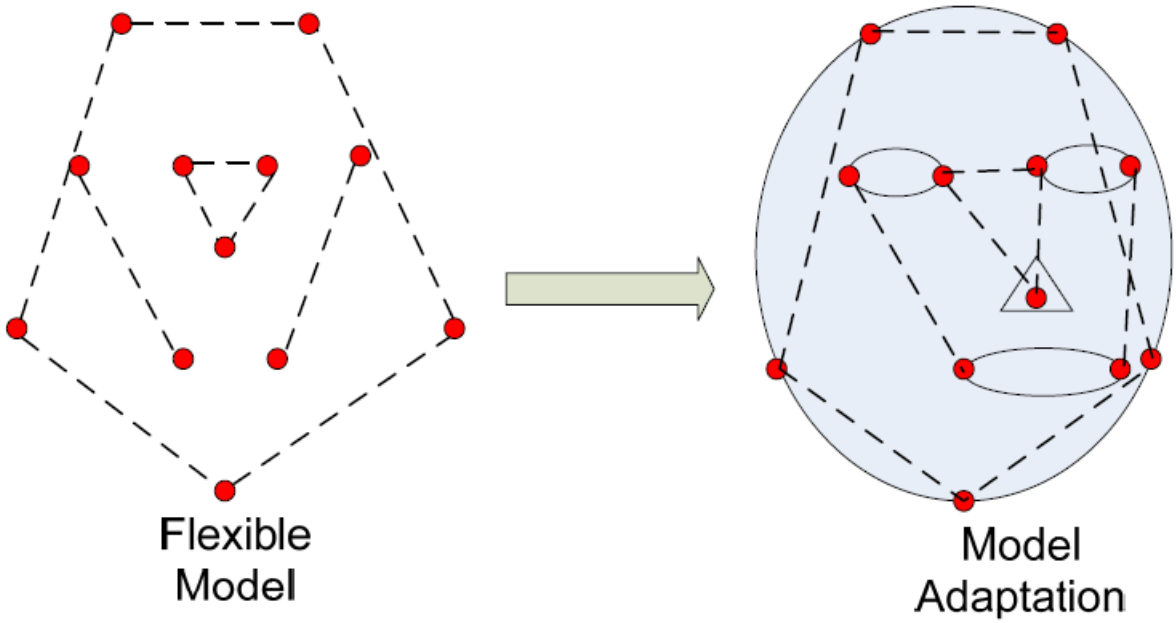


Figure 2.6: Flexible Methods

features of each person. In case of appearance based models, the chance of getting a perfect overlap of two images belonging to different persons is very low. Flexible methods do not suffer from this problem since they do not rely on matching a rectangular region of the image. In these methods, even if the facial features do not match perfectly due to inter-subject variation, the shape of the deformed model is more similar for different individuals. The schematic illustration of flexible methods is presented in Figure 2.6.

Flexible methods require training data with annotated facial features and not only the corresponding face poses. In the training phase, facial features of each image are labeled and local features are extracted at each location. To improve the robustness and invariance to inter-subject variations, the training is done using a large set of data and features are extracted from different views. The latter model is called Elastic Bunch Graph [81]. In order to compare a new test image to an elastic bunch graph, the graph is placed over the

image and is deformed so that each node overlaps with its relevant feature point location. This process is called Elastic Graph Matching (EGM). In the case of face pose estimation, for each discrete pose an elastic bunch graph is created and the test image is compared to each one of them. The image is assigned to the pose corresponding to the graph which gives maximum similarity [82, 83].

Active Appearance Model (AAM) [84] is another well known flexible method which learns primary modes of variation in facial shape and texture from 2D perspective. Consider N facial points such as eyes corners, ear tips, mouth etc. These points can be arranged in a vector of length 2N based on the coordinate of each point. These vectors can be used for face pose estimation by comparing variation in facial shape after calculating the features form many different faces. The application of PCA on this data will result in an Active Shape Model (ASM) [85] which can represent shape variation. A joint shape and texture AAM can be presented by generating an ASM model first before adapting the images such that the feature points match those of the mean shape. The images are normalized and then used to build a shape-free texture model. Finally, the correlation between shape and texture is learned and used to generate a combined appearance model [86]. To find the pose of a test image, the combined model is fitted to the image by iteratively comparing the model to the observed image and tuning the parameters to minimize the distance between the two images.

Since AAMs adapt a statistical deformable model to the images by localizing the exact feature points, these approaches are more robust and invariant to face localization error. The main limitation of AAMs is that facial features localization is required in the training stage and they fail if the outer corners of the eyes are not visible.

2.1.8 Hybrid Methods

Hybrid techniques mix two or more of the previous methods to estimate pose, as illustrated in Figure 2.7. They combine many techniques trying to overcome the limitation of each of individual techniques. A common Hybrid approach is to use a static face pose estimation technique for the first frame of a video combined with a tracking algorithm to track the pose over time. When the tracker lose the pose, the static method can reinitialize the system.

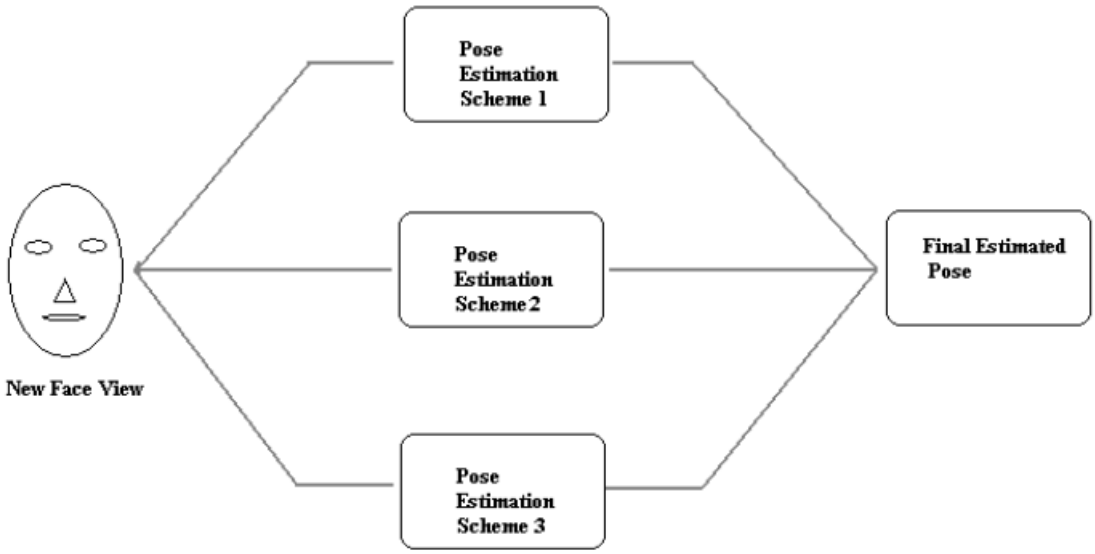


Figure 2.7: Hybrid Methods

Many researchers have presented successful hybrid methods like combining Geometric methods with point tracking [87, 88, 89], combining PCA with optical flow [90] or with a Markov model [91]. The work of [92] presented a combination of manifold embedding and flexible methods where they used Elastic graph matching to refine the pose calculated by manifold embedding. Hybrid approaches can also use many methods independently and then combine the result to enhance the precision of the estimation [93, 92].

The main advantage of these techniques is that they overcome the limitation of each of the previously discussed methods allowing to enhance the precision of the face pose estimation. Nevertheless, the combination of several methods increase the computational expenses and are not suitable in real-time applications.

2.2 Manifold learning: related work

While a face image is considered as data with a high dimensionality, only few dimensions are affected by the pose variation. Hence, we can suppose that each high dimensional image lies on a low dimensional manifold which can be used to estimate the pose. Manifold learning methods are used to reduce the dimensionality of the space and thus enhancing the performance of subsequent image-based tasks.

The classic linear embedding methods such as Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA) [94], Maximum Margin Criterion (MMC)[95]) are proved to be computationally efficient and suitable for practical applications, such as pattern classification and visual recognition. PCA projects the samples along the directions of maximal variances and aims to preserve the Euclidean distances between the samples. Unlike PCA which is unsupervised, LDA [94] is a supervised technique. One limitation of PCA and LDA is that they only see the linear global Euclidean structure. In addition to the Linear Discriminant Analysis (LDA) technique and its variants [96, 94], there is recently a lot of interest in graph-based linear dimensionality reduction. Many dimensionality reduction techniques can be derived from a graph whose nodes represent the data samples and whose edges quantify the similarity among pairs of samples [97, 98]. Recent proposed methods attempt to linearize some non-linear embedding techniques. This linearization is obtained by forcing the mapping to be explicit, i.e., performing the mapping by a projection matrix. For example, Locality Preserving Projection (LPP) [99] and Neighborhood Preserving Embedding (NPE) [100] can be seen as linearized versions of Laplacian Eigenmaps (LE) and Local Linear Embedding (LLE), respectively. The main advantage of the linearized embedding techniques is that the mapping is defined everywhere in the original space.

2.2.1 LsLPP adapted to face pose estimation

In this section, we review the LsLPP (Label-sensitive Locality Preserving Projections) method described in [21]. This method is a graph-based linear embedding technique. It can be considered as a supervised version of LPP (Locality Preserving Projections) method when the sample label is given by a real score. More precisely, the authors of [21] introduce a new framework for LPP and uses the resulting embedding for age estimation from facial images.

They divided a new affinity graph in which the edges weights explore the connections between facial features and age labels. This work introduces the label-sensitive concept that makes the affinity matrix dependent on both face image similarities and label similarities (age order). LsLPP method adopts the LPP framework [99, 101] in which the affinity matrix is built upon the use of both feature and label similarities. The proposed embedding is expected to better exploit the ordinal relationship among age labels. A label-sensitive concept is introduced, which regards the label similarity during the training phase of LPP.

LsLPP is a dimensionality reduction algorithm that aims to reduce the dimensionality of the data while preserving the relationship that relates the data in the high dimensional space. The relationship preserved by the LsLPP algorithm can be divided into two types:

- A spatial relationship which consists of the Euclidean distance between the data points in the high dimensional feature space.
- A label relationship which consists of the Euclidean distance between the labels (age) of the data points. In image-based age estimation, there exist the ordinal relationship and correlations among ages; for example, an age of 30 is closer to age 25 than to age 10.

Since face pose is a continuous variable, the LsLPP framework of [21] can be also used for face pose estimation. Similar labels (poses) are defined by the threshold ε, and the weights of the edges are computed in a way that takes into account the pose similarity. The LsLPP algorithm for face pose estimation is summarized in Figure 2.8.

Once the linear transform is known, any datum $\mathbf{x}$ can be projected onto the new subspace of dimension p using $\mathbf{z} = \mathbf{W}^T\mathbf{x} \in \mathbb{R}^p$. One can notice that all steps of the above algorithm are similar to those of a classic LPP algorithm. The main difference is in the computation of the similarity matrix $\mathbf{B}$ which incorporates both the features and labels. The LsLPP algorithm has four parameters (k_1, σ, t, ε) that need to be tuned in order to obtain the most accurate estimation possible.

Algorithm 1: Ls-LPP

Input: Training set represented by the data matrix $\mathbf{X} = \{\mathbf{x}_i \in \mathbb{R}^d\}_{i=1}^N$, labels $\mathbf{y} = \{y_i\}_{i=1}^N$ representing the real-valued pose (angle) of each sample, number of neighboring samples k_1, label-sensitive threshold ε, that defines the range of similar labels, parameters t and σ.
Output: A linear transform matrix $\mathbf{W}$

Presetting:

– Define the similar-label set $N^+(i)$ for each sample $\mathbf{x}^{(i)}$ as: $N^+(i) = \{\mathbf{x}_j\,, \|y_i - y_j\| \leq \varepsilon \text{ and } j \neq i\}$ where ε is the label-sensitive threshold that defines the range of similar labels.
– Set the $N \times N$ sample similarity matrix $\mathbf{B}$ to zero, i.e., $\{b_{ij} = 0\}_{(1 \leq i,\, j \leq N)}$

Algorithm:

– For each sample $\mathbf{x}_i$ find its k_1-nearest samples in $N^+(i)$ based on the Euclidian distance. Denote this sample set by as $KNN^+(i)$. Thus, the set $KNN^+(i)$ contains all samples that satisfy both feature similarity and label similarity with respect to the current sample $\mathbf{x}_i$.
– For each sample pair $\{\mathbf{x}_i, \mathbf{x}_j\}$, if $\mathbf{x}_j \in KNN^+(i)$ or $\mathbf{x}_i \in KNN^+(j)$ set: $B_{ij} = \exp(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{t}) \exp(-\frac{(y_i - y_j)^2}{\sigma})$
– Compute the Laplacian matrix associated with graph $\mathbf{B}$ as $\mathbf{L} = \mathbf{D} - \mathbf{B}$, where $\mathbf{D}$ is a diagonal matrix whose elements D_{ii} are computed by: $D_{ii} = \Sigma_j\, B_{ij}$
– Solve the generalized eigen decomposition problem: $(\mathbf{X}\mathbf{L}\mathbf{X}^T)\mathbf{v}_i = \lambda_i(\mathbf{X}\mathbf{D}\mathbf{X}^T)\mathbf{v}_i \longrightarrow (\mathbf{X}\mathbf{L}\mathbf{X}^T)\mathbf{V} = (\mathbf{X}\mathbf{D}\mathbf{X}^T)\mathbf{V}\Lambda$ where Λ is a diagonal matrix formed by the eigenvalues λ_i arranged in the ascending order and $\mathbf{V}$ is a matrix whose columns are the eigenvectors ordered in the ascending order of the corresponding eigenvalues.
– Output: Linear transform $\mathbf{W} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_p] \in \mathbb{R}^{d \times p}$

Figure 2.8: Algorithm LsLPP.

2.3 Proposed framework

2.3.1 Overview of the proposed approach

The machine learning pipeline used in the proposed work is summarized in Figure 2.9. The remaining of the section will describe each module. The first module is a classic image preprocessing which involves a feature extraction technique. Thus, the input data can be any descriptor extracted from the raw images (e.g., Local Binary Patterns (LBP), Gabor images, etc.). The second module is a simple dimensionality reduction that can be achieved by PCA. It is a pure unsupervised technique. The third module is given by the new method Sp-LsLPP that is based on a new method for affinity matrix estimation. The latter estimation relies on ℓ_1 coding (sparse representation) that naturally incorporates the pose labels. The fourth module enforces the discrimination among the classes (poses). The final module provides a regression on the projected data in order to predict the pose.

2.3.2 Preprocessing

In this step, image processing techniques are used to prepare the images. During this step, face alignment and cropping might be performed in order to eliminate from the input image, as much as possible, the information that is irrelevant to the pose problem (background). This operation aims at making the pose estimation more robust. The obtained image is then normalized and eventually reshaped as a vector that contains its pixels or the elements of its descriptor. A typical normalization scheme is given by the zero-mean unit-variance scheme. Thus, at the end of the preprocessing step, face images are represented by high dimensional normalized vectors. A typical pipeline of sub-processes can be as follows: $\mathbf{I} \longrightarrow \mathbf{z} \longrightarrow \mathbf{x}$. Here, $\mathbf{I}$ denotes the cropped 2D face image, $\mathbf{z}$ is the vectorized form of $\mathbf{I}$, and $\mathbf{x} = \frac{\mathbf{z}-m_z}{\sigma_z}$ is the normalized $\mathbf{z}$ where m_z is the mean of the $\mathbf{z}$ elements and σ_z is their standard deviation.

2.3.3 Dimensionality reduction

When the feature vector (image or descriptor) has high dimensions and is suspected to contain notoriously redundant data, it must be converted into another representation (possibly with lower dimension) that eliminates the redundancy and helps making the estimation more robust to noise. In

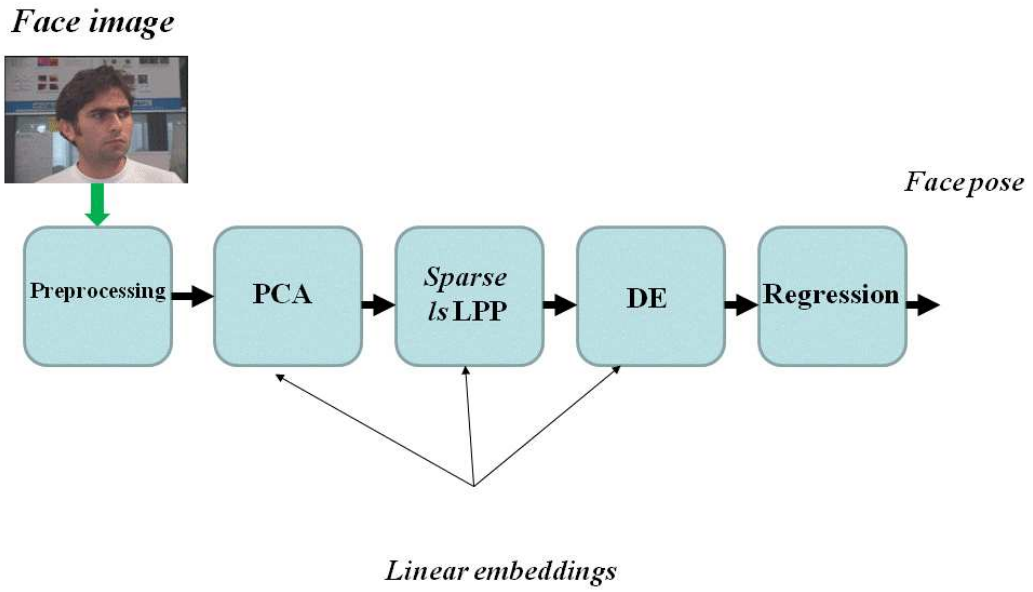


Figure 2.9: The proposed machine learning pipeline. The contributions are the modules: Sparse LsLPP and Discriminant Embedding (DE).

the pose estimation problem, it is likely that the largest variance that exists in the data is due to pose variation. Based on this, PCA [102] is used in the proposed work to find the appropriate space that keeps the information related to the pose and eliminates the redundant information. Following the above notations and assuming that the PCA transform is given by the matrix $\mathbf{W}_{PCA}$ and that the data mean is $\overline{\mathbf{x}}$, then the PCA projection is given by $\mathbf{x}_{PCA} = \mathbf{W}_{PCA}^{T}(\mathbf{x} - \overline{\mathbf{x}})$.

2.3.4 Sparse Label sensitive Locality Preserving Projections (Sp-LsLPP)

While the LsLPP method can integrate the concepts of locality and label sensitivity, it has four parameters. Two parameters are related to feature similarity (i.e., k_1 and t). These are very often hard to fix in advance. In order to release the use of these parameters, and therefore to reduce the total number of parameters that need to be tuned in the LsLPP algorithm and to model the relationship between the samples in a better way, we propose a new method for building the affinity matrix which utilizes sparse coding. The main difference of the proposed approach with the LsLPP method of [21] is that the construction of the graph similarity matrix $\mathbf{B}$ will be carried out via a weighted sparse coding that integrates label sensitivity. Sparse representation is, in general, a representation of images that is the most compact in terms of linear combination of atoms taken form an over-complete dictionary [103]. A dictionary is formed by many images. Sparse representation means that only some images will participate in the reconstruction process while the rest of the images will have zero contribution. This coding of images is done by minimizing the following objective function:

$$min_{\mathbf{b}}(\|\mathbf{y} - \mathbf{X}\,\mathbf{b}\|_2^2 + \lambda\|\mathbf{b}\|_1) \tag{2.1}$$

where $\mathbf{y}$ is the image to be decomposed, $\mathbf{X}$ is the matrix of bases forming the dictionary (in the addressed case it will be the data samples themselves as it will be seen in Algorithm 2.10) and $\mathbf{b} = (b_1, b_2, \ldots, b_N)$ is a column vector formed by the reconstruction coefficients. $\|\mathbf{b}\|_1$ denotes the ℓ_1-norm of $\mathbf{b}$. λ is a scalar regularization parameter that balances the tradeoff between reconstruction error and sparsity. Sparse coding is a well-known tool that provides coefficients that respect feature similarities and closeness.

The proposed Sp-LsLPP algorithm is summarized in Algorithm 2.10. Unlike the formulation in (2.1) which does not integrate label sensitivity, a weighted sparse coding (See Eq. (2.2)) is used to construct the affinity matrix graph where weights are depending on the labels. Each training sample is constructed from the remaining training samples using Eq. (2.2). Thus, the optimization problem is invoked N times. The optimization problem in (2.2) can be solved by many existing algorithms. In the current work, the off-the-shelf code from SLEP package [104] is used.

The absolute values of the reconstruction coefficients present the weights on the edges of the graph since they give an indication of the relationship that

exists between the samples. It should be noticed that the sparse coding is achieved in a way that respects the label-sensitive concept of poses. Indeed, in Eq. (2.2), the label sensitivity concept is enforced by the individual weights represented by the diagonal matrix $\mathbf{P}$. Each weight $P(j,j)$ is a function of the distance between the label (pose) y_j and the label (pose) y_i (where i is the index of the image to be coded). As the distance between label y_j and label y_i increases, the coefficient b_j is forced to be as small as possible (hence no real connection between the image to be coded and the image j).

Sp-LsLPP

Input: Training set represented by the data matrix $\mathbf{X} = \{\mathbf{x}_i \in \mathbb{R}^d\}_{i=1}^N$, labels $\mathbf{y} = \{y_i\}_{i=1}^N$ representing the real-valued pose (angle) of each sample, parameters σ, and λ.
Output: A linear transform matrix $\mathbf{W}$

- For each sample $\mathbf{x}_i$:
  - Compute the diagonal matrix $\mathbf{P}$ such that $P(j,j) = 1 - exp(-(\frac{y(i)-y(j)}{\sigma})^2)$, $j = 1, ..., N$.
  - Estimate the coding vector $\mathbf{b}$ using the following weighted sparse optimization problem:

$$\min_{\mathbf{b}}(\|\mathbf{X}'\,\mathbf{b} - \mathbf{x}_i\|^2 + \lambda\,\|\mathbf{P}\,\mathbf{b}\|_1) \tag{2.2}$$

    where $\mathbf{X}'$ is a matrix formed by the remaining $N-1$ training samples. By introducing the auxiliary vector $\mathbf{a} = \mathbf{P}\,\mathbf{b}$, (2.2) becomes a classic sparse coding problem $\min_{\mathbf{a}}(\|\mathbf{X}'\,\mathbf{P}^{-1}\,\mathbf{a} - \mathbf{x}_i\|^2 + \lambda\,\|\mathbf{a}\|_1)$
  - The i^{th} row of $\mathbf{B}$ is given by $\mathbf{B}(i,:) = (\mathbf{b}_i)^T = (\mathbf{P}^{-1}\mathbf{a})^T$
- Make the affinity matrix symmetric, i.e., $\mathbf{B} \leftarrow |\mathbf{B}| + |\mathbf{B}|^T$
- Compute the Laplacian $\mathbf{L} = \mathbf{D} - \mathbf{B}$ where $\mathbf{D}$ is a diagonal matrix with $D_{ii} = \sum_j B_{ij}$
- Solve the following generalized eigenvector problem: $(\mathbf{X}\mathbf{L}\mathbf{X}^T)\mathbf{v}_i = \lambda_i(\mathbf{X}\mathbf{D}\mathbf{X}^T)\mathbf{v}_i \longrightarrow (\mathbf{X}\mathbf{L}\mathbf{X}^T)\mathbf{V} = (\mathbf{X}\mathbf{D}\mathbf{X}^T)\mathbf{V}\Lambda$
- Output: Linear transform $\mathbf{W} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_p] \in \mathbb{R}^{d\times p}$

Figure 2.10: The proposed Sp-LsLPP.

The generalized eigenvector problem is due, like in the case of LPP, to

the minimization of the criterion: $\sum_{i,j} \|\mathbf{z}_i - \mathbf{z}_j\|^2 B_{ij}$ where $\mathbf{z}_i$ and $\mathbf{z}_j$ are the projections of the i^{th} and j^{th} images. B_{ij} is the weight of the edge that connects the two images. It is obvious that the proposed method has the benefit that it needs only two parameters that are very easy to tune, λ and σ. λ is the sparsity parameter and must be chosen small to prevent the $\mathbf{B}$ matrix to contain lots of zeros which does not describe well the relationship between the samples. Thus, the proposed Sp-LsLPP has released the use of the neighborhood size parameter k_1, the threshold ε, and the scale parameter t.

2.3.5 Discriminant Embedding

LsLPP or Sp-LsLPP have taken into account the order of the continuous labels (poses), and introduced locality preserving projections that exploit both pose orders and feature similarities. However, both methods do not use any explicit discrimination for the images having different labels. In order to tackle that, an additional manifold linear technique (cascaded with the output of Sp-LsLPP) is applied. This technique is inspired by the framework of Linear Discriminant Embedding (LDE) [105]. The objective of LDE is to estimate a linear mapping that simultaneously maximizes the local margin between heterogeneous samples and pushes the homogeneous samples closer to each other. Obviously, a direct application of LDE framework is not feasible since the concept of discrete classes does not exist for the problem of continuous pose estimation. However, in a similar way that label sensitivity is defined, the homogeneous images (w.r.t. a given image) can be defined by all images having a pose close to that image, and the heterogeneous images are the remaining images. In this way, the within-class and between class graphs needed for the LDE technique are easily set.

2.3.6 Regression

The final step in the approach will be the estimation of the pose value given the embedded face image. Regression techniques model the dependency or the relationship between a scalar dependent variable (e.g., the yaw angle) and one or more explanatory variables (in the proposed work, these variables will be the generated projected data). In the proposed work, Partial Least Square Regression (PLSR) [54] is used to perform regression. In particular, the Kernel PLS [106] is used. This choice is motivated by the fact

that non-linear PLS are well suited to the problem at hand. Two kinds of regression were tested: the global regression and the local regression. The global regression techniques are the commonly known regression algorithms that take all the data available in the training set into account when training and calculating the parameters of the regression model. On the other hand, local regression techniques look for the k-nearest neighbors and train the regression model on these neighbors only (e.g., KNN-PLSR [107]). In the sequel, only the results obtained with local regressions are presented, since they gave better performances than the global ones.

