Face recognition in low-resolution images under small sample conditions with face-part detection and alignment

Karaaba, Mahir Faik

Publication date:
2016

FACE RECOGNITION IN LOW-RESOLUTION IMAGES UNDER SMALL SAMPLE CONDITIONS WITH FACE-PART DETECTION AND ALIGNMENT

MAHIR FAIK KARAABA

Printed by: Haveka
ISBN: printed version: 978-90-367-5322-7
digital version: 978-90-367-5322-7

university of groningen

Face Recognition In Low-Resolution Images under Small Sample Conditions with Face-Part Detection and Alignment

PhD thesis

to obtain the degree of PhD at the
University of Groningen
on the authority of the
Rector Magnificus Prof. E. Sterken
and in accordance with
the decision by the College of Deans.

This thesis will be defended in public on

Friday 30 September 2016 at 11.00 hours

by

Mahir Faik Karaaba

born on 28 May 1980
in Istanbul, Turkey

Supervisor
Prof. L.R.B. Schomaker

Co-supervisor
Dr. M.A. Wiering

Assessment Committee
Prof.dr. R.C. Veltkamp
Prof.dr. U. Halici
Prof.dr. M. Biehl

ACKNOWLEDGEMENTS

First of all, I should explain my appreciations to my supervisors Dr. Marco Wiering and Prof. Dr. Lambert Schomaker who provided their best to help me to publish good quality papers in high ranked journals and conferences. I am especially grateful for my daily supervisor, Dr. Marco Wiering's efforts for correcting my papers by which I have mastered the art of academic writing. As an expert in machine learning and computer vision, his suggestions were invaluable to me to come up with good novel methods blending my own ideas and to make me an independent thinker and researcher.

I am also thankful for Prof. Lambert Schomaker's consultancy especially for my face recognition studies. His advices kept me stay on the main road instead of being stuck in irrelevant problems which can nothing but results in wasting my valuable time.

Another person whom I should give my sincere thanks is Olarik Surinta, who always cheered me up in my difficult times and contributed to a majority of my papers in many ways. We worked together on our final papers and this sped up the way of obtaining our degrees, so I am really glad that we cooperated.

I should also thank people who helped me with the final preparations of my PhD dissertation. Burcu and Olarik, thank you both for being my paranymphs. Harmen and Jort, thank you both for being my translators from English to Dutch. Though I can write and read in Dutch, my translations could be quite unnatural and rather more difficult to correct.

Apart from help, I had a very nice time and good experiences throughout the stay of my PhD studies. I am very happy to meet the people whom I worked with in the department, in University as well as in all Groningen. My sincere apologies if I could not write your names here.

I cannot finish this letter without showing my gratitude to my parents who supported me financially and emotionally. Without their support, everything could be much more difficult and slower.

CONTENTS

ACRONYMS

BOW	Bag of Words
CNN	Convolutional Neural Network
DoG	Difference of Gaussians
Feret	Face Recognition Technology
HOG	Histogram of Oriented Gradients
k-NN	*k*-Nearest Neighbor
LFW	Labeled Faces in the Wild
MLP	Multi Layer Perceptron
MSRS	Most Similar Region Selection
MLPD	Multi-Layer Perceptron Based Distance Function
MMD	Mean of Minimum Distances
ORL	Olivetti Research Laboratory
OWR	Overlapping Windows Ratio
PCA	Principal Component Analysis
RBF	Radial Basis Function
RBM	Restricted Boltzmann Machine
SIFT	Scale Invariant Feature Transform
SVM	Support Vector Machine
IMM	Informatics and Mathematical Modelling

1 INTRODUCTION

Today we are living in a highly technological environment. We use devices which are getting smarter every day. Artificial intelligence has become one of the important technological aspects of today's high-tech world. Face recognition (Jafri and Arabnia, 2009), (Jain et al., 1999), as a biometric authentication technique (Jain and Kumar, 2012), is an important application field of artificial intelligence (Jain and Klare, 2012). Its main advantage is that, unlike other biometric techniques such as finger print (Jain and Maltoni, 2003), iris (Bowyer et al., 2013) and speaker recognition (Saquib et al., 2010), it does not require the applicant to spend time in the personal data acquisition process. For instance, facial recognition software, which is deployed in a public area where many different people pass by, can recognize faces of passers in a crowd and can help identifying a criminal (Brey, 2004). Its main disadvantage is the sensitivity to illumination variances, poses and occlusions which occur in unstructured environments.

This chapter is organized as follows: In section 1.1, an automatic face recognition system together with its preprocessing functions are discussed. In section 1.2, the goals of the research described in this dissertation are explained. In section 1.3, the contributions are given and the chapter is finally concluded with an overview of this dissertation in section 1.4.

1.1 Face Recognition Systems

In this section, typical steps of an automatic face recognition system such as localization, aligning and recognition are explained.

Face Detection, Localization and Alignment

Detection In realistic conditions, faces of people are mixed with other faces as well as with other objects in camera images. For a face recognition application to function automatically, faces should be detected and localized first to be useful for the recognition application. Face detection (Yang et al., 2002), a special case of object detection, uses a search algorithm whose goal is finding the location of a face in an image. To do this, several samples (sub-images) are cropped from the source image and analyzed by a binary classifier that decides whether an image patch contains a face. After this, the sub-images which contain a face will be returned as detected faces. A very well-known state-of-the-art face detector has been developed by Viola and Jones (2004), which uses AdaBoost as a machine learning method (Freund and Schapire, 1999) combined with Haar features. In this dissertation, we have used this face detector as well to obtain face images.