2.4 Performance evaluation

In order to test the efficiency of the proposed framework, several experiments are performed on three face datasets using different data embedding techniques.

2.4.1 Experimental setup

The experiments are performed on three different benchmark datasets in which only the yaw angle varies. The first dataset is the FacePix database [1].

An example of this set is shown in Figure 2.11. All the face images are 128 pixels wide and 128 pixels high. These images are normalized, such that the eyes are on the 57th row of pixels from the top, and the mouth is centered on the 87th row of pixels. FacePix is a face image database created at the Center for Cognitive Ubiquitous Computing (CUbiC) at Arizona State University, and made available free of charge to the worldwide research community. In the FacePix database, called FacePix(30), there are 181 face images for each of 30 people where each image corresponds to a rotational interval of 1 degree, across a spectrum of 180 degrees. These images were captured with a moving video camera, using two stationary diffuse light sources that simulate ambient light. The face images in this set contain very little shadowing. Pose angle variations vary across a range from +90 degrees to -90 degrees, where +90 degrees represents a left profile view, 0 degrees represents a frontal view, and -90 degrees represents a right profile view.

The second dataset is the Taiwan dataset [2] which contains images of 90

1. https://cubic.asu.edu/content/facepix-database
2. http://bml.ym.edu.tw/bmlab/

persons taken at a 5-degree step. Acquisition conditions are similar to those of FacePix dataset. Figure 2.12 shows some samples from this database. While the faces in FacePix dataset are already aligned, those of Taiwan dataset are not. In order to align them, a similar procedure used for aligning faces in FacePix dataset is adopted. That is achieved by detecting the eyes in every image.

The third dataset is the Columbia dataset [3]. It is a database made originally for evaluating the eye gaze estimation. This database also contains images depicting face poses of different persons. This database contains high-resolution images of 56 different people (32 male, 24 female) and each image has a resolution of 5184 x 3456 pixels. 21 of the subjects were Asian, 19 were White, 8 were South Asian, 7 were Black, and 4 were Hispanic or Latino. The subjects ranged from 18 to 36 years of age, and 21 of them wore prescription glasses. For each subject, the database contains images for five horizontal face poses (0°, ±15°, ±30°), i.e, only five yaw angles. Since this database does not have very much variation in the face pose like the previous databases, the performance evaluation of the face pose will adopt classification of the yaw angles. Figure 2.13 shows samples from this database.

Two kinds of image descriptors are tested: (i) raw images; and (ii) LBP histograms (one block and multiblock) [108, 109].

For LBP descriptors, we use a radius of one pixel and eight neighboring pixels. This choice of LBP parameters can be considered as a good trade-off between detecting fine local features and the size of the feature vector. For the multi-block LBP, 25 blocks are used (See Figure 2.14(b)). Each region/block is represented by its LBP histogram and the whole image descriptor is the concatenation of these 25 histograms.

2.4.2 Method comparison

The tested data embedding methods are defined and described in Table 2.1. Note that all methods are linear except for the Supervised Laplacian Eigenmaps (S-LE) which is non-linear. For instance, the combination Sp-LsLPP+DE (proposed method) means that the training phase uses the training data samples in order to recover three linear transforms: the PCA transform denoted by $\mathbf{W}_{PCA}$, the sparse Ls-LPP transform denoted by $\mathbf{W}_{SP}$ and, the DE transform denoted by $\mathbf{W}_{DE}$. Thus, the final embed-

3. www.cs.columbia.edu/CAVE/databases/columbia_gaze/

ding of an unseen face image $\mathbf{x}$ using the combination Sp-LsLPP+DE will be $\mathbf{f}(\mathbf{x}) = \mathbf{W}_{DE}^T \mathbf{W}_{SP}^T \mathbf{W}_{PCA}^T \mathbf{x}$. In this formula, $\mathbf{x}$ is a vector that contains either the vectorized raw image or the LBP descriptor of the image. Finally, the yaw angle is given by: $Angle(\mathbf{x}) = PLSR(\mathbf{f}(\mathbf{x}))$. We also use the supervised non-linear Laplacian Eigenmaps (S-LE) in [110]. Since this method is non-linear, the test images are projected using the out-of-sample method described in [111].

The parameters of each embedding technique are set to a subset of values that are used and only the best performance will be reported. The LPP and LsLPP methods have the Gaussian similarity scale, t, and the neighborhood size k_1. t is set to the average of squared distances among the samples in the training set. The neighborhood size $k_1 \in \{5, 10, 15, 20, 25, 30, 35, 40\}$. The LsLPP technique has two additional parameters ε and σ. $\varepsilon \in \{4, 6, 8\}$ and σ is set to 100. The regularization parameter of the sparse LsLPP is set to 0.01. The threshold used by the DE technique considers a pair of images as homogeneous if their label difference is below 10 degrees, and it considers the pair as heterogeneous if this difference exceeds 20 degrees. The PCA method retained 53 components for raw images of size 1024 and 540 components for the block based LBP descriptor whose size is 6400. The dimension of the final projected data is set to 45 features. The γ parameter of the S-LE method is set to 0.1. The values of neighborhood used by PLS regressor are $\{10, 20, 30, 40, 50\}$. The number of independent latent variables in the PLSR technique was set to one and three.

Figure 2.11: A subset of images belonging to seven persons available in the FacePix database. The images are chosen at a step of 10 degrees.

Figure 2.12: Face images that belong to seven persons available in the Taiwan database. The images are chosen at a step of 10 degrees.

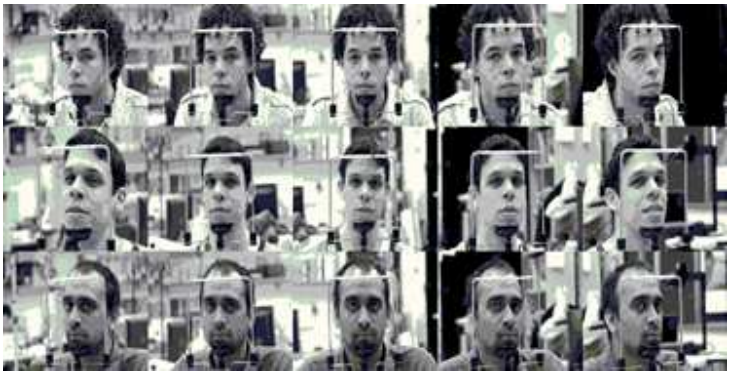


Figure 2.13: Face images from Columbia face dataset.

For the evaluation, we have adopted the person-independent pose estimation protocol. For each dataset and for each embedding method, the set of images is split into a training part and a test part. Five training/test splits are considered. In each split, 25 random persons (with all their images) are used for training. The images of the remaining persons are used for testing.

The method comparisons for the FacePix dataset are reported in Table 2.2. In this table, the Mean Average Error (MAE) and its standard devia-

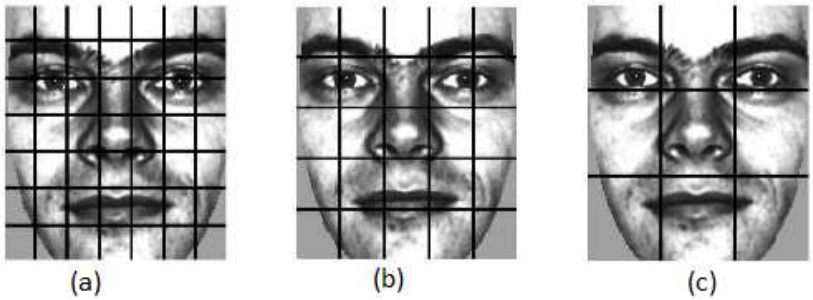


Figure 2.14: Three different partitions of the same face image: 7×7 blocks; (b) 5×5 blocks, (c) 3×3 blocks.

tion are depicted. The best performances are highlighted in bold. It can be observed that LsLPP, Sp-LsLPP, and Sp-LsLPP+DE methods give the best results. The superiority is held by the Sp-LsLPP+DE method despite the small improvement. It can be seen that the LsLPP method and the proposed method Sp-LsLPP give very close results and they both outperform the classic LPP. Since Sp-LsLPP method needs two parameters to be tuned while Ls-LPP needs four, it is obvious that the proposed Sp-LsLPP is more flexible. Indeed, by considering that the regularization parameters are always set to a small positive number, it turns out that the proposed Sp-LsLPP method has only one single parameter which is the Gaussian scale used in the weighted ℓ_1 regularization. SSE method and the LsLPP+SSE method have given the same performance. This is explained by the fact that the synchronization of samples using simplexes in SSE give the same embedding whether the data are projected onto LsLPP or not.

With regards to the descriptors used, it can be observed that the block-based LBP descriptors have given better accuracy than the raw images. This can be explained by the fact that the local LBP histograms can characterize well the local textures of faces and thus give better relation to the real face pose. One can notice that the average error has decreased from 4.41 degrees (Sp-LsLPP) to 2.94 degrees (Sp-LsLPP + DE), suggesting that the DE stage can be useful in some cases. The non-linear embedding method (S-LE) has only outperformed the methods in only one split (split number 5). However, on average, it could not outperform the linear methods.

The method comparisons for the Taiwan dataset are reported in Table

2.3. For the Taiwan dataset, similar observations to those associated with the FacePix dataset can be drawn. One can also observe that the non-linear method was the second best after the proposed method Sp-LsLPP+DE. It can be observed that the accuracy obtained with the Taiwan dataset is slightly worse than that obtained with the FacePix dataset. This is due to the fact that the training images in Taiwan dataset correspond to a step of five degrees instead of one degree.

A method comparison for the Columbia datasets is illustrated in Table 2.4. In this table, the percentage of correct classification of the yaw angle (among five classes) is shown. The classifier used is given by the Support Vector Machines (SVM) with Radial Basis Function. The upper part corresponds to a partition of 30% training and 70% testing. The lower part to a partition of 90% training and 10% testing. As can be seen, for large training sets the label sensitive methods can outperform the classic transform. The proposed sparse method has the obvious advantage of having less parameters than the other methods.

Table 2.1: Definition of the cascaded manifold learning techniques used in the experiments.

Method	Description
S-LE	PCA followed by Supervised Laplacian Eigenmaps
LPP	PCA followed by Locality Preserving Projections
LsLPP	PCA followed by Label sensitive Locality Preserving Projections
Sp-LsLPP	PCA followed by Sparse LsLPP
SSE	PCA followed by Synchronized Submanifold Embedding
LsLPP+SSE	PCA followed by LsLPP followed by Synchronized Submanifold Embedding
Sp-LsLPP+DE	PCA followed by Sparse LsLPP followed by Discriminant Embedding

Figure 2.4.2 illustrates the obtained MAE as a function of the real yaw angle associated with the FacePix dataset. The embedding method was the 25 block based LBP with the Sp-LsLPP embedding. As can be seen the profile views ranging from 80° to 90° have a relatively large MAE.

Table 2.2: Mean Average Error (MAE) in degrees and Standard deviation obtained for FacePix dataset and for different embedding methods. There are 5 training/test splits. In each splits, 25 random persons (with all their images) are used for training. The images of the remaining 5 persons are used for test. The regression method used is the local PLS.

Method (Raw images)	Split1	Split2	Split3	Split4	Split5	Mean
S-LE	$4.50_{\pm 4.8}$	$4.50_{\pm 4.8}$	$4.33_{\pm 4.8}$	$4.80_{\pm 6.8}$	$\mathbf{1.27}_{\pm 6.5}$	$3.88_{\pm 5.5}$
LPP	$4.93_{\pm 5.1}$	$4.80_{\pm 4.9}$	$6.80_{\pm 7.1}$	$4.87_{\pm 5.2}$	$5.82_{\pm 6.2}$	$5.44_{\pm 5.7}$
LsLPP	$3.00_{\pm 2.9}$	$\mathbf{3.9}_{\pm 4.6}$	$3.34_{\pm 3.0}$	$3.84_{\pm 5.2}$	$4.16_{\pm 4.7}$	$3.65_{\pm 4.1}$
Sp-LsLPP	$\mathbf{2.59}_{\pm 3.1}$	$4.03_{\pm 4.9}$	$3.23_{\pm 3.2}$	$4.27_{\pm 5.8}$	$4.89_{\pm 6.1}$	$3.80_{\pm 4.6}$
SSE	$4.45_{\pm 4.1}$	$4.77_{\pm 5.8}$	$4.077_{\pm 3.9}$	$4.28_{\pm 5.6}$	$4.82_{\pm 6.9}$	$4.47_{\pm 5.3}$
LsLPP+SSE	$4.45_{\pm 4.1}$	$4.77_{\pm 5.8}$	$4.077_{\pm 3.9}$	$4.28_{\pm 5.6}$	$4.82_{\pm 6.9}$	$4.47_{\pm 5.3}$
Sp-LsLPP + DE	$2.84_{\pm 2.9}$	$4.09_{\pm 5.0}$	$\mathbf{3.16}_{\pm 2.9}$	$\mathbf{3.4}_{\pm 5.00}$	$3.85_{\pm 4.6}$	$\mathbf{3.48}_{\pm 4.1}$

Method (LBP descriptors)	Split1	Split2	Split3	Split4	Split5	Mean
One block	$4.93_{\pm 7.7}$	$9.00_{\pm 16.4}$	$5.40_{\pm 8.2}$	$5.81_{\pm 9.3}$	$6.01_{\pm 9.0}$	$6.23_{\pm 10.1}$
25 blocks	$4.17_{\pm 5.5}$	$4.53_{\pm 5.9}$	$4.77_{\pm 6.4}$	$4.42_{\pm 6.1}$	$5.08_{\pm 7.2}$	$4.59_{\pm 6.2}$
25 b: Sp-LsLPP	$3.48_{\pm 4.3}$	$3.94_{\pm 4.8}$	$6.09_{\pm 8.6}$	$4.58_{\pm 6.5}$	$3.96_{\pm 5.2}$	$4.41_{\pm 5.9}$
25 b: Sp-LsLPP + DE	$\mathbf{2.44}_{\pm 3.0}$	$\mathbf{2.98}_{\pm 3.8}$	$\mathbf{3.32}_{\pm 4.3}$	$\mathbf{3.03}_{\pm 3.9}$	$\mathbf{2.95}_{\pm 4.1}$	$\mathbf{2.94}_{\pm 3.8}$

Table 2.3: Mean Average Error (MAE) in degrees and Standard deviation obtained for Taiwan dataset and for different embedding methods. There are 5 training/test splits. In each split, 25 random persons (with all their images) are used for training. The images of the remaining 5 persons are used for test. The regression method used is the local PLS.

Method (Raw images)	Split1	Split2	Split3	Split4	Split5	Mean
S-LE	$7.34_{\pm 9.7}$	$\mathbf{4.65}_{\pm 4.8}$	$\mathbf{4.55}_{\pm 4.8}$	$5.11_{\pm 7.0}$	$\mathbf{4.57}_{\pm 4.9}$	$5.24_{\pm 6.2}$
LPP	$6.14_{\pm 4.8}$	$5.82_{\pm 4.7}$	$5.50_{\pm 4.7}$	$5.43_{\pm 4.6}$	$5.40_{\pm 4.6}$	$5.65_{\pm 5.7}$
LsLPP	$5.66_{\pm 4.8}$	$5.38_{\pm 4.3}$	$5.07_{\pm 4.3}$	$\mathbf{5.05}_{\pm 4.2}$	$4.84_{\pm 4.1}$	$5.20_{\pm 4.3}$
Sp-LsLPP	$5.72_{\pm 7.4}$	$5.07_{\pm 6.6}$	$5.93_{\pm 7.5}$	$6.01_{\pm 7.8}$	$5.37_{\pm 6.9}$	$5.62_{\pm 7.2}$
SSE	$5.63_{\pm 7.2}$	$5.35_{\pm 6.7}$	$5.50_{\pm 6.9}$	$5.39_{\pm 6.8}$	$5.47_{\pm 7.0}$	$5.46_{\pm 6.9}$
LsLPP+SSE	$5.63_{\pm 7.2}$	$5.35_{\pm 6.7}$	$5.50_{\pm 6.9}$	$5.39_{\pm 6.8}$	$5.47_{\pm 7.0}$	$5.46_{\pm 6.9}$
Sp-LsLPP + DE	$\mathbf{5.48}_{\pm 7.1}$	$5.34_{\pm 6.7}$	$4.96_{\pm 6.1}$	$5.12_{\pm 6.6}$	$4.97_{\pm 6.4}$	$\mathbf{5.17}_{\pm 6.5}$

Method (LBP descriptors)	Split1	Split2	Split3	Split4	Split5	Mean
One block	$10.89_{\pm 14.8}$	$12.18_{\pm 18.0}$	$11.42_{\pm 15.5}$	$11.46_{\pm 18.5}$	$10.79_{\pm 15.2}$	$11.34_{\pm 16.4}$
25 blocks	$6.89_{\pm 9.1}$	$7.16_{\pm 9.5}$	$7.10_{\pm 8.7}$	$7.37_{\pm 9.8}$	$7.13_{\pm 9.6}$	$7.13_{\pm 9.3}$
25 b: Sp-LsLPP	$6.85_{\pm 8.6}$	$6.12_{\pm 7.8}$	$6.77_{\pm 7.8}$	$6.96_{\pm 9.1}$	$7.12_{\pm 9.1}$	$6.76_{\pm 8.5}$
25 b: Sp-LsLPP + DE	$\mathbf{5.46}_{\pm 7.2}$	$\mathbf{4.73}_{\pm 6.3}$	$\mathbf{4.74}_{\pm 5.8}$	$\mathbf{4.93}_{\pm 6.5}$	$\mathbf{4.96}_{\pm 6.5}$	$\mathbf{4.96}_{\pm 6.5}$

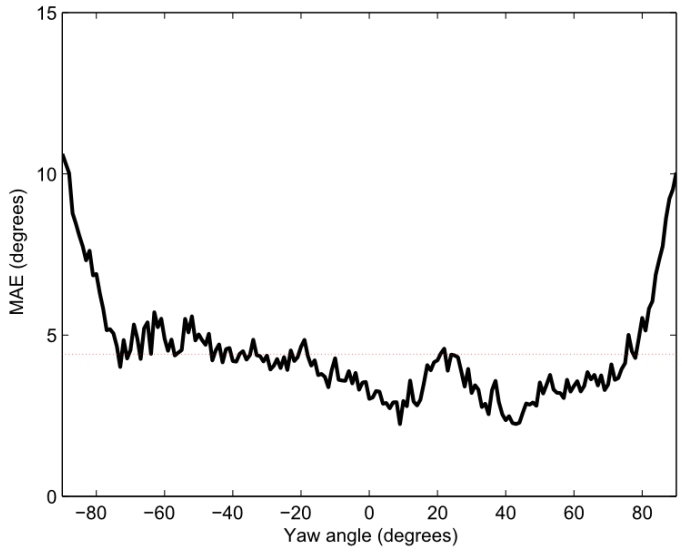


Figure 2.15: MAE as a function of the real yaw angle associated with the FacePix dataset. The embedding method was the 25 block based LBP with the Sp-LsLPP.

Table 2.4: Percentage of correct classification of the yaw angle for Columbia dataset and for different sets formed by 90% training and 10% testing. There are 5 training/test splits. In each split, 50 random persons (with all their images) are used for training. The images of the remaining 6 persons are used for test. The classifier used after embedding is the SVM with Radial Basis Kernel.

30% training and 70% testing						
Method (Raw images)	Split1	Split2	Split3	Split4	Split5	Mean
LPP	89.7	**89.7**	**89.2**	**89.7**	**89.7**	**89.6**
LsLPP	**90.2**	88.2	88.2	91.8	87.2	89.1
Sp-LsLPP	88.7	86.7	86.1	89.7	84.6	87.2

90% training and 10% testing						
Method (Raw images)	Split1	Split2	Split3	Split4	Split5	Mean
LPP	**96.0**	92.0	96.0	92.0	88.0	92.8
LsLPP	92.0	92.0	96.0	96.0	96.0	**94.4**
Sp-LsLPP	92.0	**92.0**	**96.0**	**96.0**	**96.0**	**94.4**

2.5 Conclusion

In this chapter, a new machine learning framework was used to estimate the pose of human faces in images. More precisely, a new embedding algorithm based on a sparse representation was proposed. The resulting technique is called Sparse Label sensitive Locality Preserving Projections. For enhancing the discrimination between poses, the projected data obtained by the Sparse Label Sensitive Locality Preserving Projections are fed to a discriminant embedding that exploits the continuous labels. The obtained results show that the sparse representation with label similarity is an efficient method for data embedding that outperforms state-of-the-art embedding algorithms used for such applications, in the sense that it is easy to adapt to different datasets and that it only needs two parameters to tune.

Chapter 3

Person Re-identification through hand-crafted features

Abstract

In this chapter, we describe our proposed framework for appearance based person re-identification. We consider the single-shot scenario where each person is associated with only one image in each of the two cameras. First, we review some typical descriptors based on texture and color. Then we present our Color Categorization method and the feature extraction technique adopted to increase the discriminability. Finally, we describe our re-identification process based on a prototype formation technique before presenting the experimental results.

Contents

3.1 Appearance-based Person Re-identification: A review

Person re-identification is the process of identifying the same person from images taken from different cameras. Most of the existing works generate a ranked vector of all the persons in a gallery set (images of labeled persons) based on their resemblance with the probe image (unlabeled person). The highest similarity score in the ranked vector will lead us to a specific label for the probe image. Existing techniques on person re-identification usually fall into two categories. The first category known as 'single-shot approach' [7], focuses on connecting pairs of images, each containing one instance of an individual. The second category uses multiple images of the same person as training data and considers short sequences for testing (usually obtained via tracking). It is known as 'multiple-shot approach' [8]. Most of the existing approaches are based on appearance similarity where they aim to find a good representation to establish correspondences between images.

Typical descriptors, like texture and color extracted from clothing, have been widely used. Popular descriptors like SIFT (Scale Invariant Feature Transform), which consists of computing a histogram of the gradient orientation distribution in the region around a detected interest point have been used in [112]. Other descriptors have been used like Texture filters [113], color and shape features [114] and Principal axis [115]. The authors in [116] seek the most distinctive representation of an individual, in which they collect two images of each subject, one for the whole body and the other can be a zoomed body part (head, torso or legs). We can find in [117] a review of descriptors based on appearance.

Algorithms relying on Multiple Part Multiple Component (MPMC) were used to design region based descriptors, by dividing the body into many parts to take into consideration the non-rigid nature of the human body [118]. In [119], the human body was subdivided into head, torso and legs. Each part was represented by a Gaussian color histograms in addition to pyramid of histograms of oriented gradients and texture description using Haralick features. Authors in [120] proposed to use the Haar-like features and DCD (Dominant Color Descriptor), where an MPEG7 descriptor is used to characterize each part of the subdivided body. The same authors pro-

posed to use HOG (Histogram of Oriented Gradient) descriptor to describe an appearance model based on covariance regions extracted from the body [121].

Color is one of the most important features for computer vision systems, it has been found as the most important factor in many person re-identification studies. In general, the color value can be transformed conveniently from RGB to HSV [122]. The reason is that it provides an intuitive representation and approximates the way in which human perceive color. Color histogram has been successfully applied to the person re-identification problem like in [123]. Color histograms in 3 color spaces (RGB, HSV and YCrCb) have been used in [124]. In order to maintain vertical color shape in appearances, the work of [125] integrated spatial data into color characteristics. To handle changes in illumination, they used color normalization and color rank features. In [126], color and texture were merged and PLS (Partial Least Squares) is used to reduce the dimensionality. The authors in [127] demonstrated that there is a trade-off between illumination invariance and discriminative power. To counterbalance between illumination invariance and the discriminative power, the authors in [128] made a fusion of color models which enhanced the discrimination compared to standard weighting schemes.

After feature extraction, these methods normally choose a standard distance measurement to calculate the similarity between images, e.g, Bhattacharyya distance [113], L1-Norm [129] or L2-Norm [115]. In [130], the authors proposed to learn a metric from pairs of samples from different cameras using discriminative Mahalanobis metric learning. In [131], Zheng et al. formulated the person re-identification as a relative distance comparison problem with a Mahalanobis distance metric. All these techniques suffer from some difficulties to some extent because of the low resolution of images, camera settings and lighting conditions.

3.2 Color Catgorization

The first stage of our re-identification process is the Color Categorization. In this section, we first present the motivation for the Color Categorization procedure. Then, we briefly review the Probabilistic Latent Semantic Analysis (PLSA) method. Finally, we introduce our proposed method.

3.2.1 Motivation

Color is one of the most expressive and powerful clue for re-identification. However, color description is still very complicated and it has been the subject of many studies. One of the most influential works in color description is the study of Berlin and Kay [132] on basic color terms. Humans differentiate between two persons wearing the same colors based on distinct shoes, hair styles, bags. In computer vision, if large changes occur in lighting or in background, a person's appearance can change significantly between cameras. These changes can make different persons appear more alike than the same person across different camera views, which increases the difficulty of finding correct association. Humans use color names to describe the colors of objects such as 'black', 'white' and 'blue'. Though we use them routinely without effort, this task is not trivial. For example, in the RGB color Space, [1 0 0] will be identified as red, [1 0.1 0.1] will also be identified as red, which means there are boundaries to each set of color. The fuzziness of these boundaries makes this problem difficult. In Figure 3.1 the test person is wearing red. We can see that many persons who are not wearing red are ranked higher than the true person. A simple question may arise here: why for example the person wearing black is ranked higher than 3 persons wearing red? To overcome this weak point, our proposal (detailed in the sequel) is to put each image in a category depending on its color. Thus, a test person will be compared first to the persons from its color category, then to all the remaining.

3.2.2 A brief review of PLSA

The Probabilistic Latent Semantic Analysis (PLSA) investigates the relation between a set of documents and the terms they contain to achieve a set of topics [133]. Weijer et al. adjusted PLSA in two ways [134] [135]: They turned PLSA into a supervised multiclass learning approach by directly linking the topics with the class labels. And they proposed a background class shared across topics reflecting that images generally have a foreground object on a background. In [134], A PLSA model is learned on the Google images, in the form of $32 * 32 * 32$ lookup table which allows to map pixel values to color names. Eleven basic color terms of the English language are used in the PLSA approach: black, blue, brown, grey, green, orange, pink, purple, red, white, and yellow. After learning, given a pixel, the model provides a probability value for each color category: $P(n_i|f(x))$. Where n_i is the i^{th}

Figure 3.1: Image (a) is the Test Image, (b) are the images of the gallery set with sorted order. The feature vector used is the RGB channel values and its gradients. For the similarity score we calculated the distance between the covariance matrices.

color name and f(x) is the color value of the pixel x, and the name assigned is the one with maximum probability.