Localization The location found by the face detector can be used for face tracking as well as for face recognition purposes, both of which require a different level of accuracy. For face recognition, the location is used to align the face. This location, if accurate enough, can be used to translationally align the face. However, if the face detector is not very robust and accurate then extra facial landmark information such as the center of a face is necessary. Besides, as will be seen in the next section, to align a face rotationally usually a single location parameter is not enough.

Alignment Basically 2 main types of alignment can be defined: 2D and 3D alignment. In 2D face alignment, the main idea is capturing some important facial parts which can be used to compute the face position and rotation angle by means of landmark information. In one approach, called active shape models (ASM), many fiducial points are labeled and used to model a shape being composed of points (Cootes et al., 1995). In an updated version of this approach, called active appearance models (AAM) (Cootes et al., 1998), intensities of pixel values are also used to obtain better accuracies. The main difficulty in these methods is however the labeling effort. In another approach, fiducial features of faces are modeled as local features. To extract these points, for instance, the scale invariant feature transform (SIFT) (Lowe, 2004) is used. After these points are acquired from the training images, they are compared against points of a test input image which are also extracted by the SIFT method.

In a third approach, a few important landmarks such as the centers of eyes and mouth can be used to estimate the positional and rotational offsets of the face image. For instance Hasan and Pal (2011) use Haar-like features and the AdaBoost algorithm for detecting the eyes and mouth. As Haar features are weak features, the candidate fiducial points are filtered and the best one is selected by a heuristic rule. In (Monzo et al., 2011; Kroon et al., 2009), eyes are detected by histograms of oriented gradients (HOG) and local binary patterns (LBP) feature descriptors respectively.

In 3D face alignment, not only the appearance of a face is to be aligned but also the head pose. For instance, a profile or a half profile face cannot be geometrically aligned with a 2D alignment technique which is not able to reach the depth information. In one approach, similar to ASM and AAM for 3D, landmarks containing 3D information are used for alignment. In such a work (Xiao et al., 2004) 3D data are integrated with the 2D AAM algorithm which is named as *Combined 2D+3D*. According to their paper, 6 times more parameters needed to model a face for AAM are required to model a 3D AAM, but at the profit of a faster convergence. In (ter Haar and Veltkamp, 2008), a coarse to fine 3d model fitting approach for face identification is presented. According to this, a 3D face model is roughly adjusted to the automatically segmented 3D face scan by single or multiple face components. Using multiple face components is reported to result in better accuracy for the final face identification algorithm. In a similar but a more recent paper (Chen et al., 2012), ASM for 3D alignment is integrated with speeded-up robust features (SURF) for texture modeling. In another recent face alignment and verification method (Taigman et al., 2014), a generic morphable 3D face and a 3D affine camera model are used to warp the 2D appearances into the 3D space. This makes it possible to create unseen views of a face artificially.

Recognition

After the detection, localization and alignment steps are finished, the face image is ready for a face recognition algorithm. There are two kinds of recognition problems: face verification and face identification. While in the former, the goal is to find whether two input faces belong to the same person, in the latter given an input face image the purpose is assigning the correct identity.

After the first working face recognition system was developed by Turk and Pentland (1991), who employed the eigen faces approach, a lot of research has been carried out to handle this task. There are many different approaches and methods in the field. In the following paragraphs, some of the most recent approaches are briefly explained.

Usually aligning of faces is a preprocessing step which should be done before the recognition phase. In the aligning process sometimes some information, which could also possess some identity information, is removed due to the nature of the process. To address this problem in (Berg and Belhumeur, 2012), an identity based alignment method, also called Tom-vs-Pete classifier, is proposed. In their algorithm, aligning is done pairwise (for 2 faces) by taking into account the identity information, so that only the redundant information is kept during the alignment which results in a better accuracy.

In recent years, algorithms that utilize multi-layered neural network architectures named deep learning are beginning to become state-of-the-art for face verification. These algorithms use many layers of feature detectors which work hierarchically to obtain general features and remove the noise existing in high-dimensional image data. Deep belief networks (DBN) (Hinton et al., 2006) and especially convolutional neural networks (CNN) (Lecun et al., 1998) are attracting a lot of attention from the researchers.

In (Sun et al., 2014), multiple CNNs are trained for face identification by employing more than 10K subjects to make general face representations which are to be used for face verification. In this work they even do not use a special aligner in order to test the performance of their method in a more challenging condition. In (Taigman et al., 2014), faces are first aligned using 6 fiducial points in 2D and then frontalized using a 3D warping technique which makes use of 67 fiducial points to localize. They use a deep neural network which is composed of more than 120 million parameters without weight sharing properties. In (Reed et al., 2014), several restricted Boltzmann machines are used to disentangle factors of variations such as pose, identity and/or expression. This framework is applied to expression, digit recognition as well as face verification efficiently (30% performance boost is reported for face verification). In (Zhu et al., 2014), inspired by the primate brain, which is considered to model view and identity separately, a deep multi-view perceptron is proposed for face recognition. It contains 6 main layers of identity neurons and 3 layers of view layers. The proposed model is also able to create unseen views. In an industrial paper a naive version of CNN for face verification is built and trained (Zhou et al., 2015). In this work, according to their result the human face verification performance is surpassed with 99.5% accuracy. Although deep learning algorithms are now seen as state-of-the-art solutions to many face recognition problems, prerequisites to train a deep network are powerful computers and a large amount of training data (which should also be in high resolution) and also a long training time is not uncommon.

Usually high-dimensional data are said to contain noise which can harm the discrimination power of a classifier. However, a high-dimensional data vector is not always harmful as shown in (Chen et al., 2013). Here, a very high dimensional feature vector creation method based on LBP features is presented. The size of the vector is 100K that proves high dimensionality helps to improve performance of face recognition algorithms significantly. Here, a joint Bayesian approach is used as a classifier which is said to obtain the best results according to the paper. In (Lu and Tang, 2014), a Gaussian process and a multi-source based learning algorithm is applied to the face verification problem for the labeled faces in the Wild (LFW) dataset (Huang et al., 2007). According to their results, human performance is surpassed for the first time. That is a very remarkable result since the human face recognition ability is known to be always better than machines for decades.

Generally full face images are required to feed into a face recognition algorithm. However, in (Liao et al., 2013) face parts instead of full face images are used as data. These face patches are processed by Gabor ternary patterns (GTP) and SIFT to create feature vectors. These vectors are used to create a dictionary which becomes an input for a face recognition algorithm employing a sparse coding scheme. In (Cao et al., 2013), a simple generative Bayesian transfer learning method is developed for face verification which shows promising results. In transfer learning, the idea is using a source dataset (source-domain, usually large in number) in combination with a target dataset (target-domain, usually limited in number) to improve the performance of a classifier. Similar to (Chen et al., 2013), a very high dimensional LBP filter is used as the feature vector and the Joint Bayesian method is used as the classifier.

Sparse coding is also a well-known method applied to the face recognition problem. In the sparse coding scheme, from training data, a sparse set of vectors is extracted along with the same number of coefficients which can be used to represent any test image drawn from the same subject set from the training dataset. In (Wagner et al., 2012), a sparse representation based face recognition method, inspired by (Wright et al., 2009), with iterative face registration and an illumination correction method addressing accurate alignment and handling illumination variations is presented. In (Gui et al., 2012), a dimensionality reduction method using sparse representations with a supervised learning scheme called discriminant sparse neighborhood preservation embedding is proposed. In another sparse learning based face recognition method (Jiang and Lai, 2015), a sparse code representation is boosted also with a supervised low-rank dictionary decomposing algorithm. In (Zhuang et al., 2014), to cope with the illumination variance problem, an illumination dictionary using a separate face image library comprising different illumination properties is created. This algorithm is applied to the single sample per person face recognition problem. In another method (Yin et al., 2011), proposed for face verification, a generic face dataset is used, where for any subject samples including a variety of pose and illumination differences exist. When two input faces are fed into the system, they are compared to the generic data according to their pose and illumination attributes and the closest ones are selected which provides the final verification decision.

Although most of face recognition approaches are 2D based, 3D based face recognition also consists of an important family of methods due to its advantages. A typical advantage of 3D face recognition is that depth information as an additional dimension boosts the overall performance (Abate et al., 2007). In fact, in (Xu et al., 2004), the depth value (value of z in a 3D coordinate system) is shown to improve the general accuracy if combined with the pixel intensity values. While 2D based face recognition algorithms attempt to model the face as a 2 dimensional image, the goal for 3D based ones is modeling the 3D face geometry and also the shape of the head. There are basically two ways of creating 3D dimensional face models: One is using many range images which have the depth information. These are used to reconstruct the 3D face. This method does not provide data accurate enough for a perfect face recognition, because in the merging process self-occlusion arising from pose changes can cause information loss. The other one is using directly the output of a 3D camera and gives more precise information because of the nature of the technique. Both approaches usually need, however, multi-view cameras and/or special hardware and advanced image manipulation algorithms which can manipulate 3D object data.

1.2 Objectives of this Dissertation

We can define three objectives for the conducted research we describe in this dissertation. These have to do with face localization, alignment and identification. Although becoming a mature field, many challenges still remain to be handled for face recognition. Pose and illumination variations, face similarities within a family as well as possible difficulties to find enough training samples (which also need to have high enough resolution) for each subject are some of these challenges. From obtaining the raw data to providing them to a face recognition algorithm some processes are obligatory such as locating the face in a camera image and aligning it to prevent suffering from noise due to variances of pose and illumination. Because of all these facts, our first objective is localizing the face accurately. This is done right after the face detection. After the rough face frame is obtained from the face detector, we have developed a novel eye-pair detector algorithm that finds the eye-pair as an important fiducial location in a face. In this algorithm, an eye-pair which denotes two eyes in a face is searched and located by means of a sliding window technique.

Finding and localizing a face are necessary but not enough steps for many cases if the face as a rectangular object is rotated. Therefore, our second objective is aligning faces properly to increase the accuracy performance of a face recognition algorithm. To align the faces, we propose an algorithm that uses the centers of eyes to calculate the face rotation angle to rotate the face to the position in which the angle is 0. The search frame for the eyes is the detected eye-pair frame which is found previously. Using the eye-pair frame instead of the whole face image helps avoiding false positives which can deteriorate the aligner performance.

In some application domains, the performance of a face identification algorithm is limited, if there are not enough reference samples for training.