3.2.3 Color Categorization

In this section we propose a new Color Categorization method, which relies primarily on the PLSA method proposed in [134]. PLSA is efficient if working under the same light conditions, but it fails when working with two different cameras under two different light conditions like in our case. In fact, changes in the illumination affect object colors far beyond the tolerance required for reasonable object recognition. Thus, to increase robustness to illumination changes, we begin by adopting a new color palette. Some colors are visually similar and can be merged to further reduce the number of colors. In our work, grey is considered white. Orange, pink and purple are considered Red. Brown is considered Black. We have tested on various images, and found that adopting 6 colors (Black, White, Red, Green, Blue and Yellow) instead of 11 would be suitable for most cases. One might argue that having a finer quantization may better discern different images, e.g., telling a grey shirt from a white shirt. Unfortunately, finer quantization leads to less reliable color prediction, and can be counter-productive in improving prediction accuracy (a white shirt in the shadow can appear to be

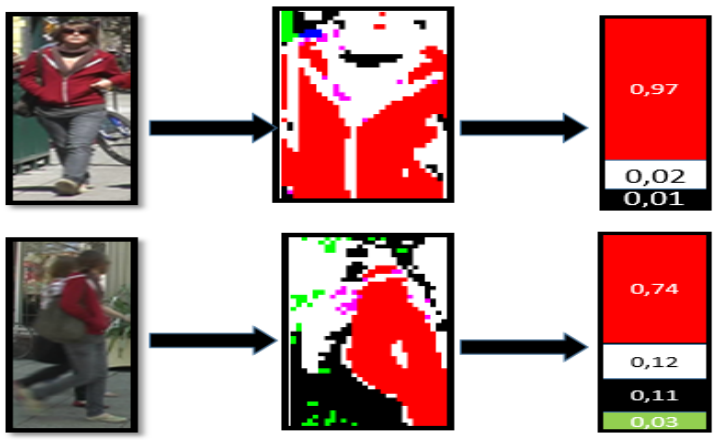


Figure 3.2: Color Categorization procedure, both images are assigned to the red category since the majority of the detected pixels are red. The first image is from camera 1, the second is for the same person from camera 2

grey). After color name annotations based on PLSA, we propose a robust color calibration procedure as follows: for each color name we build our own binary lookup table. A given color can be found at the X,Y,Z coordinates corresponding to the R,G,B normalized values. There are several different public color databases available online, in our work we employ the database defined in [136]. Each of the RGB channels is divided into 52 levels. Pixel coordinates, are treated as the index of the look-up table. We assign the value 0 for all the pixels that do not belong to the color category in question, and 1 otherwise. To predict color membership of a new pixel we interpolate the new triplet of a pixel (scaled to [0,1]) with the lookup table of each of the colors. The results represent how much does it belong to each of the colors (a value from 0 to 1). Differences in lighting conditions between the cameras make the images generated from camera 1 brighter than the ones generated by camera 2. Therefore, we tune our settings to be less sensitive to white color in camera 1 (0.7) than in camera 2 (0.3). For example, in Figure 3.2 first line, the white color is considered to be only 2% from the image but in reality it is much more. Having a probe image, every pixel can be assigned to a color category. Finally, the image is assigned to the color category of the majority of the detected pixels.

3.3 Feature Descriptor

The second stage of our model generates the low-level feature descriptor extracted from raw images. Most of the existing methods are based on extracting color channels individually (for example RGB or HSV) neglecting the relation between each of these color channels. LBP (Local Binary Pattern) has already proved its efficiency in describing image's local information. However, performing LBP on each color channel respectively ignore the relations between the colors. QLBP (Quaternion Local Binary Pattern) takes advantage of both LBP to extract features and quaternions to represent each color pixel such that we can handle all color components at once.

3.3.1 Quaternions

Quaternions were introduced in 1843 by Sir William Rowan Hamilton [137]. A quaternion number is composed of a real part and three-dimensional imaginary part:

$$Q = a + ib + jc + kd \tag{3.1}$$

where a, b, c and d are real numbers, i, j and k are orthogonal imaginary operators which define a hyper-complex space subject to the following restriction:

$$i^2 = j^2 = k^2 = ijk = -1 \tag{3.2}$$

A color pixel at location (n, m) in an RGB image is represented using the imaginary part of the quaternion:

$$Q = R(n, m)i + G(n, m)j + B(n, m)k \tag{3.3}$$

where R, G and B denote the red, green and blue components respectively.

3.3.2 Local Binary Pattern

The Local Binary Patterns (LBP) [138] was originally designed for texture description and it has been widely used in computer vision. It basically assigns a label to every pixel of an image by thresholding a circular neighborhood surrounding the reference pixel and representing the result as a binary number, called the LBP code. The LBP code for a reference pixel c would

be formally defined as follows:

$$LBP_{(M,R)}(c) = \sum_{m=0}^{M-1} s(g(m) - g(c))2^m \tag{3.4}$$

M is the number of sampling values used within a circle neighborhood of radius R, $g(.)$ is the intensity value of a pixel and the function s(x) is expressed as:

$$s(x) = \begin{cases} 1 & if \ \ x \geqslant 0 \\ 0 & if \ \ x < 0 \end{cases} \tag{3.5}$$

An LBP operator generates 2^M binary patterns but some of them contain more information than others when describing image textures. In consequence, an extension to the original operator, called uniform patterns, can be used to reduce the length of the feature vector and implement a simple rotation invariant descriptor. A local binary pattern is called uniform if the binary pattern contains at most two bitwise transitions from 0 to 1 or vice versa when the bit pattern is considered circular. For example, 00010000(2 transitions) is a uniform pattern, 01010100(6 transitions) is not. In the computation of the LBP histogram, the histogram has a separate bin for every uniform pattern, and all non-uniform patterns are assigned to a single bin. Using uniform patterns, the length of the feature vector for a single cell reduces from 256 to 59. In [139], it was observed that the uniform patterns for the operator LBP (8, 1) account for about 90% of all possible patterns in texture images.

3.3.3 Quaternion Local Binary Pattern

The Pseudo Rotation of Quaternion (PRQ) was first introduced in [140]. The right and left pseudo rotation of q by p are defined as:

$$PRQ_r(q,p) = qp \ , \ \ PRQ_l(q,p) = pq \tag{3.6}$$

Let $q = ir + jg + kb$ where r, g, and b denote the red, green, and blue components of a pixel in a color image. Given $p = ix + jy + kz$ as a unit quaternion, the right pseudo rotation of q by p is:

$$PRQ_r(q,p) = -(rx + gy + kb) + i(gz - by) + j(bx - rz) + k(ry - gx) \tag{3.7}$$

The phase difference in this case informs us about potential spatial shifts in images. Since PRQ does not change the modulus, the phase of the PRQ is used to rank the quaternions:

$$\theta = \arctan(-\frac{\sqrt{(gz-by)^2+(bx-rz)^2+(ry-gx)^2}}{rx+gy+bz}) \tag{3.8}$$

Given a reference quaternion p, we first apply the PRQ on both q_i and q_j in order to obtain two pseudo rotated quaternions. Then the ranking function $R(q_i, q_j)$ compute the phase difference of the two pseudo rotated quaternions:

$$R(q_i, q_j) = \theta_{PRQ_{(q_j,p_1)}} - \theta_{PRQ_{(q_i,p_1)}} \tag{3.9}$$

The QLBP description of a pixel x_i centered in a 3 x 3 block S_i is defined as follows:

$$QLBP_{x_i} = \sum_{j=0}^{|Si|-1} s(R(q_i, q_j))2^j \tag{3.10}$$

The QLBP feature descriptor is summarized in Algorithm 1.

3.4 Discriminant Projection

After representing each image by its feature descriptor, we aim to learn a linear subspace to enhance the discriminative capabilities of our representation. This can be achieved by seeking a projection matrix W that finds a new representation y_i for each sample x_i: $y_i = W^T x_i$. Assuming we have two cameras, the feature descriptors representing images from camera c_j, where $j \in \{1, 2\}$, can be denoted as:

$$X^{cj} = \{x_1^{c_j}, x_2^{c_j}, x_3^{c_j}, ..., x_n^{c_j}\}$$

where $x_i^{c_j}$ is the i^{th} feature vector representing the i^{th} image of camera c_j and n is the number of persons. On the one hand, to promote the seperability between different persons, the projection matrix W should maximize the distances between different persons. Let S_{max} be:

$$S_{max} = \sum_{i=1}^{n}\sum_{j=1}^{n} \|W^T x_i^{c1} - W^T x_j^{c2}\|^2 I_1 \tag{3.11}$$

Algorithm 1 Feature descriptor

Input: Image I
N: number of unit quaternions
A set of unit quaternions :$[p_1, p_2, ..., p_N]$
Output: Feature vector f
Procedure:
Find the quaternionic representation Q_I of image I using Eq. (3.3)
For i = 1 to N, do:
Calculate the phases of the rotated quaternionic image $PRQ_r(Q_I, p_i)$.
Extract the QLBP using Eqs. (3.9) and (3.10).
Divide the QLBP image into 8x16 subimages overlapped by half.
Calculate the histogram of each subimage.
Concatenate all histograms to obtain the vector F_i.
end
$f = [F_1, F_2, ..., F_N]$

$$I_1 = \begin{cases} 1 & if \ \ the\ person\ in\ image\ i\ and\ the\ person\ in\ image\ j\ are\ different \\ 0 & else \end{cases}$$

On the other hand, the projection matrix should minimize the distances between two images belonging to the same person. Let S_{min} be:

$$S_{min} = \sum_{i=1}^{n}\sum_{j=1}^{n} \|W^T x_i^{c1} - W^T x_j^{c2}\|^2 I_2 \tag{3.12}$$

$$I_2 = \begin{cases} 1 & if \ \ the\ person\ in\ image\ i\ and\ the\ person\ in\ image\ j\ are\ the\ same \\ 0 & else \end{cases}$$

To achieve this dual problem, we derive the projection matrix as:

$$Arg\max_W(\frac{S_{max}}{S_{min}}) = Arg\max_W(\frac{Trace(W^T S1\, W)}{Trace(W^T S2\, W)}) \tag{3.13}$$

Where:

$$S1 = \sum_{i=1}^{n}\sum_{j=1}^{n}(x_i^{c1} - x_j^{c2})(x_i^{c1} - x_j^{c2})^T I_1 \tag{3.14}$$

$$S2 = \sum_{i=1}^{n}\sum_{j=1}^{n}(x_i^{c1} - x_j^{c2})(x_i^{c1} - x_j^{c2})^T I_2 \tag{3.15}$$

Eq. (3.13) involves a search for a transformation matrix W that maximizes a term: $Trace(W^T S1\, W)$ and at the same time minimizes another: $Trace(W^T S2\, W)$. This is equivalent to maximizing the quotient $\frac{S_{max}}{S_{min}}$. Generally, the trace ratio problem (Eq. 3.13) is often simplified into a more tractable one called the ratio trace problem [141], because the trace ratio problem does not have a closed-form solution:

$$Arg\max_{W}\ Trace(\frac{W^T S1\, W}{W^T S2\, W}) \tag{3.16}$$

This problem can be solved using the generalized Eigen value decomposition method as:

$$S1\, W_i = \lambda_i\, S2\, W_i \tag{3.17}$$

where W_i is the eigenvector corresponding to the i^{th} largest eigenvalue λ_i and it is the i^{th} column vector of the projection matrix W. At this stage, an informative descriptor f(I) of the template I is obtained.

3.5 Re-Identification Process

We divide the database into 3 subsets: Prototype set, Gallery set and Probe set. Given a probe image, we compare it to all the images in the gallery set and then we sort them in similarity order. The division into the three sets is performed as follows. First we randomly choose half of the images of each of the two cameras to be our Prototype set. The remaining images are used for the gallery set (from Camera1) and probe set (from Camera2). Note that the prototype set is also used for estimating the discriminant projection matrix presented in the previous section. The similarity between two images is measured using a kernel function:

$$K : (f(I), f(J)) \rightarrow S_{I,J} \in \mathbb{R}$$

where $f(I)$ is the feature vector representing image I. Let U be a set of the n_t training images containing the prototypes of the Camera2 : $U = \{T_1, T_2, ..., T_{nt}\}$.

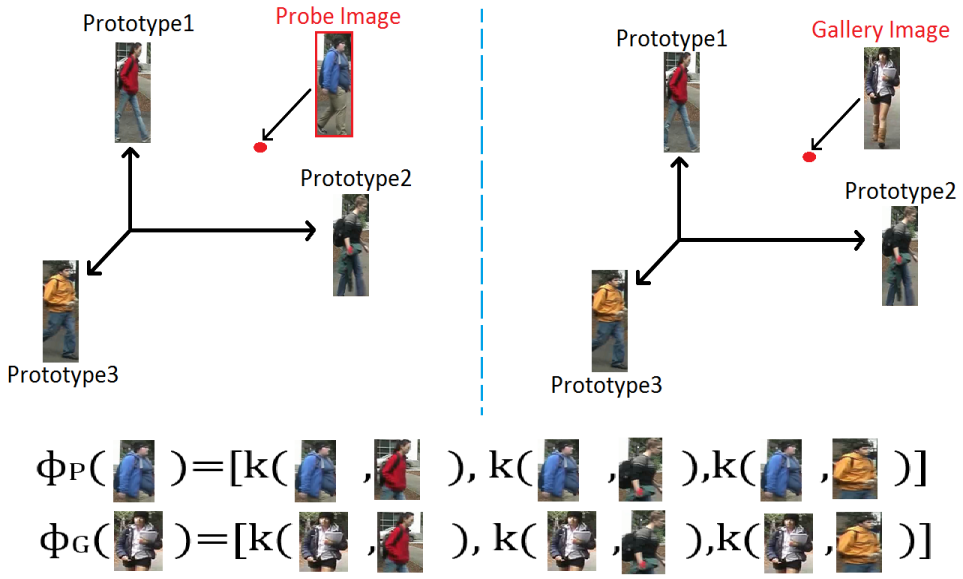


Figure 3.3: Prototype process: Kernel similarities are computed for each image with each of the prototype images. $\phi_P(p)$ and $\phi_G(g)$ are the vectors of similarities between prototype images with a probe image and with a gallery image respectively.

We compute the Kernel similarity function between the prototypes with the probe and between the prototypes with the gallery subjects.

$$K(f(I), f(T_i)) = \frac{< f(I, f(T_i) >}{||f(I)||.||f(T_i)||} \tag{3.18}$$

The cosine kernel was chosen because it is devoid of parameters. Let g represent a gallery image and p some unknown probe image ($g, p \notin U$). The function $\phi_P(p)$ gives a vector containing the similarity between each image T_i in U and the probe image p. And for the gallery set, $\phi_G(g)$ gives a vector containing the similarity between each image T_i in U and the gallery image g (illustrated in Figure 3.3).

$$\phi_P(p) = [K(f(p), f(T_1)), ..., K(f(p), f(T_{nt}))]^T$$

$$\phi_G(g) = [K(f(g), f(T_1)), ..., K(f(g), f(T_{nt}))]^T$$

We finally define $S(p, g)$, the similarity between a probe image p and a gallery image g:

$$S(p, g) = \frac{< \phi_P(p), \phi_G(g) >}{||\phi_P(p)||.||\phi_G(g)||} \tag{3.19}$$

Algorithm 2 Person Re-identification

Input: Probe image I
Output: A vector of similarities
Procedure:
Assign a color category to each image.
Extract features using Algorithm 1. ($p_1 = (1, 0, 0), p_2 = (0, 1, 0)$, $p_3 = (0, 0, 1)$).
Apply PCA reduction technique retaining 90% of the variance.
Compute $S1$ and $S2$. (Eq. 3.14 and 3.15)
Find the Projection matrix W (Eq. 3.17). Project the feature vectors onto the new feature subspace.
Compute the kernel similarity function (Eq. 3.18) between each image and the prototypes to construct the vectors of similarities.
Calculate the similarity (Eq. 3.19) between the probe image and each of the gallery images taking in consideration the color category of each image.

3.6 Experimental Results

3.6.1 Dataset and evaluation protocol

In this section we present the evaluation of our method. The experiments are performed on images from VIPeR [1] dataset, the most challenging dataset for person re-identification. This dataset contains a significant amount of viewpoint changes (0, 45, 90, 135 and 180 degree), illumination variations and occlusions between persons. It is designed for a single-shot scenario and it contains image pairs of 632 persons normalized to 48 x 128 pixels. Most

1. Available at http://soe.ucsc.edu/dgray/

of the images contain a viewpoint change of 90 degrees or more in addition to changes of the illumination, making this problem very challenging. The images are randomly split into 2 sets, 316 pairs for each. We use one set for training and the other for testing. The steps we followed in our method are summarized in Algorithm 2.

3.6.2 Results

The performance of person re-identification is usually measured with the Cumulative Matching Characteristic (CMC) curve [142]. The CMC curves represent the probability of finding the true match over the top k ranks. Figure 3.4 shows the CMC curves of different combinations using VIPeR dataset. The experiments are repeated 5 times using five random splits and the results are reported using their average values. The methods without Color Categorization means that the similarity score was calculated without taking in consideration the color category of each person. The NN (Nearest Neighbor) method means that for classification, a simple L2 norm distance measure was calculated on the final features.
Table1 compares the result obtained using our proposed method PreidPFCC (Person re-identification via Prototype Formation and Color Categorization) and some of the existing approaches, including ELF (Ensemble of Localized Features) [113], SDALF (Symmetry-Driven Accumulation of Local Features) [119], PRDC (Probabilistic Relative Distance Comparison) [131], RankBoost [143] and KISSME (Keep It Simple and Straightforward Metric) [144]. The performance is presented by rank1 and rank20 accuracy. All the methods have used 316 persons for testing on the VIPeR dataset. The results show that our method outperform all the other approaches, with the rank1 matching rate of 28 % and rank20 matching rate of 87%. The rank1 matching rate has increased 7% when using Color Categorization with the nearest neighbor method (0.20 instead of 0.13) and 10% when using the Prototype Formation as a re-identification process (0.27 instead of 0.20). The experimental results show that both the Prototype Formation and Color Categorization contribute to the improvement of overall performance.

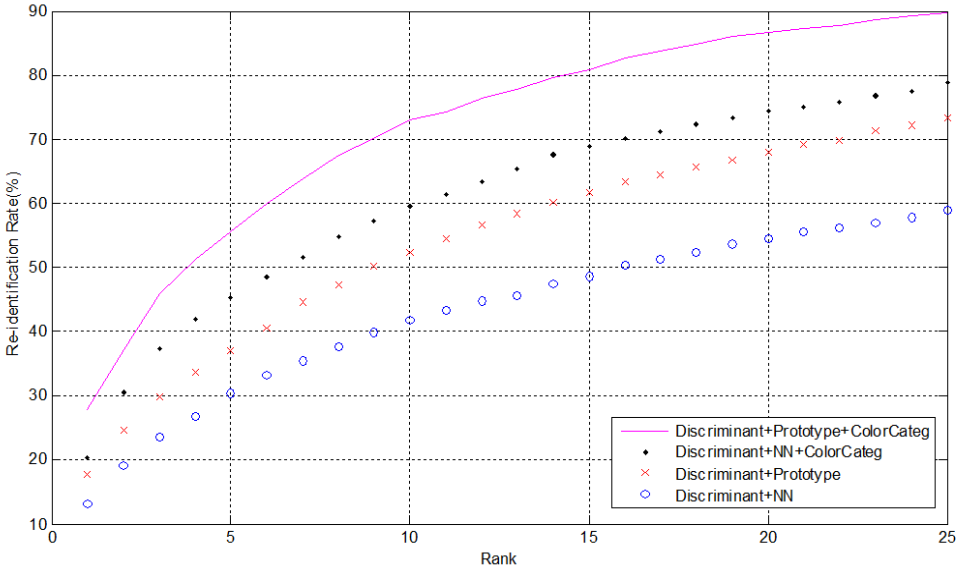


Figure 3.4: CMC plots for different combinations are shown. It is shown that both Prototype Formation and Color Categorization contribute to the improvement of overall performance.

Table 3.1: Top ranked matching rate for different methods.

Method	Rank1 (%)	Rank20 (%)
PreidPFCC	**28**	**87**
ELF	12	61
SDALF	19.87	65.73
PRDC	15.66	70.09
RankBoost	23.92	68.73
KISSME	20	76

In order to highlight the impact of the number of prototypes on the performance, only the Prototype Formation method has been used in Figure 3.5 without the Color Categorization step. It shows that with the increasing number of prototypes, the CMC initially grows before reaching a nearly

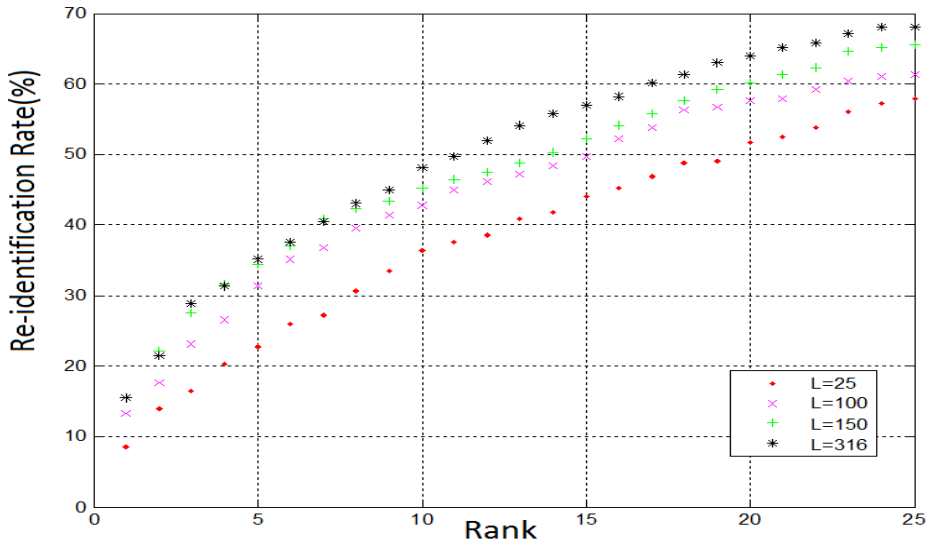


Figure 3.5: CMC plots with different number of prototypes (L).

constant value. That means, once we have a sufficient number of prototypes so that the great part of the discriminative characteristics have been collected, no additional information is embedded in the new prototypes. In Table 2 we analyze the computation time of our method using a Matlab implementation on a 2.8 GHz quad core CPU. Indeed, a key advantage of our approach is its training time efficiency (7.2min) since it does not rely on computationally complex optimization schemes. KISSME seems to have better computational effciency than our approach, however, considering Table1, which reports the matching rate of all the methods, our approach can achieve better performance with acceptable computing complexity. On the other hand, RankBoost with the second best matching rate, has a very long training time and thus it is computationally very expensive.

Table 3.2: Average training time on the VIPeR dataset.

Method	PreidPFCC	ELF	SDALF	PRDC	RankBoost	KISSME
Time	7.2 minutes	5 hours	1.4 hours	15 minutes	10 days	1.34 seconds

3.7 Conclusion

In this chapter, a new framework for single shot person re-identification under two non overlapping cameras was proposed. Unlike previous approaches in which a simple distance metric is learned, a collection of prototypes is utilized to provide a measure to recognize or classify an unseen object. It has been shown that this is more effective than direct comparison between the test image and the gallery. For enhancing the discrimination between different persons, a new representation is given for each feature vector by projecting to a new linear subspace. By applying the Color Categorization procedure, we obtain results that are closer to what humans consider intuitive. The experiments on a challenging dataset show that our proposed method greatly improved the performance of person re-identification. We would like to point out that the proposed approach, due to its robustness to variations in lighting conditions, can be used in many applications of image and video processing.

Chapter 4

Person Re-identification through Deep Learning

Abstract

In this chapter, we applied a deep learning approach on the problem of person re-identification.We constructed a siamese Convolutional Neural Network and trained it by minimizing a contrastive loss function. Experimental results are presented and a comparison between handcrafted features and learned features is given in the end.

Contents

4.1 Introduction

In conventional machine learning, the performance of a model is usually highly affected by the way the data is represented. Thus, these methods were limited in their capacity to deal with raw images. As Chapters 2 and 3 demonstrate, it is evident that constructing a pattern recognition system relying on careful hand-engineered features demands considerable domain expertise for data representation. These features transformed the raw images into suitable representation that is robust to the high amount of variation present in the raw data.

Comparing pairs of images and deciding if they correspond to each other or not is quite challenging and is probably one of the most fundamental problems in computer vision. Feature based matching was for years the most common form of machine learning. Usually, a feature descriptor is defined to extract features and a compatible matching strategy is used for classification. These two steps are independent and a wise combination of different descriptors and matching strategies might adapt to some specific applications.

While hand-engineering is certainly one way of approaching this problem of data representation, in many cases, these features are becoming more and more complex resulting in a difficulty with coming up better, more complex features. Meanwhile, some researchers have been focusing on developing algorithms which incorporate automatic learning of features from raw images. These models present an alternative way of representing the data relying on multiple layers of non-linearity. Moreover, these models generate richer and more sophisticated features than would be possible by hand-engineering alone. This property was considered to be very important and this lead to the development of the first deep learning models.

Despite being around since the 1990s, deep neural network blossomed in the last few years leading to excellent performance on multiple tasks such as natural language processing, speech recognition and visual recognition. At first, it was difficult to train deep neural networks because the lack of the training data. But with the fast progress in the amount of labeled training data and the recent advances in the computational power of GPUs, the research on deep networks emerged quickly and achieved very good performances on various applications. Among all the types of deep neural network,

convolutional neural network was particularly studied for image analysis and are considered state of the art for a number of tasks including image classification [145], face recognition [146] and object detection [147].