Sometimes the number of available samples is no more than a few. This is called the small sample problem (SSP). In some cases, there is even only one sample per subject. It is the extreme case of SSP and also known as the single sample per person problem (SSPP). Our third objective is finding a solution to these problems.

To handle the SSPP problem, we propose two novel algorithms which work hierarchically. These are the maximum similarity based region selection (MSRS) algorithm and the Multi-HOG based distance computation method. While the first method attempts to find the most similar regions to avoid cropping and pose errors, the second one uses a multi-HOG based distance computation function to finally obtain the face identities. To handle the SSP problem, we propose a HOG based bag-of-words method (HOG-BOW). In the HOG-BOW method, a codebook is constructed by the k-means clustering algorithm trained on many patches extracted in the training stage. After the codebook construction, it is used to create feature vectors by means of a soft-assignment method. Results of both methods showed a significant improvement for face recognition in the cases of SSP and SSPP.

1.3 Contributions

Eye-pair Detection for Facial Feature Localization

Face localization can be seen as being one step beyond face detection. In Chapter 2 of this dissertation, we propose a novel eye-pair detection algorithm from a loosely detected face image to obtain the face location accurately.

In Chapter 2, we attempt to answer the research question: *How can a face be localized accurately in the case that face detector output is not accurate enough?* To give a solution to this problem, we worked on detecting a face mark which is a very important part of a face: *the eye-pair*. Moreover, instead of detecting eyes separately which is a more common approach found in the literature, we chose to find the eye-pair as a whole rectangle. This is not much investigated except some research on an eye-pair detector designed with the Viola-Jones approach (Castrillón-Santana et al., 2008b).

In this research topic we have demonstrated three important findings: First, finding the eye-pair as a whole in a face was faster and more accurate than a single eye detector which can find two eyes consecutively. Second, our eye-pair detector outperforms the eye-pair detector designed by using the Viola-Jones method. Third, as for feature extractors, the restricted Boltzmann machine (RBM) (Hinton, 2002) with a linear layer was the best filter compared to principal component analysis, Gabor filters and difference of Gaussians filters.

Eye and Eye-Pair Detection for Face Alignment

To fine tune the face detection process and normalize it to the frontal position, face alignment is a required step before the actual face recognition process. In Chapter 3, we focus on this research question: *How can eyes be detected accurately and how can a face image be aligned using limited facial mark information?* We propose to detect eye-pairs and eyes in a cascaded way to be used for face alignment. To rotationally align a face we will use the centers of the eyes which are detected in an eye-pair window.

In this work we have shown two important results: First, the RBM obtains better results than the histogram of oriented gradients (HOG) in terms of rotation angle correction error. Second, rotational errors cause deterioration for the face recognition performance of 6 to 8 percent accuracy in our experiments.

Face Recognition for Single Sample per Person

In Chapter 4 of this dissertation, two Multi-HOG based distance computation functions are proposed for the face recognition problem. One of them is the mean of minimum distances (MMD), and the other is the multi-layer perceptron based distance (MLPD) algorithm. These are combined with a new face similarity search algorithm that finds the best face regions to compare to the other face. In this chapter, we aimed to describe three new algorithms for answering the research question: *How robust can a face recognition algorithm be if only one sample face is available for training?*

In this research topic, our findings are as follows: First, using several HOG filter features contributes to the performance of a 1-NN classifier for the classification of faces. Second, the selection of minimum distances shows better performance compared to making use of all the distances, possibly due to the elimination of occlusions and some easily visible facial expressions (such as big smiles). Third, the multi-layer perceptron based distance computation function achieves the highest accuracy if compared to the other methods. This shows that using a generic face dataset for learning semantic similarities contributes in terms of increasing the actual recognition performance. Also, mirrored images are shown to improve the overall accuracy in general.

Face Recognition for Small Sample per Person

In Chapter 5 of this dissertation, a bag of visual words (BOW) approach combined with HOG for face recognition with a small sample size per person is proposed. The classifier being used here is the L2-SVM for its robustness to deal with long feature vectors. We seek a solution to the research question of *how faces can be recognized under limited data conditions*. In this work, we have made these observations: First, using the HOG-BOW method has been shown to be better than using single HOG or SIFT features. Second, HOG-BOW, combined with the L2-SVM, obtains state-of-the-art face identification performances with very few training examples. Finally, as in Chapter 4, data augmentation by means of mirrored images also help improving the results when the training samples are very limited in number (e.g. 1 sample per person). When more training data are used, however, this technique does not add any improvement to the overall performance.

1.4 Overview of this Dissertation

The rest of this dissertation is organized as follows: In Chapter 2 our eye-pair detection system that is developed for accurate face localization is explained. In Chapter 3, we present the face alignment method which is composed of eye and eye-pair detectors. In Chapter 4, our multi-HOG based face recognition algorithm which is designed for the single sample conditions is described. In Chapter 5, our HOG-BOW method for face recognition which is effective for the small sample problem is explained. Finally in Chapter 6, conclusions and future directions are given.