A convolutional neural network is a one particular type of deep, feedforward network that is easy to train and can be generalized much better than other types of deep neural networks. In this chapter, we applied a siamese convolutional neural network, initially proposed for signature verification, to the problem of person re-identification. Moreover, in order to understand the contribution of each body part alone, we constructed 3 siamese networks for 3 different body parts. Finally, a comparison between handcrafted features and learned features is presented. In order to familiarize the reader with the concept of convolutional neural networks, we will first represent the basic operational principles of these networks as well as a small historical overview of their development and motivation.

4.2 Biological Motivation

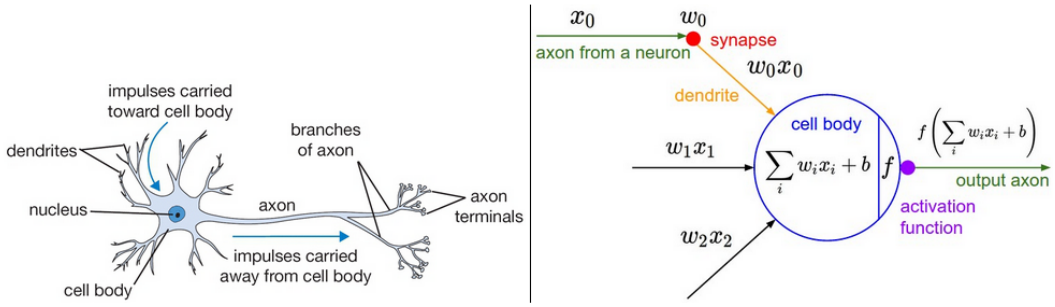


Figure 4.1: Biological neuron (left) and its mathematical model (right). (Image from http://cs231n.github.io/neural-networks-1/)

Every mathematical model is developed based on some computational item, for example, sets, numbers, and vectors. While in the world of mathematics the basic component can be invented from scratch, the case of the brain is constrained by its biophysical structure. The neuronal cell is the basic computational device of the brain. The average human brain has approximately 100 billion neurons. Each neuron may be connected to 10,000

other neurons, passing signals to each other via as many as 1,000 trillion synapses, equivalent to a computer with 1 trillion bit per second processor. Neurons receive signals from its dendrites and send outputs via its axon. Figure 4.1 shows an illustration of a neuron (left) and a common mathematical model (right).

In the mathematical model of a neuron, the pulse travels along the cell's axons (e.g. x0), and is transfered across a multiplicative synapse (e.g. w0x0) to a neighboring neuron which receives it through its dendrites. A synapse is a complex membrane junction characterized by its synaptic strength (e.g. w0). In other words, the influence of neurons on each other is quantified by the synaptic strengths called the weigths (w). The electro-chemical pulse sent by a particular neuron may be such as to encourage the receiving cell to also fire. The signal received at the cell body are all get summed, and the neuron is encouraged to fire or prevented from firing depending on the comparison between the result and a predefined threshold.

4.3 Artificial Neural Networks

A neural network is a biologically inspired mathematical model made of artificial neurons and interconnections in a way that mimic a biological neural network. In 1952, Alan Lloyd Hodgkin and Andrew Huxley won the Nobel prize for characterizing a differential equation that describes the membrane potential with a neurone. However, a practical model that only describe the basic input output relationship was proposed by Warren S. McCulloch and Walter Pitts in 1943. It is an adaptive model, in the sense that the interconnections can be updated with a learning technique. Three main components are used in different combinations in order to create a neural network:

- Weighted inputs, corresponding to the interconnecting synapses and the dendrites of a neuron.
- Summation of the weighted inputs, corresponding to the membrane and soma of the neuron.
- Activation function, which historically has been a threshold function yielding a (binary) action potential, corresponding to the axon of the neuron. However, in artificial neural networks, other activation func-

tions are typically used.

The output of each neuron (illustrated in Figure 4.1 (right)) is the weighted sum of the input values mapped by the activation function f:

$$\phi = f(\sum_i \omega_i x_i + b) \tag{4.1}$$

In general, different activation function achieve similar results, unless specific requirements exist on the activation for the given task. However, different activation functions can lead to very different training speeds and computation times. Instead of considering the activation function to be an operational part of a neuron, it can be thought of as a special type of neuron that maps one input to one output. For the sake of simplicity, the most common ones are characterized below.

4.3.1 The Sigmoid activation function

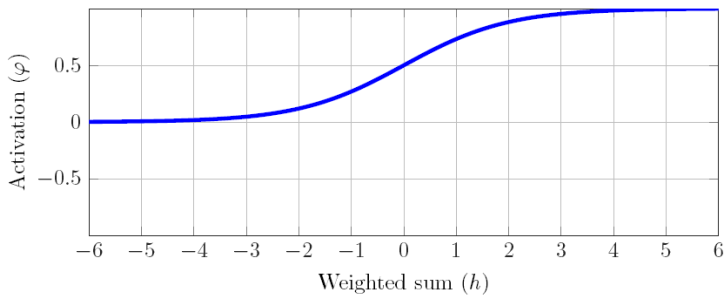


Figure 4.2: Sigmoid activation function

The Sigmoid function (see Figure 4.2) is a continuous, saturating activation function described by a special case of the logistic function:

$$\phi(h) = \frac{1}{1 + e^{-\beta h}} \tag{4.2}$$

where $\beta = 1$. The SIGMOID does not yield negative activation and is therefore only antisymmetric w.r.t. the y-axis. A very unwanted property of the sigmoid function is that when the neuron's activation saturates at either

tail of 0 or 1, the gradient at these points is close to null. Recall that during backpropagation, this gradient will be multiplied to the gradient of this gate's output for the whole objective. Thus, in case the local gradient is close to zero, it will vanish the gradient and almost no signal will flow through the neuron to its weights and recursively to its data. Moreover, one must pay attention when initializing the weights in this case to prevent saturation. For example, if the initial weights are too large then most neurons would become saturated and the network will hardly learn.

4.3.2 Rectified Linear Unit (ReLU)

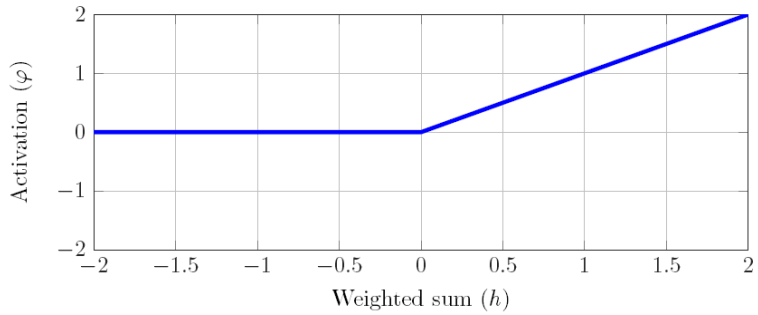


Figure 4.3: The Rectified Linearity Unit (ReLU)

The Rectified Linear unit (see Figure 4.3) is a non-continuous, non-saturating activation function described by $\phi(h) = max(h, 0)$:

$$\phi(h) = \begin{cases} h & \text{if} \quad h > 0 \\ 0 & \text{if not} \quad h < 0 \end{cases} \tag{4.3}$$

The rectifier (ReLU) is, as of 2015, the most popular activation function for deep neural networks. It accelerates the convergence of stochastic gradient descent compared to the sigmoid functions. This is because of its linear nature and non-saturating form. Compared to sigmoid neurons, the rectifier can be implemented by simply thresholding a matrix of activations at zero. Unfortunately, ReLU units can be expired during training. For example, a large gradient can update the weights so that the neuron will never fire again. In this case, the gradient will be zero from that point on.

4.4 Deep Learning

Deep-learning methods are hierarchical learning methods based on multiple levels of features extracted by a set of algorithms. High level representations are obtained by modeling non-linear modules that each convert the representation at one level (from raw images) into a representation at a higher level. The objective is to escape from handcrafted features through end-to-end learning on raw data. In fact, raw data interacts with various factors that can be learnt from one layer to another with the composition of enough transformation. This learning procedure forms the credit assignment path (CAP) that specify the possible connections between the input and the output [148], known as Deep Learning.

The term deep comes from the fact that the depth in these networks is more as compared to the classical shallow neural networks. By introducing more hidden layers, Machine Learning moved onto an influential trend, which can be proved by several successful applications. For example, Google has declared that its own voice search had taken a new turn by adopting Deep Neural Network as the core technology to model the sounds [149]. Deep neural network replaced GMM (Gaussian Mixture Model) which has been leading for many years. Google and Stanford also developed virtual drug detection techniques with the use of deep learning architecture [150]. Another successful application in Deep Learning is the self driving cars, which map raw pixels from a single-front-facing camera directly to steering commands [151].

4.4.1 Convolutional Neural Network

Convolutional Neural Network (CNN) is a one particular type of deep, feedforward network inspired by the natural visual perception mechanism of the living creatures. Compared with other deep neural networks, it is easier to train and it has achieved very good performances during the time when neural networks were out of favor. Convolution neural networks are designed to process data in the form of multiple arrays (for example color images) and have been used for many computer vision applications. These networks take advantage of the properties of natural signals like local connections, shared weights, pooling and the use of many layers. Starting from their roots in

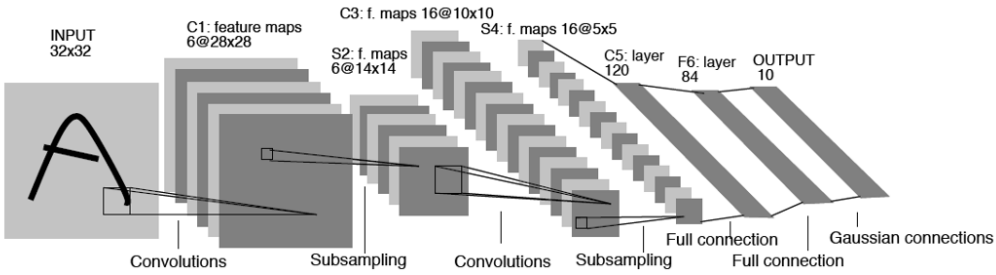


Figure 4.4: A convolutional neural network for classification, from LeCun et al. [2]

digit classification [152], to more recent systems for object detection and image classification [153, 154], CNNs have been widely used for many applications. A major recent success for convolutional neural network is face recognition[155].

The success of these networks originate from their capacity to learn many features and merge them all from raw data itself. Despite these successes, they were largely abandoned by the computer vision communities until the ImageNet competition in 2012. A million images containing one thousand different labels were supplied to a CNN and they accomplished breathtaking results, halving the error rates of the best competing approaches [153].

Recent Convolutional neural networks have ten to twenty layers of ReLUs, hundreds of millions of weights, and billions of connections between units. Take the famous convolutional network in Figure 4.4, it consists of 3 types of layers namely convolutional, pooling (subsampling) and fully connected layers. The convolutional layer aims to extract features of the input by applying several convolution kernels which are used to compute different feature maps. The activation function introduces the nonlinearities which are critical to detect nonlinear features. In order to acquire shift-invariance, pooling layers are used to reduce the resolution of the feature maps obtained. A formulation of this architecture can be written as follows:

$$h_{l+1} = pool_l(f(\omega_l * h_l + b_l)), l = 0, ..., L \tag{4.4}$$

where h_0 is the input image, h_l is the hidden layer activations at layer l, ω_l and b_l are the the convolution kernel and feature bias, f is the activation function and *pool* is the pooling layer. Finally, after several convolutional and pooling layers, fully connected layers take all neurons in the previous layer and connect them to each neuron to generate global semantic information. These models usually learn their weights using back propagation and stochastic gradient descent [2]. First an objective function is defined, e.g cross entropy classification loss, then the gradient of the error with respect to all model parameters is calculated using the chain rule through the hidden layers.

4.4.2 The Classic Backpropagation Algorithm

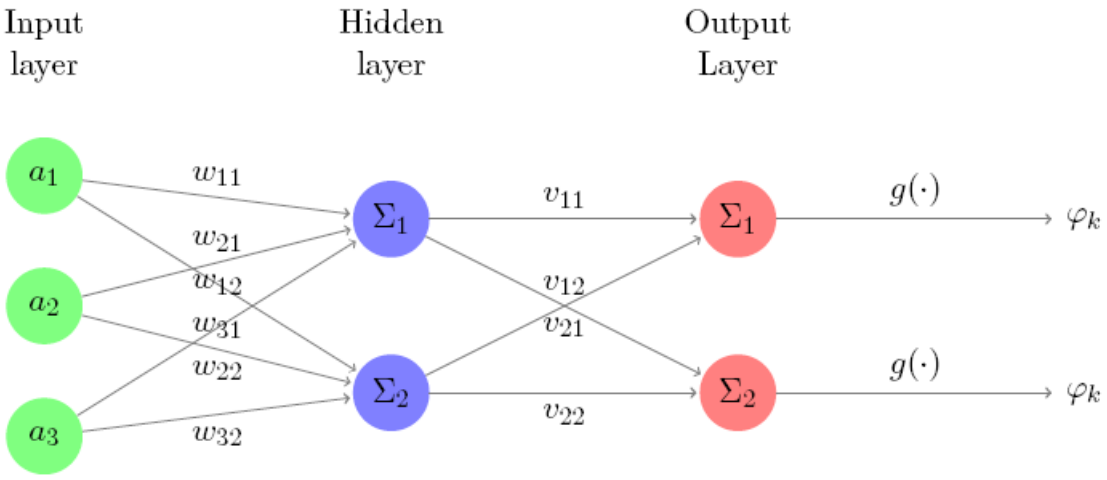


Figure 4.5: A schematic of a Multilayer Perceptron featuring three input values, a_n, two neurons in the hidden layer and two neurons in the output layer. The hidden layer has activation function f($\cdot$) and the output layer has activation function g($\cdot$). In this example N = 3, M = 2 and K = 2.

Historically, multilayer perceptron (Figure 4.5) were trained by utilizing the Backpropagation algorithm described below. In this example, the sigmoid

function is used as activation function:

$$\phi = f(h) = \frac{1}{1 + e^{-h}} \tag{4.5}$$

where h is the weighted sum computed by a neuron. The sigmoid function has the derivative:

$$\frac{\partial \phi}{\partial h} = \phi(1 - \phi) \tag{4.6}$$

The algorithm works as follows:
1) Compute the activation of all neurons in the network, yielding an output (the forward pass):

- Activation of neuron m in the hidden layer:

$$\phi_m = \frac{1}{1 + e^{-h_m}} \tag{4.7}$$

  where $h_m = \sum_{n=1}^{M} a_n w_{nm}$
- Do this for all the hidden layers until you get the output layer, in this example, we only have one layer with the following activation function:

$$\phi_k = \frac{1}{1 + e^{-h_k}} \tag{4.8}$$

  where $h_k = \sum_{m=1}^{M} \phi_m v_{mk}$

2) Compute error and correct weights layer-wise (the backward pass):

- Compute the error at the output: $\delta_{ok} = (t_k - \phi_k)\phi_k(1 - \phi_k)$
- Compute the error at the hiddenlayer: $\delta_{hm} = \phi_m(1 - \phi_m) \sum_{k=1}^{K} v_{mk}\delta_{ok}$
- Update the output layer weights: $v_{mk} = v_{mk} + \alpha\delta_{ok}\phi_m$
- Update the hidden layer weights: $w_{nm} = w_{nm} + \alpha\delta_{hm}a_n$

where α is the learning rate.

This original backpropagation algorithm is performing a gradient descent optimization. If we consider the input vector x and predicted output $\hat{y}$, we define the loss function, l, as the cost of predicting $\hat{y}$ when the target is y, i.e. $l(\hat{y}, y)$. We know that the predicted output $\hat{y}$ can be thought of as a transformation of the arbitrary input vector by a function, f, parametrized by the weights inside the network, i.e. $\hat{y} = f_w(x)$. Now, the loss function can be described as $l(\hat{y}, y) = l(f_w(x), y)$, or $Q(z, w) = l(f_w(x), y)$ where z is an input and output data pair (x,y).

In this framework, gradient descent optimization is performed by updating the weights according to:

$$v_{t+1} = \mu v_t - \alpha \frac{1}{n} \sum_{i=1}^{N} \nabla_{w_t} Q(z_t, w_t) \tag{4.9}$$

$$w_{t+1} = w_t + v_{t+1} \tag{4.10}$$

Tuning the learning rates is an expensive process, so much work has gone into devising methods that can adaptively tune the learning rates, and even do so per parameter. Many of these methods may still require other hyperparameter settings, but the argument is that they are well-behaved for a broader range of hyperparameter values than the raw learning rate. Gradient descent with backpropagation is not guaranteed to find the global minimum of the error function, but only a local minimum; also, it has trouble crossing plateaux in the error function landscape. This issue, caused by the non-convexity of error functions in neural networks, was long thought to be a major drawback, but in a 2015 review article, Yann LeCun et al. argue that in many practical problems, it is not. The efficiency of backpropagation depends partly on the choice of the parameters. The choice of learning rate is important for the method, since a high value can cause too strong a change, causing the minimum to be missed, while a too low learning rate slows the training unnecessarily. The success and speed of backpropagated learning are also dependent on the initial values assigned to the weights. This is demonstrated by considering the repeated training of a network in which the learning parameters are held constant but the initial weights are systematically adjusted between trials.

4.5 Siamese Convolutional Neural Network

4.5.1 Introduction

In the following, we will describe the siamese architecture that we are going to use for the problem of person re-identification. Figure 4.6 illustrates

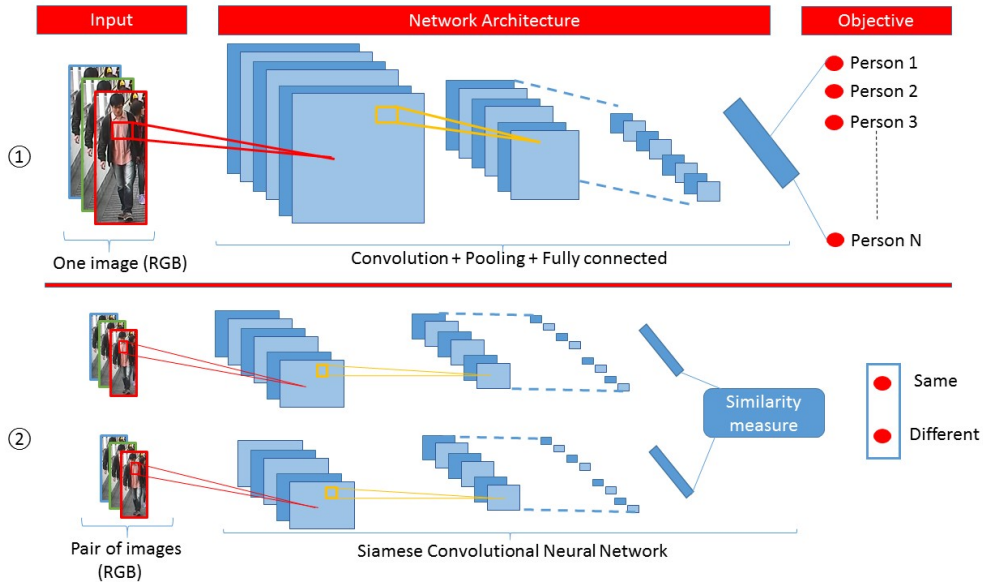


Figure 4.6: Classical CNN (image 1) and the siamese CNN (image 2)

two different approach to deal with the re-identification problem: The classical CNN (image 1) and the siamese CNN (image 2). The SCNN is a method that can be used in cases where the number of samples per category is not large. Traditional methods of classification are not suitable, in general, for the cases where the number of labels is very large and in the same time the number of samples per label is very small. These cases include for example person re-identification where the number of persons can be in hundreds with only few images for each person.

The Siamese Convolutional Neural Network was initially presented in [156] for signature verification. The word siamese is used to highlight the fact that identical CNN parameters and architecture are applied to two different input images. In [157], the authors used the siamese architecture for digit recognition. Instead of designing a feature descriptor capable of handling the considerable changes in the shape of each digit written by different persons, the siamese CNN is used to achieve a representation that can map the input image into a space where the same digits are located close to each other. This space is determined by the last layer of the CNN (a fully con-

nected layer in general). A CNN's ability to generate robust representations lies in its deep architecture (multiple layers). This characteristic gives the model the ability to generate descriptors that are invariant against transformations. Therefore, a CNN can be used to train descriptors for patch comparison.

4.5.2 Siamese descriptor's architecture

In conventional neural networks, the loss function is calculated using the sum over all samples. In the siamese case, the loss function runs over pairs of samples. During the training, the network is fed by pairs of images of which we know whether they belong to the same label or not. Within the framework of these criteria, it is crucial that the homologous training images include the transformations needed to be learned and that the network should be tolerant to. The architecture of the siamese network is given in Figure 4.7. The main idea is to apply the same CNN using the same parameters to each of the images that should be tested for correspondence . In the training phase, an objective function is optimized after calculating the L2 norm of the differences of the resultant descriptors. The parameters are updated so that the L2 distance is as discriminative as possible in differentiating homologous from different matches. As a result, the descriptor obtained is more tolerant to the kind of geometric distortions present within homologous training examples.

Let $\mathbf{X}_1, \mathbf{X}_2$ be a pair of input vectors given to the network. Let $\mathbf{Y}$ be a binary label assigned to this pair. $\mathbf{Y} = 0$ if $\mathbf{X}_1$ and $\mathbf{X}_2$ belong to the same identity, and $\mathbf{Y} = 1$ if $\mathbf{X}_1$ and $\mathbf{X}_2$ belong to two different labels. Let $\mathbf{G_W}$ represents the CNN and $\mathbf{G_W}(\mathbf{X}_1)$ is the output of the CNN when feeded by $\mathbf{X}_1$. The euclidean distance $\mathbf{D_W}$ between $\mathbf{X}_1$ and $\mathbf{X}_2$ can be defined as the following:

$$\mathbf{D_W}(\mathbf{X}_1, \mathbf{X}_2) = ||\mathbf{G_W}(\mathbf{X}_1) - \mathbf{G_W}(\mathbf{X}_2)||_2 \tag{4.11}$$

The loss function to be optimized can be written in the following form:

$$\mathbf{L}(\mathbf{W}, \mathbf{Y}, \mathbf{X}_1, \mathbf{X}_2) = (1 - \mathbf{Y})\frac{1}{2}(\mathbf{D_W})^2 + (\mathbf{Y})\frac{1}{2}max(0, m - \mathbf{D_W})^2 \tag{4.12}$$

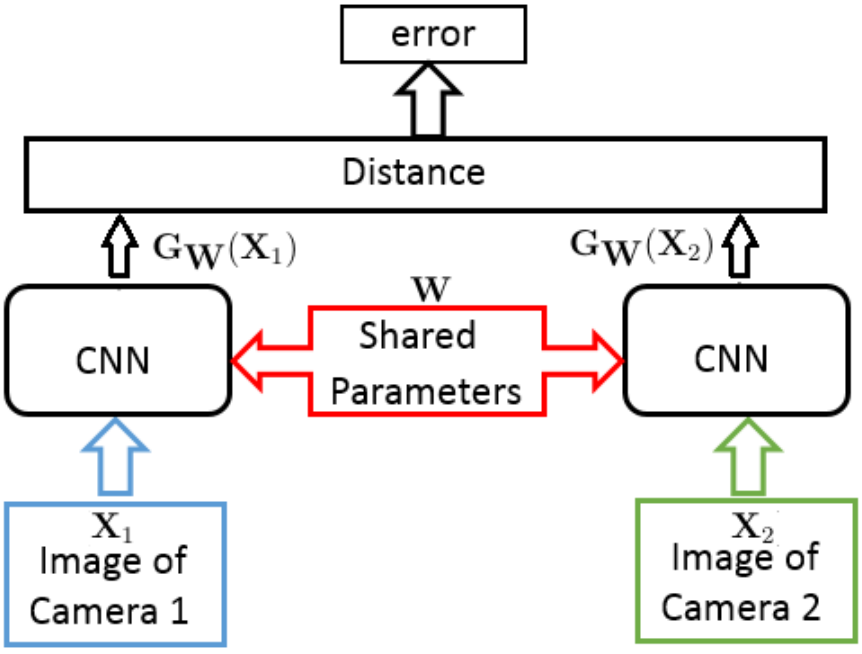


Figure 4.7: Siamese Convolutional Neural Network.

As can bee seen in Equation 4.12, heterogeneous pairs contribute to the loss function only if their distance is within the acceptable margin m, which is a positive constant defining a radius around $\mathbf{G_W(X)}$. One can clearly see that minimizing the above loss function would pull the descriptors of homologous pairs closer to each other, while pushing the descriptors of heterogeneous pairs away from each other. Figure 4.8 shows an illustration of this idea. Before training, the feature descriptors are randomly localized in the feature space, while after training the feature descriptors from images corresponding to matching pairs are closer to each other.

The remaining challenge is computing L and $\partial L/\partial W$. Authors of [157] showed that an efficient method of computing and minimizing L is to construct a siamese network which is two copies of the CNN that share the same parameters W. An indicator variable Y selects whether each input pair $X1, X2$ is a positive ($Y = 0$) or negative ($Y = 1$) example. This entire structure can now be viewed as a new bigger network that consumes inputs

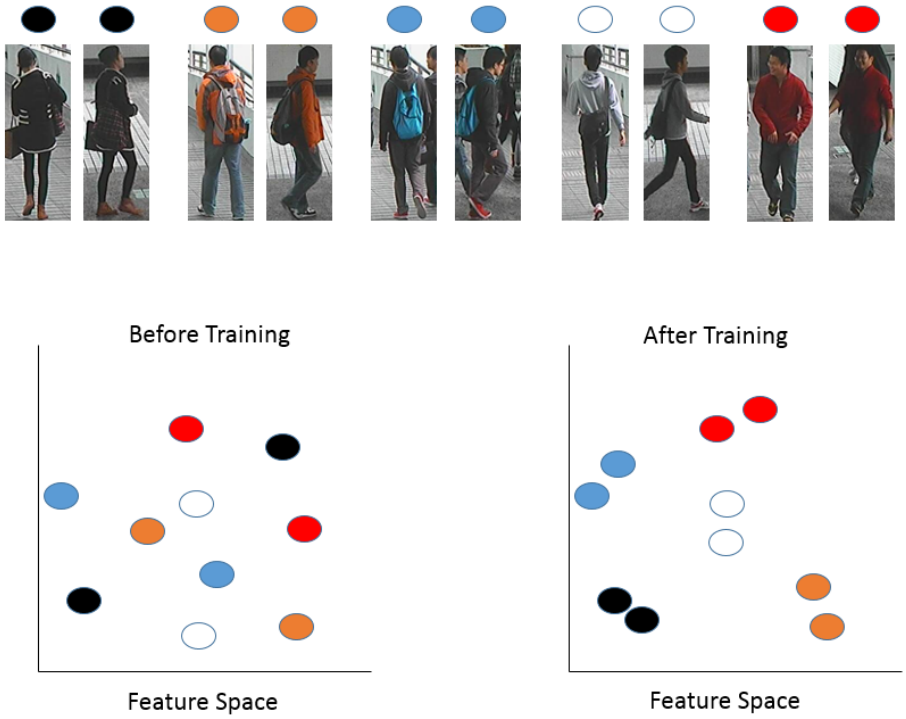


Figure 4.8: In the top part, each colored circle symbolize an image. Identical colors indicate matched pairs from different camera views. In the lower part, an illustration of the images projected into the feature space before and after training is presented.

$X1, X2, Y, W$ and outputs L. With this view, it is now straightforward to apply the backpropagation algorithm and efficiently compute the gradient $\partial L/\partial W$.

4.6 Experiments

4.6.1 Training startegy

Training the CNN is grounded on gradient descent to optimize the objective function. The back propagation algorithm presented in [158] is among the most widely used techniques to determine gradients of the objective function with respect to the parameters. Classical approaches take in consider-

ation all the training set (batch training) to calculate the gradients using only one training sample at a time. This methodology suffers from instability because of training errors when calculating gradient for only one sample. To overcome this, researchers normally update the parameters using mini-batches, computing the gradient descent for small groups of training samples in each iteration. Each mini-batch generally include several hundreds of training examples. In this work, the mini batch size was set to 300 with an equal learning rate for all layers (0.0001).

The network weights are initialized from a gaussian distribution with a mean equal to zero and a standard deviation equal to 1/fan-in where fan-in is defined as the number of incoming weights to each neurone in a particular layer of the network. During the training, pairs of camera A and camera B are randomly chosen and fed each to the corresponding sub-network in the SCNN. Pairs of images corresponding to the same label are labeled as positive ($\mathbf{Y} = 1$), and those corresponding to different label are labeled as negative ($\mathbf{Y} = 0$). In the testing stage, one image from the first camera view of the subject is used as gallery image, and the image corresponding to the other camera view is used as probe.

4.6.2 CNN architecture

This section describes the proposed CNN architecture. As can be seen in Table 4.1 the CNN is composed of 6 convolution layers, 3 pooling layers, and two fully connected layers. The output of the CNN has 100 dimensions. The pair of images are filtered by the stack of convolution layers with a very small receptive field: 3x3. The stride of the convolution is set to one pixel and the pooling is carried out through max-pooling layers performed over a 2x2 pixel window with a stride equal to two. After a stack of convolution layers, we have two fully connected layers where the first one has 1024 dimension and the second has 100 dimension. ReLU neuron is used as activation function for each layer.

Table 4.1: Convolutional Neural Network Architecture.

Name	Type	Number	Filter size	Stride	Activation function
Conv0	Convolution	16	3x3	1	Rectify
Conv1	Convolution	32	3x3	1	Rectify
Pool0	Max pooling		2x2	2	
Conv2	Convolution	32	3x3	1	Rectify
Conv3	Convolution	64	3x3	1	Rectify
Pool1	Max pooling		2x2	2	
Conv4	Convolution	64	3x3	1	Rectify
Conv5	Convolution	128	3x3	1	Rectify
Pool2	Max pooling		2x2	2	
FC1	Fully connected	1024			Rectify
FC2	Fully connected	100			Rectify

4.6.3 Implementation details

For pair generation, the simplest way is to organize the training images randomly into pairs. According to the identity of the person, we randomly generate for each identity positive and negative pairs. It is evident that the negative pairs are far more than positive pairs. This can cause over-fitting in our system. In practice, in order to reduce over-fitting, we first apply data augmentation to our dataset as in most approaches in deep learning. In fact, the availability of large supervised datasets is indispensable for machine learning to achieve good results. With this in mind, data augmentation has been used to increase the number of training examples by applying simple image transformations that does not alter the semantic level image label. In this work we applied horizontal mirroring to double our training set. Therefore, instead of having only two images for each person in the training, we now have four.

4.6.4 Dataset and evaluation protocol

The dataset used in this work is the CHUK01 released in [159]. In this dataset, there are 971 identities and each identity only has two images in each camera. One hundred persons are used for testing and the 871 remaining persons are used for training, in accordance with FPNN [160]. The CMC curve was used to evaluate the performance (Figure 4.9). Table 4.2 compares the results of our model against ITML [161], LMNN [162], SDALF [119], FPNN [160] and directly using Euclidean distance when using dense color histograms and dense SIFT [163]. The results show that our method outperform all the other approaches, with the rank1 matching rate of 31% and rank25 matching rate of 90%.

Table 4.2: Rank1, Rank5, and Rank10 recognition rate of various methods.

Method	Rank1	Rank5	Rank25
Our method	35%	71%	92%
ITML	17.10%	42%	76%
LMNN	21.17%	49%	83%
SDALF	9.9%	42%	70%
FPNN	27.87%	60%	90%
Euclid	10.52%	28%	60%

Experiments were run on NVIDIA-GTX 1070 GPU and it took around 85-86 seconds per epoch on the CUHK01 dataset. Network training converges in roughly 9-10 hours. The gradients with respect to the feature vectors at the last layer are computed from the contrastive loss function and back-propagated to the lower layers of the network. Once all the gradients are computed at all the layers, we use mini batch stochastic gradient descent (SGD) to update the parameters of the network. The Lasagne-Theano framework was employed to run our experiments.

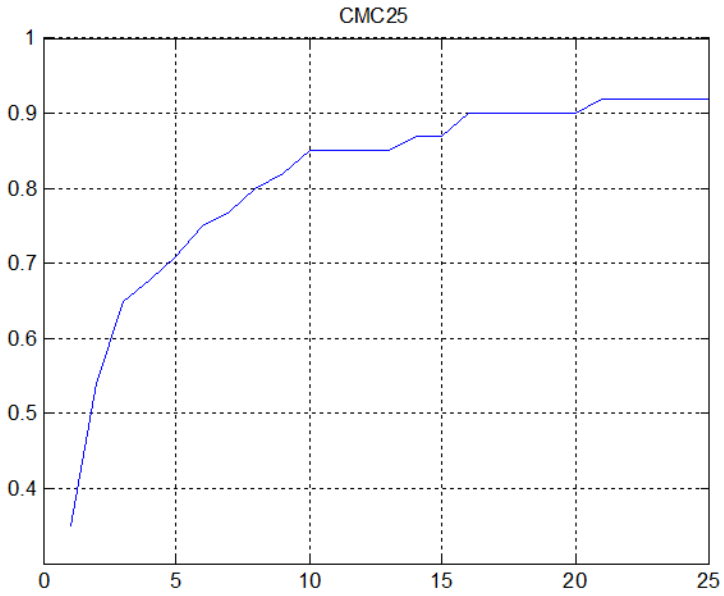


Figure 4.9: CMC curve for the SCNN

4.6.5 Analysis of different body parts

In order to understand the contribution of each body part alone, we constructed 4 different networks on different body parts. One for the head alone, one for the torso, one for the legs and one for the hole body. Figure 4.10 shows the rank curve for the three parts alone and for the person as a hole. The part that perform the best on rank1 is the torse, though at rank25 it becomes clearly that the most discriminative part is the head. This analysis is revealing a direction for future experiments in which several networks are trained and the final decision depends on the accumulation scores of all the networks. This can be very beneficial in controlling situations where severe occlusions occur. The performance differences are more obvious starting from rank12. We can notice that the order of performance after that rank goes to the head then to the leg or torso with very close performances. At rank one, the torso gives the best results, which is consistent with our intuition. Generally, the body is the most stable part in the person images and taking the risk to reidentify a person from the first attempt would be reasonable if based on the clothes of the upper body.

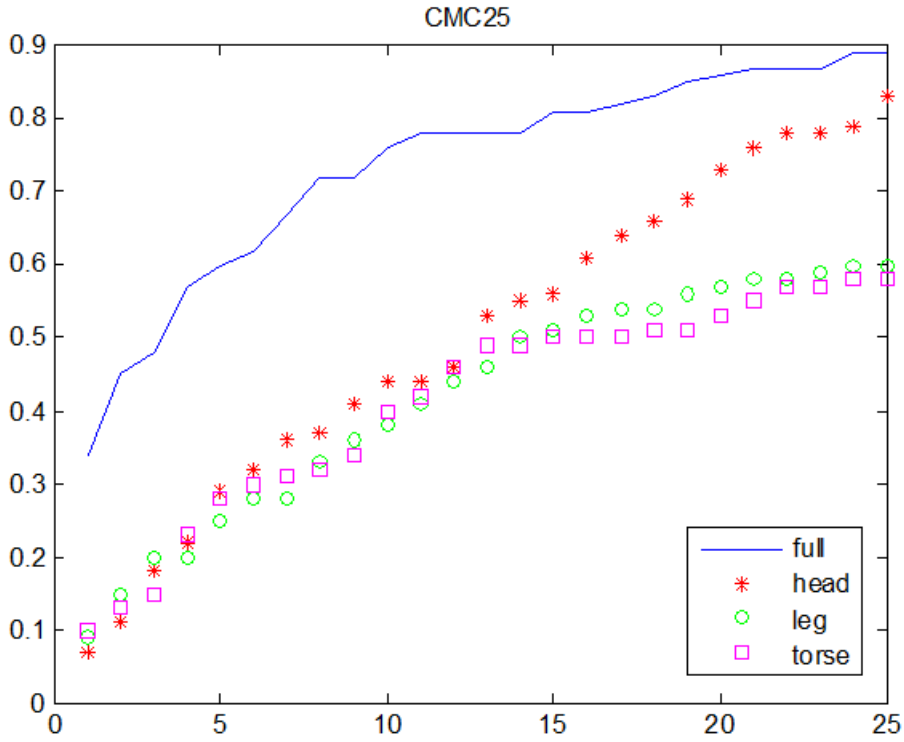


Figure 4.10: CMC curves for different body parts

4.6.6 Visualization

In Figure 4.11, we visualize the feature responses at each layer of the network. After several convolution and subsampling, we can find out which type of features the network has learned. Starting layer 4, it become obvious that stronger responses are given to the body as a whole, while the first layers focused on the upper body. The evidence from this study points toward the fact that the network generally give more attention to the center part of the human (usually torso), which confirm our hypothesis that the color of the clothes is the most important factor for the person re-identification problem.

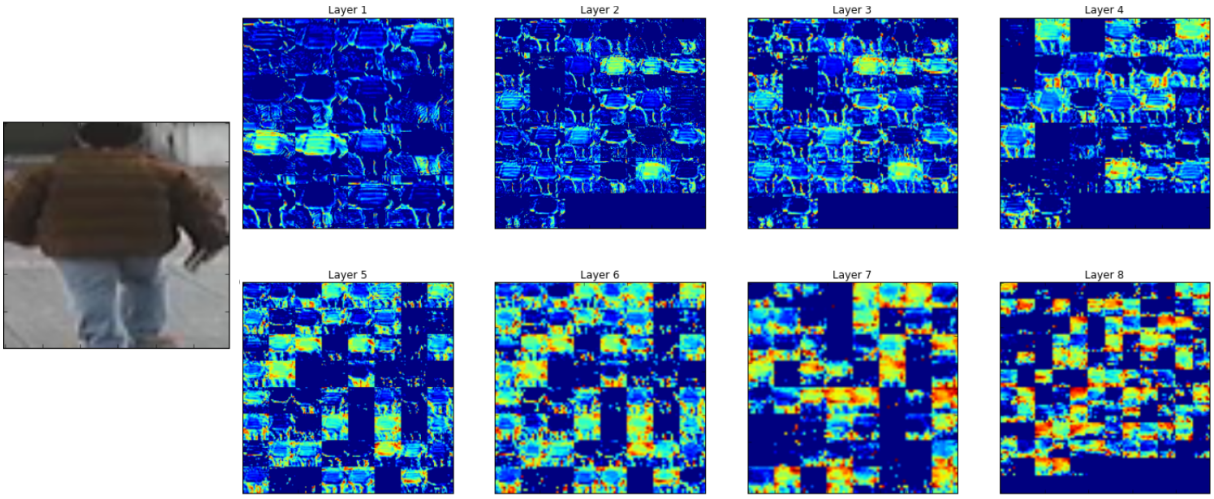


Figure 4.11: Feature visualization for each layer. Warmer colors indicate higher responses. This can be better visualized in color printing.

4.6.7 Learned vs. hand-crafted features

As reported earlier, feature extraction methods can be practically divided into two groups: the hand-crafted features and the learned ones. By hand-crafted features we mean those which are extracted following some careful hand engineered pre-designed algorithms as the ones used in Chapter 3. Unlike the hand-crafted features, the learned ones are obtained through a learning procedure by training a network with a labeled dataset.

In order to compare the performance of both approaches, we applied the same algorithm presented in Chapter 3 on the CHUK01 database using the single-shot settings. The comparative results are presented in Figure 4.12. It can be seen that the learned features obtain the best results achieving 31% on rank1 while the hand crafted features achieved 22%. These outcomes perfectly make sense since learned features have the ability to adapt to the exclusivity of the application whereas handcrafted features are not trained and, therefore, less flexible.

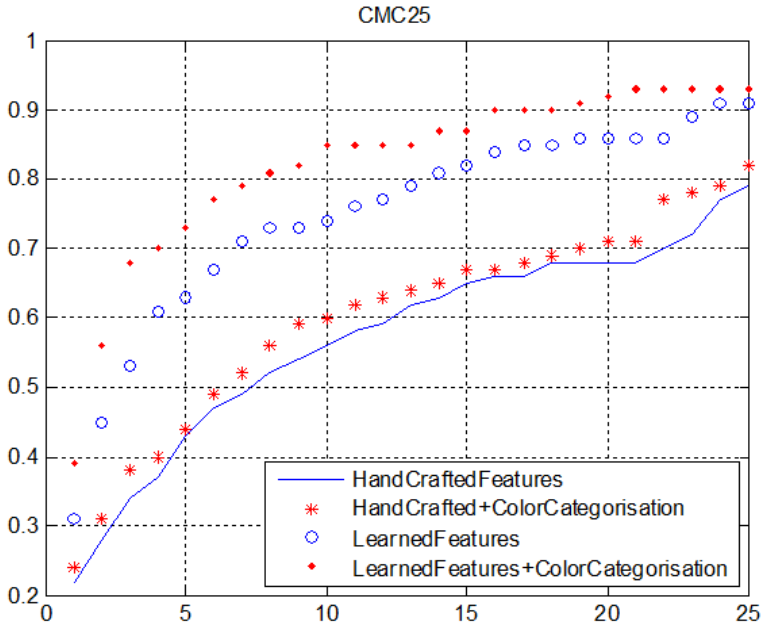


Figure 4.12: Learned vs. hand-crafted features

4.7 Conclusion

In this chapter, a similarity metric has been learned from image pixels directly. Unlike hand crafted features, this deep architecture can learn the texture information and reduce the dimensionality in the same time. This is more practical than the combination method used in Chapter 3. Our experimental results justify the importance of each part of the body in the person re-identification problem. To increase the performance, we suggests to perform more data augmentation. In this sense, a pose estimation of the full body will be needed in order to generate more virtual images of the same person. We would also like to explore end-to-end fine-tuning given the unsupervised learned networks, which is less expensive than training from scratch.

Chapter 5

Conclusion

Abstract

In this chapter, we summarize and conclude the developped work and discuss the advantages of the proposed methods as well as their limitations. Finally we show some directions for future work.

5.1 Concluding Remarks

In this thesis, we presented different approaches to be used in a video surveillance context such as re-identifying a person in different locations using both a high level layer as body appearance and a low level layer as face biometric information. For the low level layer which is focused on faces, we only tackled the face pose estimation problem, which is a major complication in the context of face recognition. For the high level appearance based approaches, the analysis is split in two different techniques. The first one is relying on careful hand-engineered features, and the other one is based on learned features through a deep learning technique.

5.2 Summary of contributions

This thesis makes several contribution to the field of video surveillance. Our work focused on constructing new approaches to the problem of re-

identification. Though we did not tackle in our work the implementation of this kind of techniques, from a theoretical point of view, this thesis provides new methods in the field of computer vision and machine learning. We can summarize these contributions in the following:

- A new technique for model-less face pose estimation was proposed. Starting from a model proposed for age-estimation, we propose a new linear embedding by exploiting the connections between facial features and pose labels via sparse coding scheme. The resulting technique is called Sparse Label sensitive Locality Preserving Projections (Sp-LsLPP). It has less parameters compared to related works and can also outperform them in terms of accuracy.
- In Chapter 3, we proposed to use the Prototype Formation in the person re-identification problem. It was proposed in psychology and cognition field suggesting that human brain recognizes and differentiates the world using prototypes. Moreover, a Color Categorisation technique was proposed in order to increase the robustness of the algorithm against results that are counterintuitive to a human operator. The discrimination between different persons is also improved by learning a linear subspace in a training phase during which person correspondences are assumed to be known.
- Contrary to the hand crafted image features presented in Chapter 3, features in Chapter 4 are learned from an image dataset by training a siamese Convolutional Neural Network. This model generates richer and more sophisticated features as long as it is provided by enough training data. Unlike hand crafted features, this architecture can learn the texture information, project to a subspace for more discrimination and apply a dimensionality reduction at once. This is more practical than the combination methods used in Chapter 3.
- A contribution that covers all the Chapters is to show different strategies that can assist to automatic analysis of images either on face pose estimation part either on person re-identification algorithms. We also compared learning based features algorithms with the handcrafted ones, showing the superiority of the first one. Though this superiority in terms of performance, these methods are easy to overfit on single shot databases as the case of Chapter 3. This outcome constitute a reminder that the handcrafted features may still have favorable characteristics and benefits specially in cases where the learning database is not sufficient to train a deep network.

5.3 Future work

In this thesis, we have developed algorithms and techniques to achieve a successful person re-identification application. There is much more to be done to achieve this goal, however we believe the models proposed in this work are valuable contributions towards realizing this objective. Several interesting problems are left open for future research:

- In the face pose estimation problem, future work may investigate the use of hybrid descriptors as well as the extension of the framework to two degrees of freedom associated with the out-of-plane rotation.
- In the classification experiments, it is obvious that low accuracy is due to data imbalance. In other words negative pairs are much more than positive pairs. A promising advancement would be to perform more data augmentation and generate more virtual images of the same person. This may necessitate a pose estimation of the full body to be able to generate out-of-plane image rotations.
- Since features learned from convolutional neural network rely heavily on the training dataset, it is crucial to feed the network with training images that include the transformations needed to be learned and the ones the network should be tolerant to. In our experiments, we used all the images present in the training dataset for simplicity. We think that it is very important for the future of deep learning and machine learning to develop an image selection step to improve the quality of the learned features.
- Our experiments revealed that the model learned can treat different body parts independently. This analysis is revealing a direction for future experiments in which several networks are trained and the final decision depends on the accumulation scores of all the networks. This can be very beneficial in controlling situations where severe occlusions occur.

Bibliography

[1] A. C. Gallagher and Tsuhan Chen. Clothing cosegmentation for recognizing people. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2008.

[2] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, Nov 1998.

[3] N. Sulman, T. Sanocki, D. Goldgof, and R. Kasturi. How effective is human video surveillance performance? In *2008 19th International Conference on Pattern Recognition*, pages 1–3, Dec 2008.

[4] Mary W Green. The appropriate and effective use of security technologies in us schools. a guide for schools and law enforcement agencies. 1999.

[5] BONE M. BLACKBURN, D. and PHILLIPS. Face recognition vendor test 2000. 2001.

[6] P. J. Phillips, P. Grother, R. Micheals, D. M. Blackburn, E. Tabassi, and M. Bone. Face recognition vendor test 2002. In *2003 IEEE International SOI Conference. Proceedings (Cat. No.03CH37443)*, pages 44–, Oct 2003.

[7] Yu-Lun Wei and Chang Hong Lin. Single-shot person re-identification by gaussian mixture model of weighted color histograms. *Intelligent Signal Processing and Communications Systems (ISPACS), 2013 International Symposium, 47-50*, 2013.

[8] Chun-Chao Guo, Shi-Zhe Chen, Jian-Huang Lai, Xiao-Jun Hu, and Shi-Chang Shi. Multi-shot person re-identification with automatic am-

biguity inference and removal. *Pattern Recognition (ICPR), 22nd International Conference, 3540-3545*, 2014.

[9] F. Dornaika, C. Chahla, F. Khattar, F. Abdallah, and H. Snoussi. Discriminant sparse label-sensitive embedding: Application to image-based face pose estimation. *Engineering Applications of Artificial Intelligence*, 50:168 – 176, 2016.

[10] "C. Chahla, F. Dornaika, F. Khattar, F. Abdallah, and H. Snoussi. Sparse feature extraction for model-less robust face pose estimation. In *Sensors, Networks, Smart and Emerging Technologies SENSET2017*, September 2017.

[11] E. Rosch. Natural categories. *Cognitive psychology, 4(3):328-350*, 1973.

[12] B. Klare and A. Jain. Heterogeneous face recognition using kernel prototype similarities. *IEEE TPAMI, 35(6): 1410-1422*, June 2013.

[13] C. Chahla, H. Snoussi, F. Abdallah, and F. Dornaika. Discriminant quaternion local binary pattern embedding for person re-identification through prototype formation and color categorization. *Engineering Applications of Artificial Intelligence*, 58:27 – 33, 2017.

[14] "C. Chahla, H. Snoussi, F. Abdallah, and F. Dornaika". Exploiting color cues to improve person re-identification:. In *7th International Conference on Imaging for Crime Detection and Prevention ICDP-16*, November 2016.

[15] Jae Young Choi, W. De Neve, K.N. Plataniotis, and Yong Man Ro. Collaborative face recognition for improved face annotation in personal photo collections shared on online social networks. *IEEE Transactions on Multimedia*, 13(1):14–28, Feb 2011.

[16] Yiqun Hu, A.S. Mian, and R. Owens. Sparse approximated nearest points for image set classification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 121–128, 2011.

[17] Shih-Chia Huang, Ming-Kai Jiau, and Chih-An Hsu. A high-efficiency and high-accuracy fully automatic collaborative face annotation system for distributed online social networks. *IEEE Transactions on Circuits and Systems for Video Technology*, 24(10):1800–1813, Oct 2014.

[18] Wonjun Hwang, Haitao Wang, Hyunwoo Kim, Seok-Cheol Kee, and Junmo Kim. Face recognition system using multiple face model of hybrid fourier feature under uncontrolled illumination variation. *IEEE Transactions on Image Processing*, 20(4):1152–1165, April 2011.

[19] J. Lu, G. Wang, and P. Moulin. Image set classification using holistic multiple order statistics features and localized multi-kernel metric learning. In *IEEE Conference on Computer Vision*, pages 329–336, 2013.

[20] X. Zhu and D. Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[21] Wei-Lun Chao, Jun-Zuo Liu, and Jian-Jiun Ding. Facial age estimation based on label-sensitive learning and age-oriented regression. *Pattern Recogn.*, 46(3):628–641, March 2013.

[22] Erik Murphy-Chutorian and Mohan Manubhai Trivedi. Head pose estimation in computer vision: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(4):607–626, April 2009.

[23] Christian Wholer. Three-dimensional pose estimation and segmentation methods. In *3D Computer Vision*, X.media.publishing, pages 89–137. Springer London, 2013.

[24] J. Whitehill and J. R. Movellan. A discriminative approach to frame-by-frame head pose tracking. In *IEEE Conference on Automatic Face and Gesture Recognition*, 2008.

[25] F. Dornaika and F. Davoine. On appearance based face and facial action tracking. *IEEE Transactions on Circuits and Systems for Video Technology*, 16(9):1107–1124, September 2006.

[26] L. Unzueta, W. Pimenta, J. Goenetxea, L. Santos, and F. Dornaika. Efficient generic face model fitting to images and videos. *Image and Vision Computing*, 32:321–334, 2014.

[27] G. Guo, Y. Fu, C.R. Dyer, and T.S. Huang. Head pose estimation: Classification or regression? In *IEEE International Conference on Pattern Recognition*, 2008.

[28] J. Aghajanian and S. Prince. Face pose estimation in uncontrolled environments. In *British Machine Vision Conference*, 2009.

[29] Y. Fu and T.S. Huang. Graph embedded analysis for head pose estimation. In *IEEE International Conference on Automatic Face and Gesture Recognition*, 2006.

[30] B. Ma, W. Zhang, S. Shan, X. Chen, and W. Gao. Robust head pose estimation using lgbp. In *Int. Con. on Patt. Recog. ICPR'06*, 2006.

[31] T. Horprasert, Y. Yacoob, and L. S. Davis. Computing 3-d head orientation from a monocular image sequence. In *Proceedings of the 2nd International Conference on Automatic Face and Gesture Recognition (FG'96)*, pages 242–247, Washington, DC, USA, 1996. IEEE Computer Society.

[32] Jian-Gang Wang and Eric Sung. Em enhancement of 3D head pose estimated by point at infinity. *Image Vision Computing*, 25(12):1864–1874, 2007.

[33] Andrew Gee and Roberto Cipolla. Determining the gaze of faces in images. *Image and Vision Computing*, 12(10):639 – 647, 1994.

[34] T. Horprasert, Y. Yacoob, and L. S. Davis. Computing 3-d head orientation from a monocular image sequence. In *Proceedings of the Second International Conference on Automatic Face and Gesture Recognition*, pages 242–247, Oct 1996.

[35] Jian-Gang Wang and Eric Sung. {EM} enhancement of 3d head pose estimated by point at infinity. *Image and Vision Computing*, 25(12):1864 – 1874, 2007. The age of human computer interaction.