2

MACHINE LEARNING FOR MULTI-VIEW EYE-PAIR DETECTION

While face and eye detection are well known research topics in the field of object detection, *eye-pair* detection has not been much researched. Finding the location and size of an eye-pair in an image containing a face can enable a face recognition application to extract features from a face corresponding to different entities. Furthermore, it allows to align different faces, so that more accurate recognition results can be obtained. To the best of our knowledge, currently there is only one eye-pair detector, which is a part of the Viola-Jones object detection framework. However, as we will show in this chapter, this eye-pair detector is not very accurate for detecting eye-pairs from different face images. Therefore, in this chapter we describe several novel eye-pair detection methods based on different feature extraction methods and a support vector machine (SVM) to classify image patches as containing an eye-pair or not. To find the location of an eye-pair on unseen test images, a sliding window approach is used, and the location and size of the window giving the highest output of the SVM classifier are returned. We have tested the different methods on three different datasets: the IMM, the Caltech and the Indian face dataset. The results show that the linear restricted Boltzmann machine feature extraction technique and principal component analysis result in the best performances. The SVM with these feature extraction methods is able to very accurately detect eye-pairs. Furthermore, the results show that our best eye-pair detection methods perform much better than the Viola-Jones eye-pair detector.

This chapter was published in:

Karaaba, M.F., Schomaker, L.R.B., and Wiering, M.A. (2014). Machine Learning for Multi-View Eye-Pair Detection. *Engineering Applications of Artificial Intelligence (EAAI), The International Journal on*, pages 392-399.

Face alignment is an important requirement for a successful face recognition application. A human face in an image can be in a variety of scales, positions and poses. Without any alignment of the face entities in an image, recognition performance is very limited. An *eye-pair* is the image that contains a pair of eyes, and it is a significant part of a face. We believe that detection of it can be easier than other parts of a face. Still to the best of our knowledge, there is currently only one eye-pair detection method based on the Viola-Jones framework (Castrillón-Santana et al., 2008b), but as we will show In this chapter, this method is not very accurate. The aim of our work is to develop a system that can accurately detect eye-pairs. This will be useful to address in the future the problem of accurate face alignment and recognition.

Eye or eye-pair detection is a sub-field of object detection in images. The approaches can be classified into three fundamental methods: Shape-based models, feature based models and appearance-based models. Shape-based models depend on a geometrical model of the eyes and use this model to decide whether an image patch contains an eye. It extracts contour properties of the image patch and compares these to the model using a similarity measure. In (Kawaguchi et al., 2000), a separability filter is used for feature extraction and the Hough transform is used for model fitting. Some researchers focus on color images in order to exploit skin color of faces. So, a color conversion algorithm is applied to the image containing a face so that the separation of skin color from the background becomes easier. After the conversion the face is detected by means of a face mask calculation. In (Kalbkhani et al., 2013), a non-linear RGB to YCBCr color conversion is adopted, and an eye mapping algorithm is applied using an already created face mask to find the eyes. In (Huang et al., 2011), an algorithm which converts color pixels from the RGB color space to the HSL space is developed and used. Then, after some image enhancement operations specific to human skin, an object searching algorithm is used for finding eye candidates. Exploiting human-skin color as a discriminator can be very efficient, provided that the background is relatively simple and different than human skin color. In another eye detection and tracking system (Abdel-Kader et al., 2014), eyes are detected and tracked by a particle swarm optimization based multiple template matching algorithm. In another paper, the Hough transform algorithm is used in combination with directional image filters previously proposed for face detection (Maio and Maltoni, 2000). In (Ilbeygi and Shah-Hosseini, 2012), luminance and chrominance values of colored image patches are extracted and given to a template matching algorithm to detect eyes. Shape-based eye detection models may be suitable for real-time eye-tracking applications if they require tracking only the iris and the pupil. However, they are sensitive to different rotation angles and image quality. Moreover, for obtaining more precise results, these models use more parameters to model the shape and this results in an extensive engineering effort and the application is computationally more demanding (Hansen and Ji, 2010).

Feature-based methods focus on finding local features related to the eye. For instance, the eyebrow, the pupil and the iris are basic parts of an eye and locating these features can be helpful for locating the eye. In (Kim and Dahyot, 2008), features of eyes and other facial parts (nose, mouth, etc) of the face are extracted with the SURF algorithm (Bay et al., 2008). Then these features are given to a support vector machine (SVM) (Vapnik, 1998) to locate these facial parts. In (Sirohey and Rosenfeld, 2001), special linear and non-linear filters constructed from Gabor wavelets are used to detect the iris and corner features of the eyes. Then these features are further filtered to remove false features from the detected feature set. A voting mechanism is finally applied to compute the most accurate location of the iris. In (Ando and Moshnyaga, 2013), integral images are utilized for face tracking, face detection, and eye detection. There, instead of eyes themselves, the area between the eyes is exploited as discriminator from other parts of a face. The face area is first obtained by subtracting the adjacent frames of video data and then a seven segmented rectangle template is used to slide through the image which contains the face. The output of the sliding window algorithm is given and processed by an algorithm according to their integral image output values. Since that application is designed for energy-constrained environments, the algorithm it uses is relatively simple which might give inaccurate results for some environments. While feature-based methods are robust to illumination and pose changes, they usually require high-quality images (Hansen and Ji, 2010).