[36] Yoshinobu Ebisawa. Head pose detection with one camera based on pupil and nostril detection technique. In *2008 IEEE Conference on Virtual Environments, Human-Computer Interfaces and Measurement Systems*, pages 172–177, July 2008.

[37] T. Weise G. Fanelli, j. Gall, and L. Gool. Real time head pose estimation from consumer depth cameras. In *DAGM*, 2011.

[38] P. Pashalis, X. Zabulis, and A. Argyros. Head pose estimation on depth data based on particle swarm optimization. In *IEEE Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2012.

[39] Sunjin Yu, Joongrock Kim, and Sangyoun Lee. Iterative three-dimensional head pose estimation using a face normal vector. *Optical Engineering*, 48(3):037204–037204–9, 2009.

[40] M. D. Cordea, E. M. Petriu, N. D. Georganos, D. C. Petriu, and T. E. Whalen. Real-time 2(1/2)-d head pose recovery for model-based video-coding. *IEEE Transactions on Instrumentation and Measurement*, 50(4):1007–1013, Aug 2001.

[41] Cristian Canton-Ferrer, Josep Ramon Casas, and Montse Pardàs. *Head Pose Detection Based on Fusion of Multiple Viewpoint Information*, pages 305–310. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[42] C. Canton-Ferrer, J. R. Casas, and M. Pardàs. *Head Orientation Estimation Using Particle Filtering in Multiview Scenarios*, pages 317–327. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[43] Javier E. Martinez, Ali Erol, George Bebis, Richard Boyle, and Xander Twombly. Integrating perceptual level of detail with head-pose estimation and its uncertainty. *Machine Vision and Applications*, 21(1):69, 2008.

[44] David J. Beymer. Face recognition under varying pose. Technical report, Massachusetts Institute of Technology Cambridge, MA, USA, Cambridge, MA, USA, 1993.

[45] S. Niyogi and W.T. Freeman. Example-based head tracking. In *Automatic Face and Gesture Recognition, 1996., Proceedings of the Second International Conference on*, pages 374–378, Oct 1996.

[46] D. J. Beymer. Face recognition under varying pose. In *1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 756–761, Jun 1994.

[47] S. Niyogi and W. T. Freeman. Example-based head tracking. In *Proceedings of the Second International Conference on Automatic Face and Gesture Recognition*, pages 374–378, Oct 1996.

[48] Jeffrey Ng and Shaogang Gong. Composite support vector machines for detection of faces across views and pose estimation. *Image and Vision Computing*, 20(5–6):359 – 368, 2002.

[49] J. Ng and Shaogang Gong. Multi-view face detection and pose estimation using a composite support vector machine across the view sphere. In *Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems, 1999. Proceedings. International Workshop on*, pages 14–21, 1999.

[50] R. Gonzalez and R. Woods. In *Digital Image Processing second edition*, pages 582–584, 2002.

[51] Jamie Sherrah, Shaogang Gong, and Eng jon Ong. Understanding pose discrimination in similarity space. In *10 th British Machine Vison Conference*, pages 523–532. BMVA Press, 1999.

[52] Y. Ma, Yoshinori Konishi, K. Kinoshita, Shihong Lao, and M. Kawade. Sparse bayesian regression for head pose estimation. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 3, pages 507–510, 2006.

[53] Harris Drucker, Christopher J. C. Burges, Linda Kaufman, Alex J. Smola, and Vladimir Vapnik. Support Vector Regression Machines. In *Neural Information Processing Systems*, pages 155–161, 1996.

[54] V. Esposito Vinzi, W. W. Chin, J. Henseler, and H. Wang. Handbook of Partial Least Squares: Concepts, Methods and Applications in Marketing and Related Fields. *European Planning Studies*, 2008.

[55] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[56] Yongmin Li, Shaogang Gong, and H. Liddell. Support vector regression and classification based multi-view face detection and recognition. In *Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition (Cat. No. PR00580)*, pages 300–305, 2000.

[57] Erik Murphy-Chutorian, Anup Doshi, and Mohan M. Trivedi. Head Pose Estimation for Driver Assistance Systems: A Robust Algorithm

and Experimental Evaluation. In *Intelligent Transportation Systems Conference (ITSC), IEEE*, pages 709–714, September 2007.

[58] Yong Ma, Y. Konishi, K. Kinoshita, Shihong Lao, and M. Kawade. Sparse bayesian regression for head pose estimation. In *18th International Conference on Pattern Recognition (ICPR'06)*, volume 3, pages 507–510, 2006.

[59] Hankyu Moon and M. L. Miller. Estimating facial pose from a sparse representation [face recognition applications]. In *Image Processing, 2004. ICIP '04. 2004 International Conference on*, volume 1, pages 75–78 Vol. 1, Oct 2004.

[60] Bernt Schiele and Alex Waibel. Gaze tracking based on face-color. In *In International Workshop on Automatic Face- and Gesture-Recognition*, pages 344–349, 1995.

[61] Liang Zhao, G. Pingali, and I. Carlbom. Real-time head orientation estimation using neural networks. In *Proceedings. International Conference on Image Processing*, volume 1, pages I–297–I–300 vol.1, 2002.

[62] E. Seemann, K. Nickel, and R. Stiefelhagen. Head pose estimation using stereo vision for human-robot interaction. In *Sixth IEEE International Conference on Automatic Face and Gesture Recognition, 2004. Proceedings.*, pages 626–631, May 2004.

[63] R. Stiefelhagen, Jie Yang, and A. Waibel. Modeling focus of attention for meeting indexing based on multiple cues. *IEEE Transactions on Neural Networks*, 13(4):928–938, Jul 2002.

[64] Y. L. Tian, L. Brown, C. Connell, Sharat Pankanti, Arun Hampapur, A. Senior, and R. Bolle. Absolute head pose estimation from overhead wide-angle cameras. In *2003 IEEE International SOI Conference. Proceedings (Cat. No.03CH37443)*, pages 92–99, Oct 2003.

[65] M. Voit, K. Nickel, and R. Stiefelhagen. A bayesian approach for multi-view head pose estimation. In *2006 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pages 31–34, Sept 2006.

[66] R. Rae and H. J. Ritter. Recognition of human head orientation based on artificial neural networks. *IEEE Transactions on Neural Networks*, 9(2):257–265, Mar 1998.

[67] Volker Krüger and Gerald Sommer. Gabor wavelet networks for efficient head pose estimation. *Image and Vision Computing*, 20(9–10):665 – 672, 2002.

[68] Margarita Osadchy, Yann Le Cun, and Matthew L. Miller. *Synergistic Face Detection and Pose Estimation with Energy-Based Models*, pages 196–206. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[69] V.N. Balasubramanian, Jieping Ye, and S. Panchanathan. Biased manifold embedding: A framework for person-independent head pose estimation. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1–7, June 2007.

[70] S.J McKenna and S Gong. Real-time face pose estimation. *Real-Time Imaging*, 4(5):333 – 347, 1998.

[71] J. Sherrah, S. Gong, and E.J. Ong. Face distributions in similarity space under varying head pose. *Image and Vision Computing*, 19(12):807 – 819, 2001.

[72] S. Srinivasan and K. L. Boyer. Head pose estimation using view based eigenspaces. In *Object recognition supported by user interaction for service robots*, volume 4, pages 302–305 vol.4, 2002.

[73] S. Z. Li, Qingdong Fu, Lie Gu, B. Scholkopf, Yimin Cheng, and Hongjiag Zhang. Kernel machine based learning for multi-view face detection and pose estimation. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, volume 2, pages 674–679 vol.2, 2001.

[74] Bingpeng Ma, Wenchao Zhang, Shiguang Shan, Xilin Chen, and Wen Gao. Robust head pose estimation using lgbp. In *18th International Conference on Pattern Recognition (ICPR'06)*, volume 2, pages 512–515, 2006.

[75] Shuicheng Yan, Huan Wang, Yun Fu, Jun Yan, Xiaoou Tang, and T.S. Huang. Synchronized submanifold embedding for person-independent

pose estimation and beyond. *Image Processing, IEEE Transactions on*, 18(1):202–210, Jan 2009.

[76] E. Osuna, R. Freund, and F. Girosit. Training support vector machines: an application to face detection. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 130–136, Jun 1997.

[77] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, volume 1, pages I–511–I–518 vol.1, 2001.

[78] J. Huang, X. Shao, and H. Wechsler. Face pose discrimination using support vector machines (svm). In *Proceedings. Fourteenth International Conference on Pattern Recognition (Cat. No.98EX170)*, volume 1, pages 154–156 vol.1, Aug 1998.

[79] H. A. Rowley, S. Baluja, and T. Kanade. Rotation invariant neural network-based face detection. In *Proceedings. 1998 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No.98CB36231)*, pages 38–44, Jun 1998.

[80] M. Viola, Michael J. Jones, and Paul Viola. Fast multi-view face detection. In *Proc. of Computer Vision and Pattern Recognition*, 2003.

[81] M. Lades, J. C. Vorbruggen, J. Buhmann, J. Lange, C. von der Malsburg, R. P. Wurtz, and W. Konen. Distortion invariant object recognition in the dynamic link architecture. *IEEE Transactions on Computers*, 42(3):300–311, Mar 1993.

[82] Junwen Wu and Mohan M. Trivedi. A two-stage head pose estimation framework and evaluation. *Pattern Recogn.*, 41(3):1138–1158, March 2008.

[83] N. Krüger, M. Pötzsch, and C. von der Malsburg. Determination of face position and pose with a learned representation based on labelled graphs. *Image and Vision Computing*, 15(8):665 – 673, 1997.

[84] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):681–685, Jun 2001.

[85] T.F. Cootes, C.J. Taylor, D.H. Cooper, and J. Graham. Active shape models-their training and application. *Computer Vision and Image Understanding*, 61(1):38 – 59, 1995.

[86] G.J. Edwards, A. Lanitis, C.J. Taylor, and T.F. Cootes. Statistical models of face images — improving specificity. *Image and Vision Computing*, 16(3):203 – 211, 1998.

[87] T. Horprasert, Y. Yacoob, and L. S. Davis. An anthropometric shape model for estimating head orientation. In *In Proceedings of the Third International Workshop on Visual Form*, 1997.

[88] Yuxiao Hu, Longbin Chen, Yi Zhou, and Hongjiang Zhang. Estimating face pose by facial asymmetry and geometry. In *Sixth IEEE International Conference on Automatic Face and Gesture Recognition, 2004. Proceedings.*, pages 651–656, May 2004.

[89] T. S. Jebara and A. Pentland. Parametrized structure from motion for 3d adaptive feedback tracking of faces. In *Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*, CVPR '97, pages 144–, Washington, DC, USA, 1997. IEEE Computer Society.

[90] Youding Zhu and K. Fujimura. Head pose estimation for driver monitoring. In *IEEE Intelligent Vehicles Symposium, 2004*, pages 501–506, June 2004.

[91] K. S. Huang and M. M. Trivedi. Robust real-time detection, tracking, and pose estimation of faces in video streams. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, volume 3, pages 965–968 Vol.3, Aug 2004.

[92] Junwen Wu and Mohan M. Trivedi. A two-stage head pose estimation framework and evaluation. *Pattern Recognition*, 41(3):1138–1158, March 2008.

[93] Jamie Sherrah and Shaogang Gong. Fusion of perceptual cues for robust tracking of head pose and position. *Pattern Recognition*, 34(8):1565 – 1572, 2001.

[94] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, New York, 1990.

[95] H. Li, T. Jiang, and K. Zhang. Efficient and robust feature extraction by maximum margin criterion. *IEEE Trans. on Neural Networks*, 17(1):157–165, 2006.

[96] D.-Q. Dai and P. C. Yuen. Face recognition by regularized discriminant analysis. *IEEE Trans. Systems, Man, Cybernetics, part B*, 37(4):1080–1085, August 2007.

[97] M. Sugiyama. Dimensionality reduction of multimodal labeled data by local fisher discriminant analysis. *Journal of Machine Learning Research*, 8:1027–1061, 2007.

[98] S.C. Yan, D. Xu, B. Zhang, H. Zhang, Q. Yang, and S. Lin. Graph embedding and extension: a general framework for dimensionality reduction. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 29(1):40–51, 2007.

[99] Y. Xu, A. Zhong, J. Yang, and D. Zhang. LPP solution schemes for use with face recognition. *Pattern Recognition*, 43:4165–4176, 2010.

[100] X.F. He, S.C. Yan, Y. Hu, P. Niyogi, and H. Zhang. Face recognition using laplacianfaces. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 27(3):328–340, 2005.

[101] F. Dornaika and A. Assoum. Enhanced and parameterless locality preserving projections for face recognition. In *Neurocomputing*, volume 99, pages 448–457, 2013.

[102] Matthew Turk and Alex Pentland. Eigenfaces for Recognition. *Journal of Cognitive Neuroscience*, 3:71–86, 1991.

[103] Ke Huang and Selin Aviyente. Sparse representation for signal classification. In Bernhard Schölkopf, John C. Platt, and Thomas Hoffman,

editors, *Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada*, pages 609–616. MIT Press, December 2006.

[104] Jun Liu, Shuiwang Ji, and Jieping Ye. *SLEP: Sparse Learning with Efficient Projections*. Arizona State University, 2011.

[105] H. Chen, H. Chang, and T. Liu. Local discriminant embedding and its variants. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 2005.

[106] N. Kramer and M. Braun. Kernelizing pls, degrees of freedom, and efficient model selection. In *International Conference on Machine Learning*, pages 441–448, 2007.

[107] J. Fox and S. Weisberg. *An R Companion to Applied Regression*. SAGE Publications, 2010.

[108] T. Ojala, M. Pietikäinen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *Transactions on Pattern Analysis and Machine Intelligence*, 24:971–987, 2002.

[109] V. Takala, T. Ahonen, and M. Pietikinen. Block-based methods for image retrieval using local binary patterns. In *Image Analysis, SCIA*, volume LNCS, 3540, 2005.

[110] B. Raducanu and F. Dornaika. A supervised non-linear dimensionality reduction approach for manifold learning. *Pattern Recognition*, 25:2432–2444, 2012.

[111] B. Raducanu and F. Dornaika. Embedding new observations via sparse-coding for non-linear manifold learning. *Pattern Recognition*, 47:480–492, 2014.

[112] M. Bauml and R. Stiefelhagen. Evaluation of local features for person re-identification in image sequences. *AVSS, 2011 8th IEEE International Conference, 291-296*, 2011.

[113] D. Gray and H. Tao. Viewpoint invariant pedestrian recognition with an ensemble of localized features. *In ECCV*, 2008.

[114] J. Kang, I. Cohen, and G. Medioni. Object reacquisition using invariant appearance model. *Pattern Recognition. Proceedings of the 17th International Conference, 759-762*, 2004.

[115] W. Hu, M. Hu, X. Zhou, J. Lou, T. Tan, and S. Maybank. Principal axis-based correspondence between multiple cameras for people tracking. *IEEE Trans. Pattern Analysis and Machine Intelligence. 28(4): 663-671*, 2006.

[116] P. Salvagnini, L. Bazzani, M. Cristani, and V. Murino. Person reidentification with a ptz camera: An introductory study. *In ICIP, 3552-3556*, 2013.

[117] R. Satta. Appearance descriptors for person re identification: a comprehensive review. *CoRR.URL http://arxiv.org/abs/1307.5748*, 2013.

[118] Federico Pala, Riccardo Satta, Giorgio Fumera, and Fabio Roli. Multimodal person reidentification using rgb-d cameras. *IEEE Transactions on Circuits and Systems for Video Technology*, 26(4):788–799, 2016.

[119] M. Farenzena, L. Bazzani, A. Perina, V. Murino, and M. Cristani. Person re-identification by symmetry-driven accumulation of local features. *CVPR 2010 IEEE Conference on San Francisco 2360-2367*, 2010.

[120] S. B?k, E. Corvee, F. Bremond, and M. Thonnat. Person reidentification using haar-based and dcd-based signature. In *2010 7th IEEE International Conference on Advanced Video and Signal Based Surveillance*, pages 1–8, Aug 2010.

[121] S. B?k, E. Corvee, F. Bremond, and M. Thonnat. Person reidentification using spatial covariance regions of human body parts. In *2010 7th IEEE International Conference on Advanced Video and Signal Based Surveillance*, pages 435–440, Aug 2010.

[122] Hong Shao, Yueshu Wu, Wencheng Cui, and Jinxia Zhang. Image retrieval based on mpeg-7 dominant color descriptor. *Young Computer Scientists, 2008. ICYCS 2008. The 9th International Conference for, 753-757*, 2008.

[123] M. Hirzer, P.M. Roth, and H. Bischof. Person re-identification by efficient impostor-based metric learning. *Advanced Video and Signal-Based Surveillance (AVSS), IEEE Ninth International Conference, 203-208*, 2012.

[124] B. Ma, Y. Su, and F. Jurie. Local descriptors encoded by fisher vectors for person re-identification. *ECCV Workshops*, 2012.

[125] Zhe Lin and Larry S Davis. Learning pairwise dissimilarity profiles for appearance recognition in visual surveillance. In *International symposium on visual computing*, pages 23–34. Springer, 2008.

[126] W. R. Schwartz and L. S. Davis. Learning discriminative appearance-based models using partial least squares. In *2009 XXII Brazilian Symposium on Computer Graphics and Image Processing*, pages 322–329, Oct 2009.

[127] J. M. Geusebroek, R. van den Boomgaard, A. W. M. Smeulders, and H. Geerts. Color invariance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(12):1338–1350, Dec 2001.

[128] Harro Stokman and Theo Gevers. Selection and fusion of color models for image feature detection. *IEEE transactions on pattern analysis and machine intelligence*, 29(3), 2007.

[129] X. Wang, G. Doretto, T. Sebastian, J. Rittscher, and P. Tu. Shape and appearance context modeling. *IEEE 11th International Conference on Computer Vision, 1-8*, 2007.

[130] M. Hirzer, P. M. Roth, M. Kostinger, and H. Bischof. Re-laxed pairwise learned metric for person re-identification. *In ECCV, 780-793*, 2012.

[131] Wei-Shi Zheng, Shaogang Gong, and Tao Xiang. Reidentification by relative distance comparison. *Pattern Analysis and Machine Intelligence, IEEE Transactions on , 35(3):653-668*, 2013.

[132] B. Berlin and P. Kay. Basic color terms: Their universality and evolution. *Berkeley, CA: Univ. California*, 1969.

[133] T. Hofmann. Probabilistic latent semantic indexing. *In SIGIR*, 1999.

[134] J. van de Weijer, C. Schmid, J. Verbeek, and D. Larlus. Learning color names for real-world applications. *IEEE Transactions in Image Processing*, 2009.

[135] J. van de Weijer, C. Schmid, and J. Verbeek. Learning color names from real-world images. *In CVPR*, 2007.

[136] D. John. Color name identification: fuzzycolor, matlab central file exchange. retrieved 20 sep 2006.

[137] W. Hamilton. Elements of quaternions. *Longmans, Green, and Company*, 1866.

[138] T. Ojala, M. Pietikainen, and D. Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition, 29(1):51-59*, January 1996.

[139] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE TPAMI, 24(7):971-987*, July 2002.

[140] R. Lan, Y. Zhou, Y Yan Tang, Chen, and C. Chen. Person reidentification using quaternionic local binary pattern. *IEEE International Conference on Multimedia and Expo , 1-6*, July 2014.

[141] H. Wang, S. Yan, D. Xu, X. Tang, and T. Huang. Trace ratio vs. ratio trace for dimensionality reduction. *2007 IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis, 1-8*, 2007.

[142] H. Moon and P. Phillips. Computational and performance aspects of pca-based face-recognition algorithms. *Perception, 30(3): 303-321*, 2001.

[143] Cheng-Hao Kuo, Khamis, and S. Shet. Person re-identification using semantic color names and rankboost. *Applications of Computer Vision (WACV), 2013 IEEE Workshop on , Tampa, FL ,281-287*, 2013.

[144] M. Kostinger, M. Hirzer, P. Wohlhart, P. Roth, and H. Bischof. Large scale metric learning from equivalence constraints. *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference, 2288-2295*, June 2012.

[145] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *CoRR*, abs/1409.4842, 2014.

[146] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1701–1708, June 2014.

[147] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Region-based convolutional networks for accurate object detection and segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(1):142–158, Jan 2016.

[148] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *CoRR*, abs/1404.7828, 2014.

[149] Kanishka Rao Françoise Beaufays Hasim Sak, Andrew Senior and Johan Schalkwyk. Google voice search: faster and more accurate. september 2015.

[150] Bharath Ramsundar, Steven Kearnes, Patrick Riley, Dale Webster, David Konerding, and Vijay Pande. Massively multitask networks for drug discovery. *arXiv preprint arXiv:1502.02072*, 2015.

[151] F Nelson. Nvidia demos a car computer trained with deep learning.

[152] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, Nov 1998.

[153] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, page 2012.

[154] Pierre Sermanet, David Eigen, Xiang Zhang, Michael Mathieu, Rob Fergus, and Yann Lecun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *http://arxiv.org/abs/1312.6229*.

[155] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1701–1708, June 2014.

[156] Jane Bromley, Isabelle Guyon, Yann Lecun, Eduard Säckinger, and Roopak Shah. Signature verification using a "siamese" time delay neural network. In *In NIPS Proc*, 1994.

[157] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742, 2006.

[158] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *Cognitive modeling*, 5(3):1, 1988.

[159] Wei Li, Rui Zhao, and Xiaogang Wang. Human reidentification with transferred metric learning. In *Asian Conference on Computer Vision*, pages 31–44. Springer, 2012.

[160] Wei Li, Rui Zhao, Tong Xiao, and Xiaogang Wang. Deepreid: Deep filter pairing neural network for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 152–159, 2014.

[161] Jason V Davis, Brian Kulis, Prateek Jain, Suvrit Sra, and Inderjit S Dhillon. Information-theoretic metric learning. In *Proceedings of the 24th international conference on Machine learning*, pages 209–216. ACM, 2007.

[162] Kilian Q Weinberger and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 10(Feb):207–244, 2009.

[163] Rui Zhao, Wanli Ouyang, and Xiaogang Wang. Unsupervised salience learning for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3586–3593, 2013.

Appendices

Titre de la thèse :

Extraction de descripteurs non linéaires pour la ré-identification d'objets dans un réseau de caméras

Juin 2017

Charbel Chahla

Introduction :

La réplication du système visuel humain utilisé par le cerveau pour traiter l'information est un domaine de grand intérêt scientifique. Ce domaine de recherche, connu sous le nom de vision par ordinateur, est notoirement difficile. Cette thèse se situe dans le cadre d'un système entièrement automatisé capable d'analyser les traits du visage lorsqu'une personne est proche des caméras et suivre son identité lorsque ses traits ne sont plus traçables. La première partie de cette thèse est consacrée aux procédures d'estimation de pose du visage pour les utiliser dans les scénarios de reconnaissance faciale. Nous avons proposé une nouvelle méthode basée sur une représentation sparse. La technique résultante s'appelle Sparse Label sensible Local Preserving Projections.

Dans un environnement encombré et incontrôlé observé par des caméras à distance inconnue, la ré-identification de personne reposant sur des données biométriques conventionnelles telles que la reconnaissance faciale n'est ni réalisable ni fiable. Par contre, les caractéristiques visuelles basées sur l'apparence des personnes déterminées par leurs vêtements, peuvent être exploitées plus efficacement pour la ré-identification. Dans ce contexte, nous proposons une nouvelle approche pour la ré-identification dans un réseau de caméras non chevauchantes. Tout d'abord, une nouvelle représentation est donnée pour chaque vecteur de caractéristiques en les projetant sur un nouveau sous-espace linéaire. Ensuite, une collection de prototypes est utilisée pour fournir une mesure afin de reconnaître un nouvel objet. La robustesse de l'algorithme contre les résultats qui sont contre-intuitifs pour un opérateur humain est améliorée en proposant la procédure Color Categorisation.

Dans la dernière partie de cette thèse, nous proposons une architecture Siamese de deux réseaux neuronaux convolutionnels (CNN), chaque CNN étant réduit à seulement onze couches. Cette architecture permet à une machine d'être alimentée directement avec des données brutes et de découvrir automatiquement les représentations nécessaires à la classification.

Chapitre2 :

L’analyse des mouvements du visage effectué par l’être humain, joue un rôle important dans la communication non verbale. Par exemple, le visage peut se substituer au doigt pour montrer une direction spécifique. Ce problème, aussi aisé pour l’être humain, demande beaucoup d’effort pour rendre une machine capable d’interpréter les mouvements d’une façon automatique. L’estimation de la pose peut également améliorer d'autres tâches de vision par ordinateur comme la reconnaissance du visage.

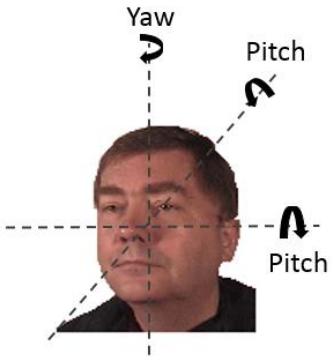


Figure 1: Les 3 angles pour l'éstimation de la pose du visage

Un des problèmes majeurs rencontrés par les techniques actuelles de reconnaissance faciale réside dans la difficulté de traiter les images faciale avec des poses non frontale. Par conséquent, afin d’améliorer la robustesse de la reconnaissance faciale et pour permettre une analyse du visage plus approfondie, beaucoup d’efforts ont été mis pour estimer la pose du visage. La figure 1 montre les trois degrés de liberté d'un visage humain qui peuvent être représentés par les trois angles de rotation: alpha (lacet), bêta (tangage) et gamma (roulis).

La méthode d'apprentissage par machine utilisée dans notre travail est présentée dans Figure 2. Le premier module est un prétraitement d'image classique qui précède une technique d’extraction de caractéristiques. Ainsi, les données d'entrée peuvent être n'importe quel descripteur d’images brutes (par exemple, Local Binary Pattern (LBP), Gabor images, etc.).
Le deuxième module est une réduction de dimensionnalité simple qui peut être réalisée par PCA (Principal Component Analysis). C’est une technique

purement non surpervisée. Le troisième module est donné par la nouvelle méthode Sp-LsLPP qui est basée sur une méthode pour l'estimation d' une matrice d'affinité. Cette dernière estimation s'appuie sur la distance l1 (représentation sparse) qui incorpore naturellement les poses correspondantes à chaque image. Le quatrième module applique la discrimination entre les classes (poses). Cette technique est inspirée par LDE (Linear Discriminant embedding), qui a comme but de rapprocher la représentation des visages ayant des poses identiques tout en éloignant la représentation des visages avec des poses différents. Le module final fournit une régression sur les données projetées afin de prédire la pose.

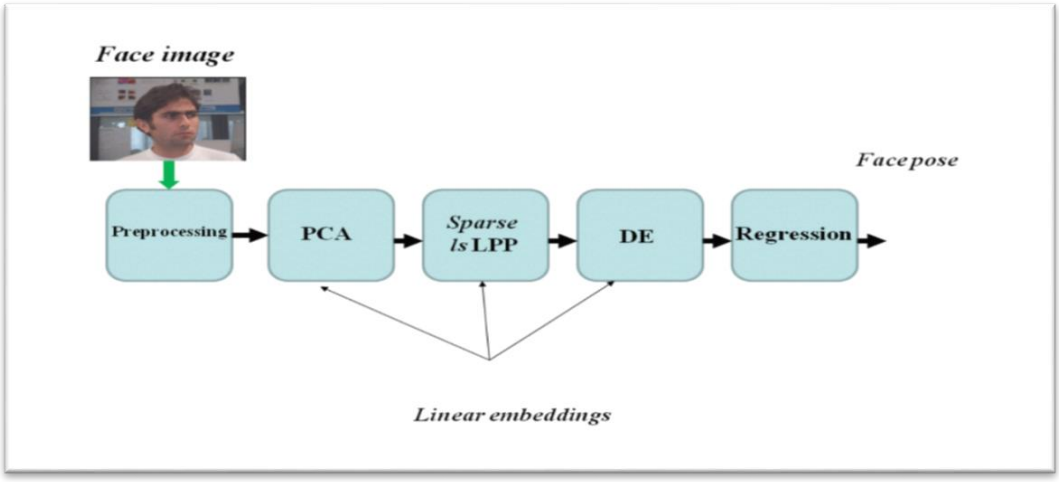


Figure 2: Méthode d'apprentissage proposée

Sparse Ls-LPP:

Bien que la méthode LsLPP (Label sensitive Locality Preserving Projection) [1] puisse intégrer les concepts de localité et de la sensibilité à l'étiquette, elle comporte quatre paramètres. Deux paramètres sont liés à la similarité entre les caractéristiques extraites et ils sont très souvent difficiles à fixer par avance. Afin de réduire le nombre total de paramètres qui doivent être réglés dans l'algorithme LsLPP et

pour modéliser mieux le rapport entre les échantillons, nous proposons une nouvelle méthode pour construire la matrice d'affinité qui utilise un codage sparse. La différence principale de l'approche proposée avec la méthode LsLPP est que la construction de la matrice de similarité de graphe sera effectuée par un codage sparse pondéré qui intègre la sensibilité de l'étiquette. La représentation sparse est, en général, une représentation d'images qui est la plus compacte en termes de la combinaison linéaire d'atomes pris en forme d'un dictionnaire trop complet [2]. Un dictionnaire est formé par de nombreuses images. Une représentation sparse signifie que seules quelques images participeront au processus de reconstruction alors que le reste des images n'aura aucune contribution. Cette méthode est résumée dans l'algorithme suivant.

Sp-LsLPP

Input: Training set represented by the data matrix $\mathbf{X} = \{\mathbf{x}_i \in \mathbb{R}^d\}_{i=1}^N$, labels $\mathbf{y} = \{y_i\}_{i=1}^N$ representing the real-valued pose (angle) of each sample, parameters σ, and λ.

Output: A linear transform matrix $\mathbf{W}$

- For each sample $\mathbf{x}_i$:
  - Compute the diagonal matrix $\mathbf{P}$ such that $P(j,j) = 1 - exp(-(\frac{y(i)-y(j)}{\sigma})^2)$, $j = 1, ..., N$.
  - Estimate the coding vector $\mathbf{b}$ using the following weighted sparse optimization problem:

$$\min_{\mathbf{b}}(\|\mathbf{X}'\mathbf{b} - \mathbf{x}_i\|^2 + \lambda \|\mathbf{P}\,\mathbf{b}\|_1) \tag{2.2}$$

    where $\mathbf{X}'$ is a matrix formed by the remaining $N-1$ training samples. By introducing the auxiliary vector $\mathbf{a} = \mathbf{P}\,\mathbf{b}$, (2.2) becomes a classic sparse coding problem $\min_{\mathbf{a}}(\|\mathbf{X}'\mathbf{P}^{-1}\,\mathbf{a} - \mathbf{x}_i\|^2 + \lambda \|\mathbf{a}\|_1)$
  - The i^{th} row of $\mathbf{B}$ is given by $\mathbf{B}(i,:) = (\mathbf{b}_i)^T = (\mathbf{P}^{-1}\mathbf{a})^T$
- Make the affinity matrix symmetric, i.e., $\mathbf{B} \leftarrow |\mathbf{B}| + |\mathbf{B}|^T$
- Compute the Laplacian $\mathbf{L} = \mathbf{D} - \mathbf{B}$ where $\mathbf{D}$ is a diagonal matrix with $D_{ii} = \sum_j B_{ij}$
- Solve the following generalized eigenvector problem: $(\mathbf{X}\mathbf{L}\mathbf{X}^T)\mathbf{v}_i = \lambda_i(\mathbf{X}\mathbf{D}\mathbf{X}^T)\mathbf{v}_i \longrightarrow (\mathbf{X}\mathbf{L}\mathbf{X}^T)\mathbf{V} = (\mathbf{X}\mathbf{D}\mathbf{X}^T)\mathbf{V}\Lambda$
- Output: Linear transform $\mathbf{W} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_p] \in \mathbb{R}^{d\times p}$

Résultats et Conclusion:

Les expériences sont effectuées sur trois bases de données dont uniquement l'angle de lacet varie. La première base de données est FacePix. Elle contient 181 images faciales pour chacune des 30 personnes où chaque image correspond à un intervalle de rotation de 1 degré sur un spectre de 180 degrés. La deuxième base de données est celle de Taiwan qui contient des images de 90 personnes à un intervalle de 5 degré. La troisième base de données est celle de Columbia. Il s'agit d'une base de données à l'origine pour évaluer l'estimation de l'œil. Cette base de données contient également des images représentant des poses de visage de personnes différentes. Elle contient des images haute résolution de 56 personnes différentes avec 5 poses pour chacun (0, -15,+15, -30, +30).

Les résultats présentés dans le tableau ci-dessous montrent que notre méthode surpasse en générale les méthodes de l'état de l'art existantes.
L'approche proposée présente encore l'avantage d'avoir un nombre réduit de paramètres. Notre travail dans le futur va explorer l'efficacité de l'utilisation de descripteurs d'images hybrides.

Tableau 1: Résultats expérimentales (Mean Average Error (MAE) en degré et le Standard deviation) sur la base de données Facepix

Method (Raw images)	Split1	Split2	Split3	Split4	Split5	Mean
S-LE	$4.50_{\pm 4.8}$	$4.50_{\pm 4.8}$	$4.33_{\pm 4.8}$	$4.80_{\pm 6.8}$	$\mathbf{1.27}_{\pm 6.5}$	$3.88_{\pm 5.5}$
LPP	$4.93_{\pm 5.1}$	$4.80_{\pm 4.9}$	$6.80_{\pm 7.1}$	$4.87_{\pm 5.2}$	$5.82_{\pm 6.2}$	$5.44_{\pm 5.7}$
LsLPP	$3.00_{\pm 2.9}$	$\mathbf{3.9}_{\pm 4.6}$	$3.34_{\pm 3.0}$	$3.84_{\pm 5.2}$	$4.16_{\pm 4.7}$	$3.65_{\pm 4.1}$
Sp-LsLPP	$\mathbf{2.59}_{\pm 3.1}$	$4.03_{\pm 4.9}$	$3.23_{\pm 3.2}$	$4.27_{\pm 5.8}$	$4.89_{\pm 6.1}$	$3.80_{\pm 4.6}$
SSE	$4.45_{\pm 4.1}$	$4.77_{\pm 5.8}$	$4.077_{\pm 3.9}$	$4.28_{\pm 5.6}$	$4.82_{\pm 6.9}$	$4.47_{\pm 5.3}$
LsLPP+SSE	$4.45_{\pm 4.1}$	$4.77_{\pm 5.8}$	$4.077_{\pm 3.9}$	$4.28_{\pm 5.6}$	$4.82_{\pm 6.9}$	$4.47_{\pm 5.3}$
Sp-LsLPP + DE	$2.84_{\pm 2.9}$	$4.09_{\pm 5.0}$	$\mathbf{3.16}_{\pm 2.9}$	$\mathbf{3.4}_{\pm 5.00}$	$3.85_{\pm 4.6}$	$\mathbf{3.48}_{\pm 4.1}$
Method (LBP descriptors)	Split1	Split2	Split3	Split4	Split5	Mean
One block	$4.93_{\pm 7.7}$	$9.00_{\pm 16.4}$	$5.40_{\pm 8.2}$	$5.81_{\pm 9.3}$	$6.01_{\pm 9.0}$	$6.23_{\pm 10.1}$
25 blocks	$4.17_{\pm 5.5}$	$4.53_{\pm 5.9}$	$4.77_{\pm 6.4}$	$4.42_{\pm 6.1}$	$5.08_{\pm 7.2}$	$4.59_{\pm 6.2}$
25 b: Sp-LsLPP	$3.48_{\pm 4.3}$	$3.94_{\pm 4.8}$	$6.09_{\pm 8.6}$	$4.58_{\pm 6.5}$	$3.96_{\pm 5.2}$	$4.41_{\pm 5.9}$
25 b: Sp-LsLPP + DE	$\mathbf{2.44}_{\pm 3.0}$	$\mathbf{2.98}_{\pm 3.8}$	$\mathbf{3.32}_{\pm 4.3}$	$\mathbf{3.03}_{\pm 3.9}$	$\mathbf{2.95}_{\pm 4.1}$	$\mathbf{2.94}_{\pm 3.8}$

Tableau 2: Résultats expérimentales (Mean Average Error (MAE) en degré et le Standard deviation) sur la base de données Taiwan

Method (Raw images)	Split1	Split2	Split3	Split4	Split5	Mean
S-LE	$7.34_{\pm 9.7}$	$\mathbf{4.65}_{\pm 4.8}$	$\mathbf{4.55}_{\pm 4.8}$	$5.11_{\pm 7.0}$	$\mathbf{4.57}_{\pm 4.9}$	$5.24_{\pm 6.2}$
LPP	$6.14_{\pm 4.8}$	$5.82_{\pm 4.7}$	$5.50_{\pm 4.7}$	$5.43_{\pm 4.6}$	$5.40_{\pm 4.6}$	$5.65_{\pm 5.7}$
LsLPP	$5.66_{\pm 4.8}$	$5.38_{\pm 4.3}$	$5.07_{\pm 4.3}$	$\mathbf{5.05}_{\pm 4.2}$	$4.84_{\pm 4.1}$	$5.20_{\pm 4.3}$
Sp-LsLPP	$5.72_{\pm 7.4}$	$5.07_{\pm 6.6}$	$5.93_{\pm 7.5}$	$6.01_{\pm 7.8}$	$5.37_{\pm 6.9}$	$5.62_{\pm 7.2}$
SSE	$5.63_{\pm 7.2}$	$5.35_{\pm 6.7}$	$5.50_{\pm 6.9}$	$5.39_{\pm 6.8}$	$5.47_{\pm 7.0}$	$5.46_{\pm 6.9}$
LsLPP+SSE	$5.63_{\pm 7.2}$	$5.35_{\pm 6.7}$	$5.50_{\pm 6.9}$	$5.39_{\pm 6.8}$	$5.47_{\pm 7.0}$	$5.46_{\pm 6.9}$
Sp-LsLPP + DE	$\mathbf{5.48}_{\pm 7.1}$	$5.34_{\pm 6.7}$	$4.96_{\pm 6.1}$	$5.12_{\pm 6.6}$	$4.97_{\pm 6.4}$	$\mathbf{5.17}_{\pm 6.5}$
Method (LBP descriptors)	Split1	Split2	Split3	Split4	Split5	Mean
One block	$10.89_{\pm 14.8}$	$12.18_{\pm 18.0}$	$11.42_{\pm 15.5}$	$11.46_{\pm 18.5}$	$10.79_{\pm 15.2}$	$11.34_{\pm 16.4}$
25 blocks	$6.89_{\pm 9.1}$	$7.16_{\pm 9.5}$	$7.10_{\pm 8.7}$	$7.37_{\pm 9.8}$	$7.13_{\pm 9.6}$	$7.13_{\pm 9.3}$
25 b: Sp-LsLPP	$6.85_{\pm 8.6}$	$6.12_{\pm 7.8}$	$6.77_{\pm 7.8}$	$6.96_{\pm 9.1}$	$7.12_{\pm 9.1}$	$6.76_{\pm 8.5}$
25 b: Sp-LsLPP + DE	$\mathbf{5.46}_{\pm 7.2}$	$\mathbf{4.73}_{\pm 6.3}$	$\mathbf{4.74}_{\pm 5.8}$	$\mathbf{4.93}_{\pm 6.5}$	$\mathbf{4.96}_{\pm 6.5}$	$\mathbf{4.96}_{\pm 6.5}$

Tableau 3: Résultats expérimentales (Mean Average Error (MAE) en degré et le Standard deviation) sur la base de données Columbia

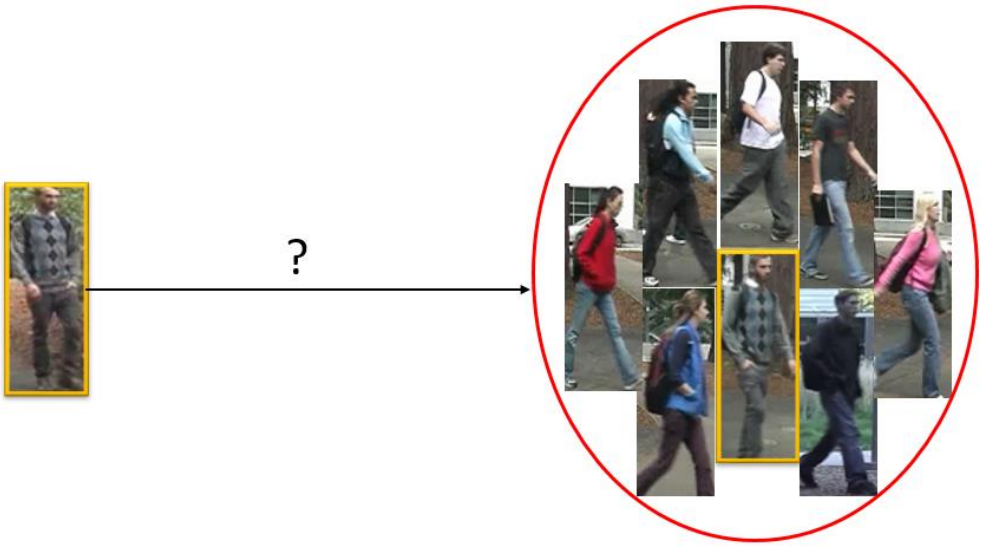
30% training and 70% testing

Method (Raw images)	Split1	Split2	Split3	Split4	Split5	Mean
LPP	89.7	**89.7**	**89.2**	**89.7**	**89.7**	**89.6**
LsLPP	**90.2**	88.2	88.2	91.8	87.2	89.1
Sp-LsLPP	88.7	86.7	86.1	89.7	84.6	87.2

90% training and 10% testing

Method (Raw images)	Split1	Split2	Split3	Split4	Split5	Mean
LPP	**96.0**	92.0	96.0	92.0	88.0	92.8
LsLPP	92.0	92.0	96.0	96.0	96.0	**94.4**
Sp-LsLPP	92.0	**92.0**	**96.0**	**96.0**	**96.0**	**94.4**

Chapitre3 :



La ré-identification d'une personne, une tâche centrale dans de nombreux scénarios de surveillance, est la capacité d'associer une nouvelle observation d'une personne à d'autres personnes ailleurs. En d'autres termes, c'est la re-identification d'une personne dans des lieux différents dans un réseau de caméras non chevauchantes, permettant de traquer une personne en le suivant dans toute la zone surveillée. Les méthodes de ré-identification des personnes dans l'état de l'art sont principalement basées sur l'apparence globale du vêtement et du corps. La reconnaissance faciale, qui est potentiellement plus efficace, n'est souvent pas pratique dans ce contexte en raison des faibles résolutions des images, de la présence d'occlusions avec des objets et des fortes variations d'éclairage. Ré-identifier les personnes qui se trouvent dans une région d'intérêt est une capacité de haut niveau qui est critique dans de nombreux domaines autre que la vidéo surveillance, comme la robotique et les environnements intelligents.

Dans cette thèse, nous proposons une nouvelle méthode de catégorisation des couleurs (Figure 3), qui se base principalement sur la méthode PLSA proposée dans [3]. PLSA est efficace si on travaille dans les mêmes conditions de lumière, mais il échoue lorsqu'on travaille avec deux caméras différentes dans deux conditions de lumière différentes, comme notre cas. Ainsi, pour augmenter la robustesse contre le changement d'éclairage, nous commençons par adopter une nouvelle palette de couleurs. Quelques couleurs sont

visuellement similaires et peuvent être fusionnés pour réduire davantage le nombre de couleurs. Dans notre travail, le gris est considéré comme blanc. L'orange, le rose et le violet sont considéré comme rouge. Le marron est considéré comme noir. Nous avons testé sur diverses images, et on a constaté que l'adoption de 6 couleurs (noir, blanc, rouge, vert, bleu et Jaune) au lieu de 11 serait approprié pour la plupart des cas.

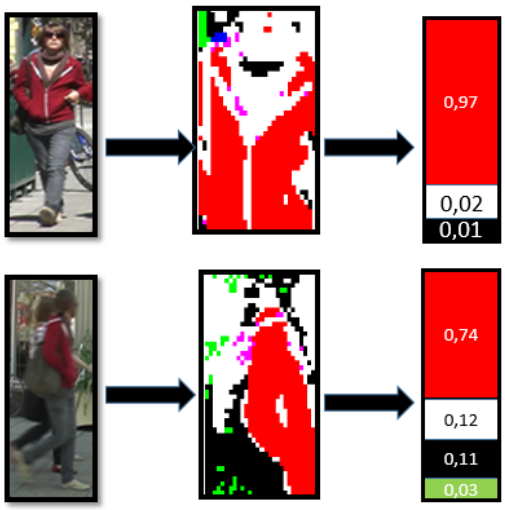


Figure 3: Color Categorisation

Chaque couleur peut être lié aux coordonnées X, Y, Z correspondant aux valeurs normalisées R, G, B. Il existe plusieurs bases de données de couleurs open source disponible en ligne. Dans notre travail, nous employons la base de données définie en [4]. Chacun des canaux R, G et B est divisé en 52 niveaux. Les coordonnées de chaque pixel sont traitées comme l'indice de la table de recherche. Nous attribuons la valeur 0 pour tous les pixels qui n'appartiennent pas à la catégorie de couleurs en question, et 1 autrement. Pour prédire l'appartenance à la couleur d'un nouveau pixel, nous interpolons le nouveau triplet du pixel avec la table de recherche de chacune des couleurs.

La deuxième étape dans notre approche est l'extraction du descripteur des images brutes. La plupart des méthodes existantes sont basées sur l'extraction des canaux de couleurs individuellement (par exemple RGB ou HSV) négligeant la relation entre chacun de ces canaux de couleur. LBP a déjà prouvé son efficacité dans la description de l'information locale de l'image. Cependant, effectuer LBP sur chaque canal de couleur à part ignorent la correlation entre les couleurs. QLBP [5] profite à la fois de LBP pour extraire

les caractéristiques et des quaternions pour représenter chaque pixel de couleur afin que nous puissions gérer tous les composants de couleur une seule foie. L'extraction de caractéristiques basée sur QLBP est présentée par l'algorithme suivant :

Algorithm 1 Feature descriptor

Input: Image I
N: number of unit quaternions
A set of unit quaternions :$[p_1, p_2, ..., p_N]$

Output: Feature vector F

Procedure:
Find the quaternionic representation Q_I of image I

For i = 1 to N, do:
Calculate the phases of the rotated quaternionic image $PRQ_r(Q_I, p_i)$.
Extract the QLBP using
Divide the QLBP image into 8x16 subimages overlapped by half.
Calculate the histogram of each subimage.
Concatenate all histograms to obtain the vector F_i.
end
$F = [F_1, F_2, ..., F_N]$

En plus, nous explorons l'adaptation du concept formation prototype dans le problème de ré-identification de la personne. Il a été proposé en psychologie et en cognition [6] et testé sur le problème de la reconnaissance faciale [7]. Il suggère que les humains classent les objets en fonction des prototypes hiérarchiques et les personnes différentient le monde en utilisant cette compétence critique pour l'apprentissage par catégorie.
Des expériences psychologiques ont révélé que le cerveau humain reconnaît et différentie les objets en utilisant des prototypes. Cela signifie que les prototypes fournissent une mesure pour reconnaître ou classer un nouvel objet. En se basant sur ça, nous proposons une approche pour la ré-identification de personne dans laquelle chaque personne est décrite comme un vecteur de similarité à une collection de prototypes d'images. Ce concept est illustré dans la Figure 4.

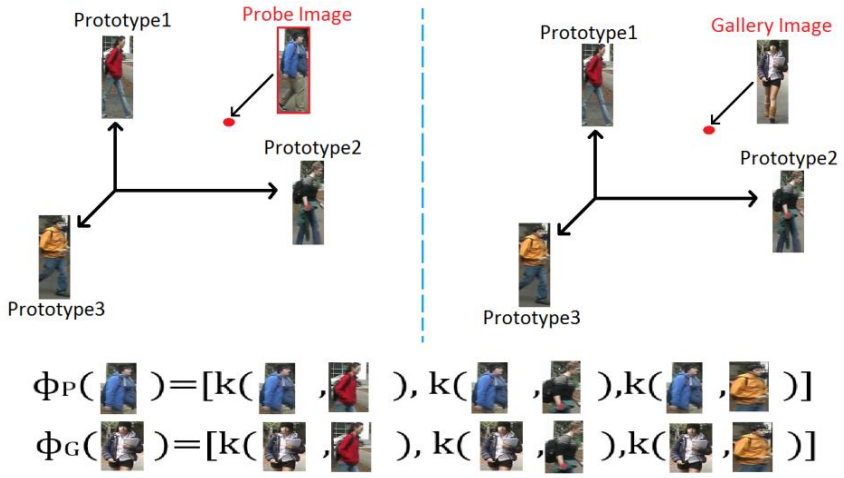


Figure 4: Chaque image sera représenté par un vecteur de similarité entre lui meme et une collection d'images prototypes

Résultats expérimentaux :

Les expériences sont effectuées sur les images de la base de données VIPeR, l'une des plus difficiles bases de données pour la ré-identification de personne. Cet ensemble de données contient une quantité significative de changements de point de vue (0, 45, 90, 135 et 180 degrés), variations d'éclairage et des occlusions entre les personnes. Il est conçu pour un scénario single-shot et il contient des paires d'images de 632 personnes normalisées à 48 x 128 pixels. Beaucoup d'images contiennent un changement de point de vue de 90 degrés et un changement d'illumination fort, ce qui rend ce problème très difficile.

La figure 5 montre les courbes CMC de différentes combinaisons utilisant la base de données VIPeR. Les expériences sont répétées 5 fois utilisant 5 split aléatoires et les résultats sont rapportés en utilisant leurs valeurs moyennes. Les méthodes sans catégorisation des couleurs signifie que le score de similitude a été calculé sans tenant compte de la catégorie de couleurs de chaque personne. Le NN (le plus proche voisin) signifie que pour la classification, une simple distance de la norme L2 a été calculée.