Appearance-based models make a model from eye images by using the photometric appearance of the eyes. Since no specific a priori information related to eyes is used, a sufficient number of training data to learn the parameters for eye detection is needed. For the purpose of eliminating noise and reducing dimensionality, feature extraction and normalization operations to training data are usually applied. As for feature extraction techniques, principal component analysis (PCA), and edge detection methods are some of the techniques being used. After all these operations the output is given to a classifier for training. As classifiers, adaptive boosting (Freund and Schapire, 1995), neural networks and SVMs have been used. In (Huang and Mariani, 2000), patches of example eye images are processed by principal component analysis (PCA) to reduce the dimensionality and make a model eye for classifying unseen image patches if they contain an eye. In (Vijayalaxmi and Rao, 2012), a Gabor filter is used as a feature extractor and an SVM is used as a classifier. To make the final detector more robust to rotations, the face images are populated by rotation, translation and mirroring operations before giving them to a Gabor filter to be processed and then finally the output of it is fed into the SVM to train the classifier. In the face and eye detection method in (Lin et al., 1997), some edge extraction techniques and a histogram equalization algorithm are applied to image patches before they are given to a probabilistic decision based neural network for detection. In (Motwani et al., 2004), wavelet coefficients of image patches are given to a multilayer perceptron. In (You-jia et al., 2010), the output of an orthogonal wavelet analysis on image patches is given to an SVM. The biggest advantage of the appearance-based methods is that they are applicable to all kinds of different objects, because they are based on machine learning algorithms to learn the model from training data. Therefore, they also often require almost no a priori knowledge and less engineering effort. A disadvantage is that they may need a lot of labelled data to learn a very good performing model.

In the Viola-Jones object detection framework (Viola and Jones, 2004), for the eye-pair detector (Castrillón-Santana et al., 2008b) an appearance-based method has been adopted as well. The framework exploits Haar wavelets as object features and these features are calculated using integral images, which makes the computation very efficient. Because of this fact, the face detector of Viola-Jones is known as a very fast face detector and is still a de-facto standard for general platforms where speed can be preferred over accuracy. The method is based on using a cascaded classifier structure using weak Haar features to build a classifier. To train the cascaded structure an adaptive boosting algorithm is used. In this scheme, if a training example is misclassified by the detector, the weight of that example is increased so that the subsequent classifier is able to correct the errors made by the previous classifiers.

Primarily meant to be used for face detection, this framework has been extended for detecting facial parts such as eye, eye-pair, mouth, nose, etc. Nevertheless, this detector is not very accurate and may not be very suitable for platforms where source images are cluttered, noisy or have low-contrast. Since we aim to develop a very robust face recognition application useful for very different types of face images taken in challenging environments, we need high accuracy rather than high speed in order to minimize the recognition error caused by incorrectly aligned face images. Because of this reason, we will utilize a strong classifier and powerful feature extraction methods to increase the discrimination power of the system.

Contributions

In this chapter a novel eye-pair detection method, addressing the problem of face alignment, is proposed. Our aim is to build a robust application, which can deal with many variances in different images, that can also be useful for robots. The system is constructed by using a feature vector extraction method that converts an image patch to the input of a support vector machine (SVM) classifier. We have compared five different feature vector extraction methods. The first one is the linear restricted Boltzmann machine (RBM) (Smolensky, 1986) that extracts activities of latent variables which model the data. The second one directly uses pixel-intensity values. The third method uses principal component analysis to extract eigenvalues from an image patch, and the last two feature extraction methods use the difference-of-Gaussians edge detector and Gabor wavelength filter before the image patch is given as input to a linear RBM. These five feature extraction methods and the SVM classifier are implemented in a sliding window method to find the best matching eye-pair region in a face image. The detector is trained on images we collected from the Internet for which we manually cropped the eye-pair regions. We have compared our methods to the Viola-Jones eye-pair detector on three different test face image datasets (with 240, 450 and 566 face images). The results show that our eye-pair detection systems consistently perform better than the state-of-the-art Viola-Jones eye-pair detector. For almost all test images, the eye-pair regions are located very accurately with our system. Besides, we compare our eye-pair detector application with a single eye detector that we constructed in a similar fashion to show the superiority of using one single wider rectangle which contains two eyes instead of two smaller ones.

Outline. This chapter is organized as follows: In Section 2.1, the classifier and feature extraction methods are described. In Section 2.2, the whole eye-pair detection algorithm is explained. After that, the experimental setup and results are described in Section 2.3. Section 2.4 discusses our findings and describes some directions for future work.

2.1 Classifier and Feature Extraction Methods

2.1.1 Support Vector Machine

The support vector machine (SVM), invented by Vapnik and co-workers (Vapnik, 1998; Boser et al., 1992), is a machine learning algorithm which is very useful for two-class pattern recognition problems (Cristianini and Shawe-Taylor, 2010). The SVM algorithm assumes that the maximum margin between two classes makes the best separation. Although originally developed as a linear classifier, an SVM can be used with non-linear kernels to produce a non-linear classifier. We will shortly describe the SVM. Let D be a training dataset,

$$D = \{(x_i, y_i), 1 \leq i \leq n\}$$

where $x_i \in R^p$ are input vectors and $y_i \in \{1, -1\}$ are binary labels. Given an input vector x_i the linear SVM outputs the following class output o_i:

$$o_i = g(x_i) = sign(w^T x_i + b)$$

where w is the weight vector and b is the bias. To compute the weight vector w and the bias b, the SVM minimizes the cost function:

$$J(w, \xi) = \frac{1}{2} w^T w + C \sum_{i=1}^{n} \xi_i$$

subject to constraints:

$$w^T x_i + b \geq +1 - \xi_i \text{ for } y_i = +1$$

and

$$w^T x_i + b \leq -1 + \xi_i \text{ for } y_i = -1$$

where C weighs the training error and $\xi_i \geq 0$ are slack variables. This is usually done by using the dual formulation, but because the SVM is a well-established machine learning method and not the main scope of our research, we will not go into details here. One possible disadvantage of this soft margin method is that it increases the number of support vectors and therefore it increases the chance of overfitting. A recent algorithm proposes a solution to this, called separable case approximation (Geebelen et al., 2012), which achieves the right separation with a decreased number of support vectors without using soft margins.