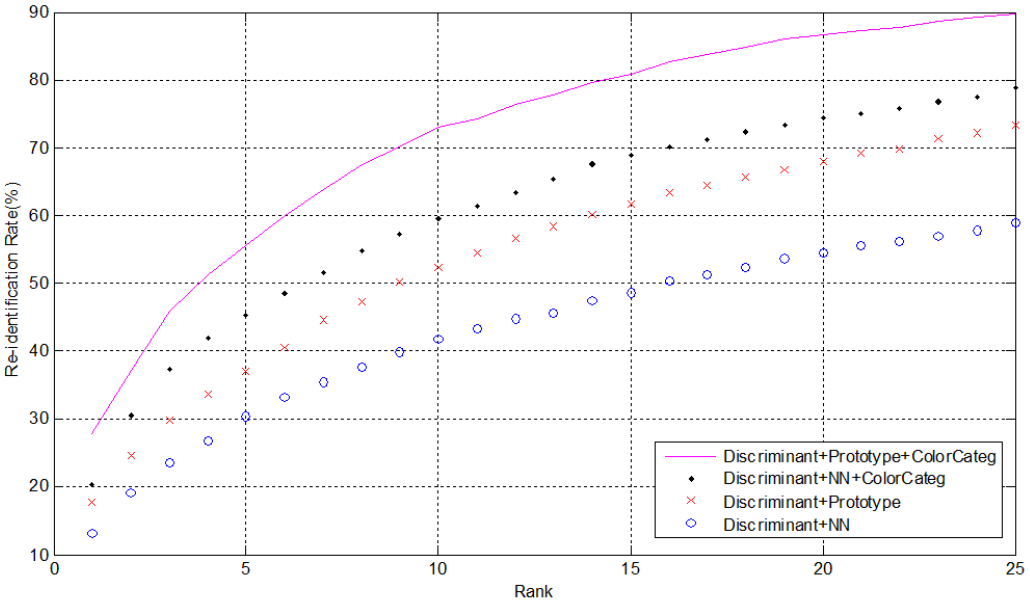


Figure 5: CMC pour différentes combinaisons

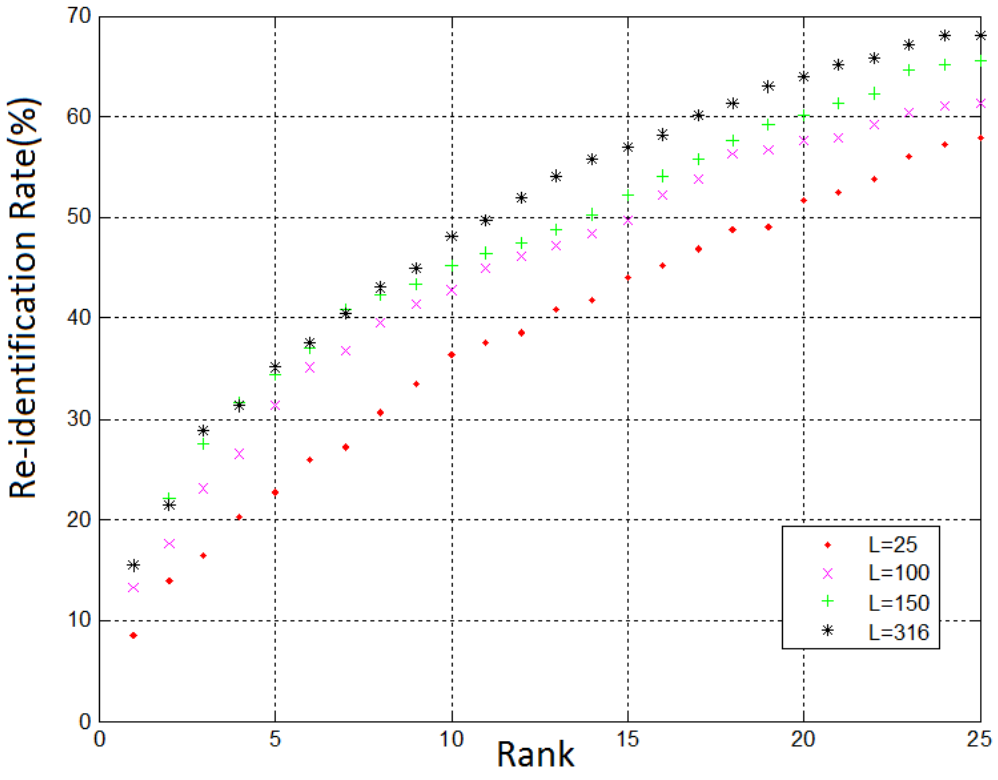


Figure 6: CMC pour différents nombres de prototypes

Tableau 2: Matching rate pour differents méthodes

Method	Rank1 (%)	Rank20 (%)
PreidPFCC	28	87
ELF	12	61
SDALF	19.87	65.73
PRDC	15.66	70.09
RankBoost	23.92	68.73
KISSME	20	76

Le tableau 2 compare les résultats obtenus en utilisant notre méthode proposée PreidPFCC (Réidentification de personne par formation de prototype et catégorisation des couleurs) et certaines des approches existantes, y compris ELF (Ensemble of Localized Caractéristiques) [8], SDALF (accumulation par symétrie des fonctionnalités locales) [9], PRDC (Comparaison de distance relative probabiliste) [10], RankBoost [11] et KISSME (Keep It Simple and Straightforward Metric) [12]. La performance est présentée par la précision rank1 et rank20. Toutes les méthodes ont utilisé 316 personnes pour les tests sur l'ensemble de données VIPeR. Les résultats montrent que notre méthode surpasse toutes les autres approches, avec le rang 1 correspondant à un taux de 28% et rang 20 de 87%. Le rang 1 a augmenté de 7% lors de l'utilisation de la catégorisation des couleurs avec la méthode du plus proche voisin (0,20 au lieu de 0,13) et 10% lors de l'utilisation de la formation de prototype en tant que processus de ré-identification (0,27 au lieu de 0,20). Les résultats montrent que la formation de prototype et la catégorisation des couleurs contribuent à l'amélioration de la performance globale.

Chapitre4 :

Dans l'apprentissage par machine conventionnel, la performance d'un modèle est normalement très affectée par la manière dont les données sont représentées. Ainsi, ces méthodes étaient limitées dans leur capacité de traiter des images brutes. Comme les chapitres 2 et 3 le démontrent, il est évident que la construction d'un système de reconnaissance en s'appuyant sur des caractéristiques artisanales exige une expertise considérable dans le domaine pour la représentation des données. Ces caractéristiques essayent de transformer les images brutes en une représentation appropriée qui est suffisamment robuste.

Alors que les caractéristiques artisanales (Handcrafted) sont certainement une façon d'aborder ce problème pour la représentation des données, dans de nombreux cas, ces caractéristiques deviennent de plus en plus complexes, ce qui se traduit par une difficulté à améliorer de plus les fonctionnalités. Pendant ce temps, certains chercheurs se sont concentrés sur le développement d'algorithmes qui intègrent l'apprentissage automatique des caractéristiques à partir d'images brutes. Ces modèles introduisent une autre façon pour représenter les données en utilisant plusieurs couches de non-linéarité. En outre, ces modèles génèrent des fonctionnalités plus sophistiquées que ce qui serait possible grâce à l'ingénierie à la main. Cette propriété a été considérée comme très importante et cela a conduit au développement des premiers modèles d'apprentissage profond.

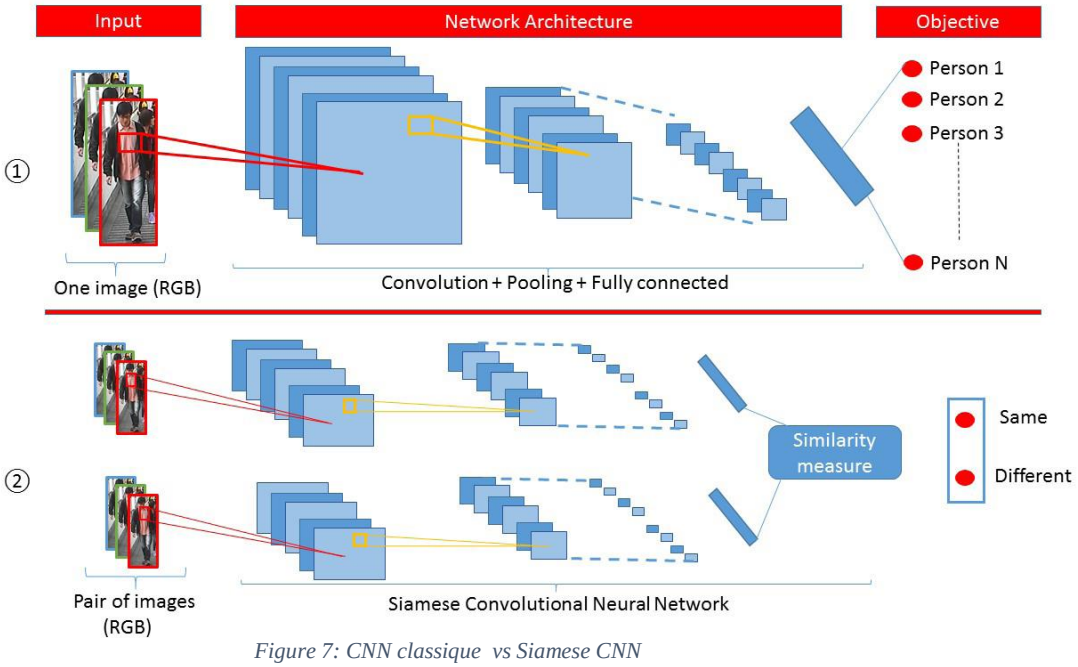


Figure 7: CNN classique vs Siamese CNN

La figure 7 illustre 2 approches différentes pour traiter le problème de ré-identification: le CNN classique (image 1) et le CNN siamois (image 2). le SCNN (Siamese Convolutional Neural Network) est une méthode qui peut être utilisée dans les cas où le nombre d'échantillons par catégorie n'est pas grand. Les méthodes traditionnelles de classification ne conviennent pas, en général, pour les cas où le nombre d'étiquettes est très grand et en même temps, le nombre d'échantillons par étiquette est très faible. C'est le cas par exemple de la ré-identification de personne où le nombre de personnes peut être des centaines avec seulement quelques images pour chaque personne.

L'architecture du réseau siamois est détaillée dans la figure 8. L'idée principale est d'appliquer le même CNN en utilisant les mêmes paramètres pour chacune des images qui doivent être testées pour la correspondance. Dans la phase d'apprentissage, une fonction objective est optimisée après le calcul de la norme L2 des différences entre les descripteurs résultants. Les paramètres sont mis à jour afin que la distance L2 soit aussi discriminante que possible pour différentier entre couples homologues et couples différents. En conséquence, le descripteur obtenu est plus tolérant au genre de distorsions géométriques présentes dans des exemples de couple homologue.

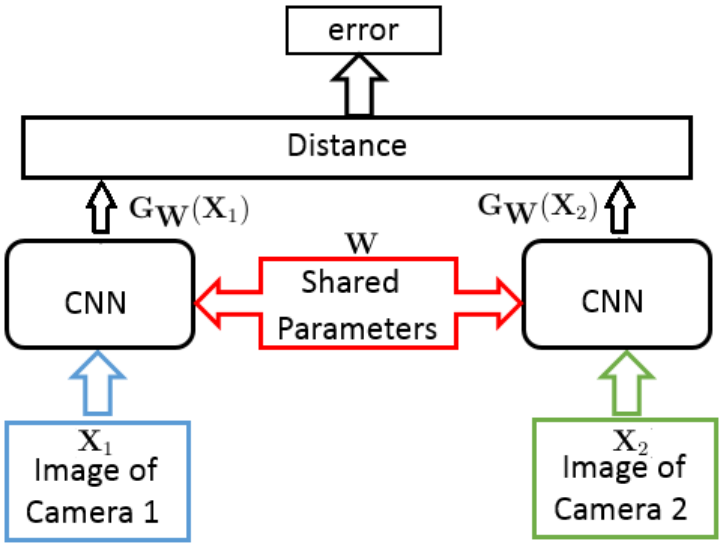


Figure 8: Architecture du Siamese CNN

Tableau 3: Architecture du CNN utilisé

Name	Type	Number	Filter size	Stride	Activation function
Conv0	Convolution	16	3x3	1	Rectify
Conv1	Convolution	32	3x3	1	Rectify
Pool0	Max pooling		2x2	2	
Conv2	Convolution	32	3x3	1	Rectify
Conv3	Convolution	64	3x3	1	Rectify
Pool1	Max pooling		2x2	2	
Conv4	Convolution	64	3x3	1	Rectify
Conv5	Convolution	128	3x3	1	Rectify
Pool2	Max pooling		2x2	2	
FC1	Fully connected	1024			Rectify
FC2	Fully connected	100			Rectify

Comme on peut le voir dans le tableau 3, notre CNN se compose de 6 couches de convolution, 3 couches de pooling et deux couches complètement connectées (Fully connected). La sortie du CNN comporte 500 dimensions. Une paire d'images est filtrée par la pile de couches de convolution avec un très petit champ réceptif: 3x3. Le stride de la convolution est réglé sur un pixel et le pooling s'effectue à travers des couches de maxpool réalisées sur une fenêtre de 2x2 pixel avec un stride égal à deux. Après une pile de couches de convolution, on met deux couches totalement connectées où le premier a une dimension 1024 et le second a une dimension de 500. Le neurone ReLU est utilisé comme fonction d'activation pour chaque couche.

Résultats et conclusion :

L'ensemble de base de données utilisé dans ce travail est le CHUK01 publié dans [13]. Dans cette base, il y a 971 identités et chaque identité n'a que deux images dans chaque caméra. Une centaine de personnes sont utilisées pour les tests et les 871 restants sont utilisées pour l'apprentissage, conformément au travail fait en FPNN [14].
La courbe CMC a été utilisée pour évaluer la performance (Figure 9). Le tableau 4 compare les résultats de notre modèle avec certains des modèles d'état de l'art. Les résultats montrent que notre méthode est supérieure à toutes les autres approches, avec le rang1 de 31% et taux d'appariement rang25 de 90%.

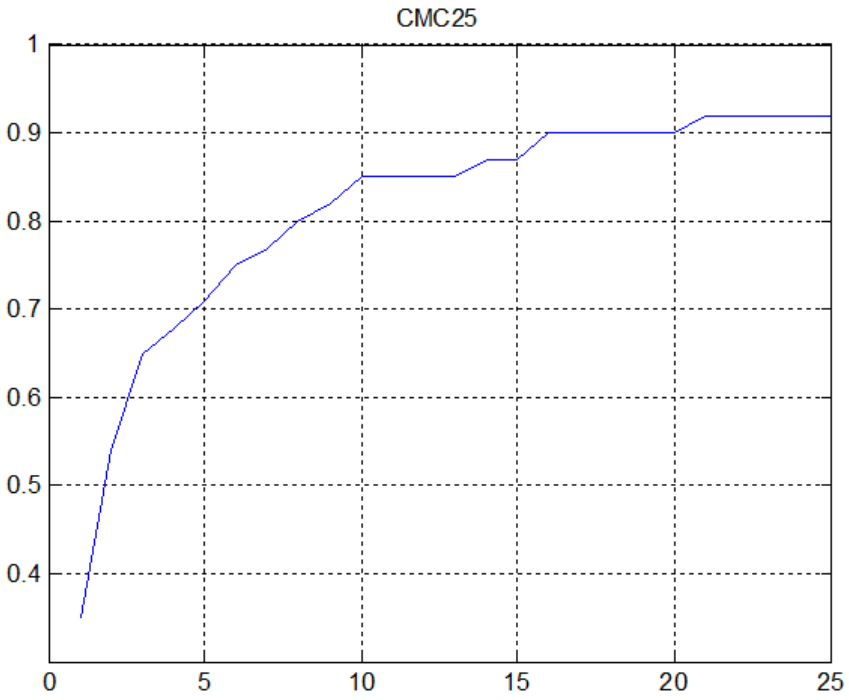


Figure 9: CMC pour differentes stratégies

Tableau 4: Rank1, Rank5 and Rank10 de différentes méthodes

Method	Rank1	Rank5	Rank25
Our method	35%	71%	92%
ITML	17.10%	42%	76%
LMNN	21.17%	49%	83%
SDALF	9.9%	42%	70%
FPNN	27.87%	60%	90%
Euclid	10.52%	28%	60%

Afin de comprendre la contribution de chaque partie du corps seule, nous avons construit 4 réseaux différents sur différentes parties du corps. Un pour la tête seule, un pour le torse, un pour les jambes et un pour tous le corps. La figure 10 montre les courbes CMC pour les trois différentes parties et pour la personne complète. La partie qui performe le mieux sur rang1 est la torse, bien qu'au rang25, il devient claire que la partie la plus discriminative est la tête.

Cette analyse révèle une direction pour des expériences dans le futur dans lesquelles plusieurs réseaux sont formés et la décision finale dépend des scores d'accumulation de tous les réseaux. Cela peut être très avantageux dans le contrôle des situations où des occlusions sévères se produisent. La différence de performance commence à être claire à partir du rang 12. On peut remarquer que l'ordre de performance après ce rang va à la tête puis aux jambes ou au torse avec des performances très proches. Au premier rang, le torse donne les meilleurs résultats, ce qui est conforme à notre intuition. Généralement, le corps est la partie la plus stable de l'image de la personne et prendre le risque de ré-identifier une personne de la première tentative serait raisonnable si elle était basée sur les vêtements du haut du corps.

Comme indiqué précédemment, les méthodes d'extraction des caractéristiques peuvent être pratiquement divisées en deux groupes: les caractéristiques artisanales (Handcrafted features) et les caractéristiques apprises (Learned features). Par les caractéristiques artisanales on veut dire celles qui sont extraites suite à une attention particulière à travers des algorithmes conçus à la main comme ceux utilisés dans le chapitre 3. Contrairement aux caractéristiques artisanales, les caractéristiques apprises sont obtenus grâce à des procédures d'apprentissage en entrainent un réseau avec une base de données bien définie. Afin de comparer les performances de ces deux approches, nous avons appliqué le même algorithme présenté au chapitre 3 sur la base de données CHUK01 en utilisant le scenario single-shot. Les résultats comparatifs sont présentés dans la figure 10. On constate que les caractéristiques apprises obtiennent les meilleurs résultats atteignant 31% sur rank1, tandis que les caractéristiques artisanales ont atteint 22%. Ces résultats sont parfaitement logique, car les caractéristiques apprises ont la capacité de s'adapter à la l'exclusivité de l'application alors que les caractéristiques artisanales ne sont pas entrainés et, par conséquent, elles sont moins flexibles.

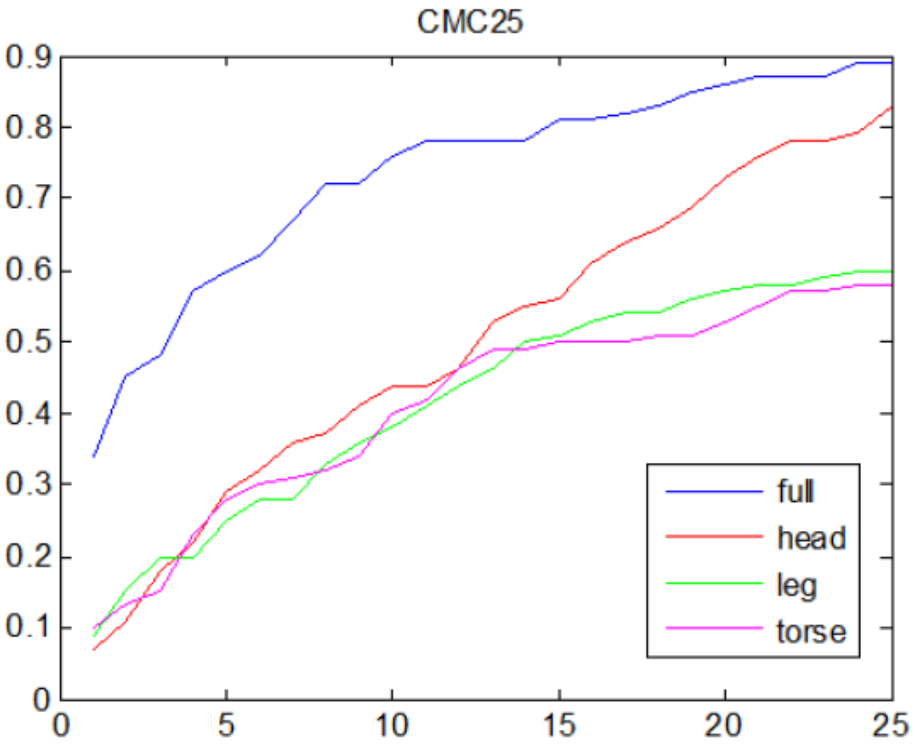


Figure 10: Résultats sur différentes parties du corps

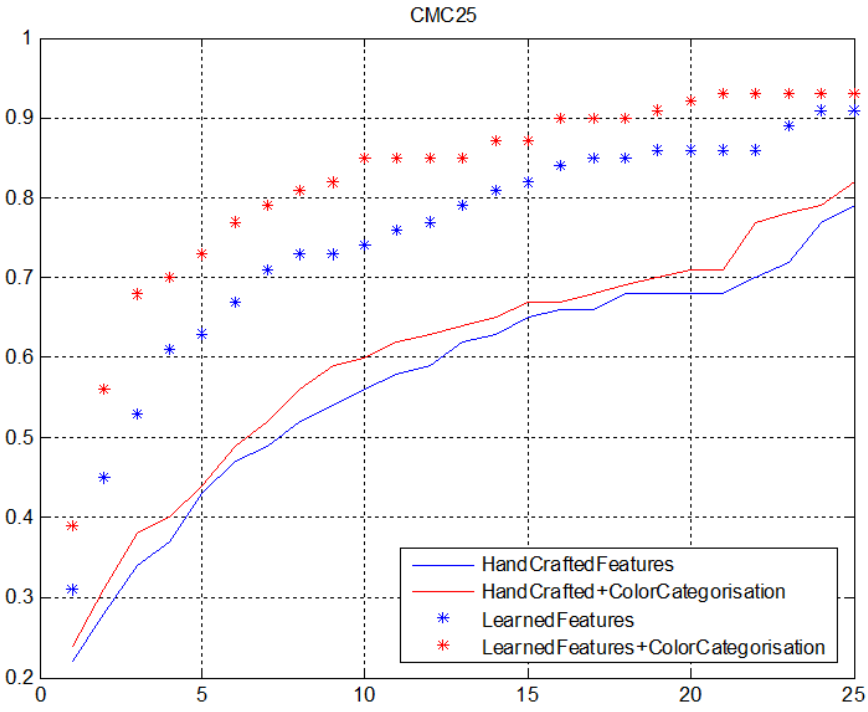


Figure 11: Handcrafted vs Learned features

Bibliographie

[1] Wei-Lun Chao, Jun-Zuo Liu, and Jian-Jiun Ding. Facial age estimation based on label-sensitive learning and age-oriented regression. Pattern Recogn., 46(3):628-641, March 2013.

[2] Ke Huang and Selin Aviyente. Sparse representation for signal classification. In Bernhard Scholkopf, John C. Platt, and Thomas Hoffman,editors, Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems,Vancouver, British Columbia, Canada,pages 609-616. MIT Press, December 2006.

[3]J. van de Weijer, C. Schmid, J. Verbeek, and D. Larlus. Learning color names for real-world applications. IEEE Transactions in Image Processing, 2009.

[4] D. John. Color name identification: fuzzycolor, matlab central file exchange. retrieved 20 sep 2006.

[5] R. Lan, Y. Zhou, Y Yan Tang, Chen, and C. Chen. Person reidentification using quaternionic local binary pattern. IEEE International Conference on Multimedia and Expo , 1-6, July 2014.

[6] E. Rosch. Natural categories. Cognitive psychology, 4(3):328-350, 1973.

[7] B. Klare and A. Jain. Heterogeneous face recognition using kernel prototype similarities. IEEE TPAMI, 35(6): 1410-1422, June 2013.

[8] D. Gray and H. Tao. Viewpoint invariant pedestrian recognition with an ensemble of localized features. In ECCV, 2008.

[9] M. Farenzena, L. Bazzani, A. Perina, V. Murino, and M. Cristani. Person re-identification by symmetry-driven accumulation of local features. CVPR 2010 IEEE Conference on San Francisco 2360-2367, 2010.

[10] Wei-Shi Zheng, Shaogang Gong, and Tao Xiang. Reidentification by relative distance comparison. Pattern Analysis and Machine Intelligence, IEEE Transactions on , 35(3):653-668, 2013.

[11] Cheng-Hao Kuo, Khamis, and S. Shet. Person re-identification using semantic color names and rankboost. Applications of Computer Vision (WACV), 2013 IEEE Workshop on , Tampa, FL ,281-287, 2013.

[12] M. Kostinger, M. Hirzer, P. Wohlhart, P. Roth, and H. Bischof. Large

scale metric learning from equivalence constraints. Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference, 2288-2295, June 2012.

[13] Wei Li, Rui Zhao, and Xiaogang Wang. Human reidentification with transferred metric learning. In Asian Conference on Computer Vision, pages 31-44. Springer, 2012.

[14] Wei Li, Rui Zhao, Tong Xiao, and Xiaogang Wang. Deepreid: Deep filter pairing neural network for person re-identification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 152-159.

Charbel CHAHLA

Doctorat : Optimisation et Sûreté des Systèmes

Année 2017

Extraction de descripteurs non linéaires pour la ré-identification d'objets dans un réseau de caméras

La réplication du système visuel utilisé par le cerveau pour traiter l'information est un domaine de grand intérêt. Cette thèse se situe dans le cadre d'un système automatisé capable d'analyser les traits du visage lorsqu'une personne est proche des caméras et suivre son identité lorsque ces traits ne sont plus traçables. La première partie est consacrée aux procédures d'estimation de pose de visage pour les utiliser dans les scénarios de reconnaissance faciale. Nous avons proposé une nouvelle méthode basée sur une représentation *sparse* et on l'a appelé *Sparse Label sensible Local Preserving Projections*. Dans un environnement incontrôlé, la ré-identification de personne reposant sur des données biométriques n'est pas réalisable. Par contre, les caractéristiques basées sur l'apparence des personnes peuvent être exploitées plus efficacement. Dans ce contexte, nous proposons une nouvelle approche pour la ré-identification dans un réseau de caméras non chevauchantes. Pour fournir une mesure de similarité, chaque image est décrite par un vecteur de similarité avec une collection de prototypes. La robustesse de l'algorithme est améliorée en proposant la procédure *Color Categorisation*. Dans la dernière partie de cette thèse, nous proposons une architecture *Siamese* de deux réseaux neuronaux convolutionnels (CNN), chaque CNN étant réduit à seulement onze couches. Cette architecture permet à une machine d'être alimentée directement avec des données brutes pour faire la classification.

Mots clés : vision par ordinateur - vidéosurveillance - perception des visages - apprentissage supervisé (intelligence artificielle) - réseaux neuronaux (informatique) - identification des personnes.

Non-linear Feature Extraction for Object Re-identification in Cameras Networks

Replicating the visual system that the brain uses to process the information is an area of substantial interest. This thesis is situated in the context of a fully automated system capable of analyzing facial features when the target is near the cameras, and tracking his identity when his facial features are no more traceable. The first part of this thesis is devoted to face pose estimation procedures to be used in face recognition scenarios. We proposed a new label-sensitive embedding based on a sparse representation called Sparse Label sensitive Locality Preserving Projections. In an uncontrolled environment observed by cameras from an unknown distance, person re-identification relying upon conventional biometrics such as face recognition is not feasible. Instead, visual features based on the appearance of people can be exploited more reliably. In this context, we propose a new embedding scheme for single-shot person re-identification under non overlapping target cameras. Each person is described as a vector of kernel similarities to a collection of prototype person images. The robustness of the algorithm is improved by proposing the Color Categorization procedure. In the last part of this thesis, we propose a Siamese architecture of two Convolutional Neural Networks (CNN), with each CNN reduced to only eleven layers. This architecture allows a machine to be fed directly with raw data and to automatically discover the representations needed for classification.

Keywords: computer vision - video surveillance - face perception - supervised learning (machine learning) - neural networks (computer science) - person Identification.

Thèse réalisée en partenariat entre :

Université Libanaise

utt UNIVERSITÉ DE TECHNOLOGIE TROYES

Ecole Doctorale "Sciences et Technologies"