Non-linear Case. Although linear separation is faster and less complex than non-linear models, it is not suitable for all kinds of data. Because of this problem, the non-linear SVM model was proposed by Boser, Guyon, and Vapnik (Boser et al., 1992). In this case, the dot product between two input vectors that leads to a linear classifier, is replaced with a non-linear kernel function that allows to separate non-linearly separable data. Many kernel functions have been proposed (Cristianini and Shawe-Taylor, 2010). The most often used kernel functions are the radial basis function (RBF):

$$RBF : K(x,y) = e^{(-\gamma||x-y||)^2}, \gamma > 0,$$

and the polynomial kernel:

$$POLY : K(x,y) = (x^T y + c)^d$$

Recently, to make benefit of discrimination capabilities of both kernels, a combination of these kernels given above is proposed (Afifi et al., 2013), where the kernel formula becomes:

$$POLY - RBF : K(x,y) = (e^{(-\gamma||x-y||)^2} + c)^d$$

When using a kernel function, the decision function becomes:

$$o_i = g(x_i) = sign(\sum_{j=1}^{n} K(x_i, x_j) w_j + b)$$

where the weight w_j for an example x_j is given by $\alpha_j y_j$. Here α_j is computed by optimizing the dual objective problem of the SVM. In this research, SVM$^{\text{light}}$ (Joachims, 1999) is used for training the SVM classifier with the RBF kernel.

2.1.2 Restricted Boltzmann Machine

An RBM is an energy-based neural network model used for suppression of noise and reducing the dimensionality of the input data. It is composed of two layers: an input layer and a hidden layer, which are connected to each other through (symmetric) weighted connections. This structure is called a bipartite graph. For the graphical depiction of an RBM, see Fig. 1. There are many possible implementation methods of these layers depending on the structure of the data to be modeled. While the two layers can be implemented with the same layer type, different activation functions in different layers can also be used. The binary stochastic layer is the most prevalent implementation.

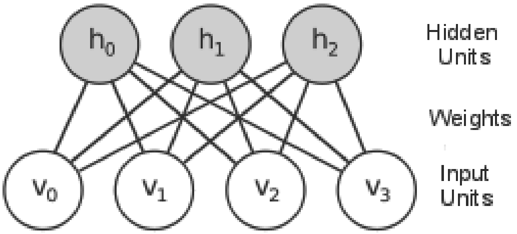


Figure 1: An RBM with 3 hidden and 4 visual (or input) units.

We adopted in this chapter, however, a fully linear RBM, as it was able to model the data better than other implementations of the RBM in our experiments. The mathematical description of the RBM is briefly given below.

Let v_i be the value of input unit i and h_j be the activity value of hidden unit j that models the input data and $\hat{v_i}$, $\hat{h}_j$ are reconstructed input and hidden values. h_j is computed from the input vector by:

$$h_j = f(b_j + \sum_i v_i w_{ij}) \tag{1}$$

$\hat{v}_i$ and $\hat{h}_j$ are computed as:

$$\hat{v}_j = f(a_j + \sum_i h_i w_{ji}), \qquad \hat{h}_j = f(b_j + \sum_i \hat{v}_i w_{ij}) \tag{2}$$

where $f(\cdot)$ is the activation function, a_j is the bias for input unit j, b_j is the bias value for hidden unit j and w_{ij}'s are weights connecting input and hidden units. For the linear function $f(x) = x$ and for the logistic function $f(x) = \frac{1}{1+exp(-x)}$.

To build a model using RBMs, the weight vector w is to be optimized. The most often used method to find the best weight vector, proposed by Hinton (2002), is the contrastive divergence algorithm. In this algorithm, the weight vector w is optimized according to the following update rule:

$$\Delta w_{ij} = \eta(\langle v_i h_j \rangle - \langle \hat{v}_i \hat{h}_j \rangle) \tag{3}$$

where η is the learning rate, $\hat{v}$ are reconstructed values of the input data and $\hat{h}$ are reconstructed values of the hidden units. The angle brackets denote the expected value of any v_i, h_j pair, which are computed using a batch of training examples. Biases are updated by:

$$\Delta a_i = \eta(\langle v_i \rangle - \langle \hat{v}_i \rangle), \quad \Delta b_j = \eta(\langle h_i \rangle - \langle \hat{h}_i \rangle) \tag{4}$$

After the optimization process, values of h_j are computed with the RBM given the input vector and then given to a classifier as a feature vector.

2.1.3 Difference-of-Gaussians Filter

The difference-of-Gaussians (DoG) filter is an edge detection algorithm that detects edges by subtraction of one blurred version of an original image from another, which is a less blurred version of the original. Let f be the image matrix, and let G_1 and G_2 be the first and second Gaussian functions, which produce Gaussian matrices for convolving the image. The Gaussian function is:

$$G_i(x, y) = \frac{1}{2\pi\sigma_i^2} e^{-\frac{x^2+y^2}{2\sigma_i^2}}, i \in \{1, 2\}$$

The Gaussian blurred images are:

$$O_i = (f \otimes G_i(x, y)), i \in \{1, 2\}$$

Where $\otimes$ is a convolution operation. Finally, the final output image is computed by: $O = O_2 - O_1$.

Blurring an image using a Gaussian convolution kernel suppresses spatial information with high-frequency properties. Subtracting one blurred image from the other helps keeping spatial frequencies that are preserved in the two blurred images. So, the DoG can be considered a low band-pass filter which discards all except some significant spatial frequencies that are present in the original image. A detailed analysis of this filter is given in (Basu, 2002).

2.1.4 Gabor Wavelets

A Gabor wavelet is a filter used for edge detection operations. It is a convolution product of a sinusoidal plane wave and a Gaussian function. The mathematical definition of the filter is given below:

$$g_{real}(x, y; \lambda, \theta, \varphi, \sigma, \gamma) = e^{-\frac{x'^2 + \gamma^2 y'^2}{2\sigma_i^2}} cos(2\pi \frac{x'}{\lambda} + \varphi)$$

$$g_{imaginary}(x, y; \lambda, \theta, \varphi, \sigma, \gamma) = e^{-\frac{x'^2 + \gamma^2 y'^2}{2\sigma_i^2}} sin(2\pi \frac{x'}{\lambda} + \varphi)$$

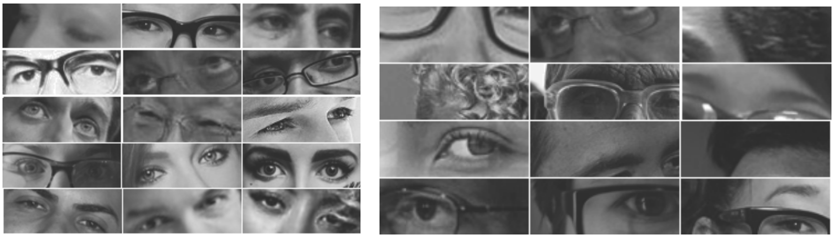
where $x^{'} = xcos\theta + ysin\theta$ and $y^{'} = xsin\theta + ycos\theta$.

As can be seen above, it has five parameters to affect the response of the filter. Here, λ represents the wavelength of the sinusoidal wave function, θ represents the orientation, φ is the phase offset, σ is the standard deviation of the Gaussian function and γ is the spatial aspect ratio which determines the ellipticity of the Gabor function.

2.2 The Eye-pair Detection System

2.2.1 The Training Method

The training method is divided into three parts: Collecting necessary training data, creating feature vectors using a feature extraction method, and supervised training using an SVM for making a model to discriminate between eye-pair and non-eye-pair regions.



(a) (b)

Figure 2: Examples of images used for training. (a) eye-pairs (b) non-eye-pairs.

2.2.1.1 *Image Dataset*

The image dataset was constructed manually at the beginning of the project by us [1] , by collecting images containing a human face from the Internet and then by cropping the eye-pairs as positives and other parts as negatives in these images. The human faces in the images, from which eye-pair and negative image patches are cropped, are in varied positions like different yaw, pitch and roll angles, and a substantial amount of them are with spectacles, also in different sizes and colors. In addition, the faces in the images are in different zoom levels.

The final training dataset constructed from face images contains 1750 eye-pair and 5700 non-eye-pair image patches. The core number of eye-pairs to start with is 300. Then, we first further populated this eye-pair dataset by adding the horizontally mirrored versions of the image patches.

[1] Autonomous Perceptive Systems(APS) Group, University of Groningen, The Netherlands

Second, we added cropped patches which are located one pixel away in the horizontal direction from the manually cropped eye-pair patches. For the non-eye-pair images, we first cropped initially around 2300 image patches from non-eye-pair regions of the faces. After collecting this initial data we evaluated the system with the training method, which is shown in Fig. 3 on training images. In this process we collected the false positives and retrained the system to make it more robust. After repeating the process around 5 times, this resulted finally in 5700 non-eye-pair image patches. For examples of positive and negative samples, see Fig. 2.

The cropping window that is used to pass the specific part of an image to the detector, is a rectangle as can be seen in the figures. We have chosen to use a single rectangle, because it also integrates some information from the upper part of the nose, which can make the detection more reliable. We have also compared this single rectangle to using two smaller rectangles that both surround a single eye. We also want to note that we use all faces in gray scales and do not make use of color information.

Since the eye-pair part of a face represents a small region of the whole face, the number of negative examples which contains non-eye-pairs should be much larger than the number of eye-pair images. Therefore the negative dataset is increased incrementally according to the false positive outputs of the detector on the training images when finding eye-pairs in the training images as shown in Fig. 3.

2.2.1.2 *Creating Feature Vectors*

In this research five feature vector creation methods have been applied. The first feature method uses the hidden activity values of the linear RBM, the second method uses normalized intensity values of image patches, the third method uses hidden activity values of the linear RBM using the output of the DoG filter as input rather than the intensity values, the fourth feature extraction method first uses Gabor filters and then the linear RBM, and the last feature extraction method uses eigenvalues computed using principal component analysis (PCA). We will now describe how we have used these feature extraction methods on our datasets, and which parameters have been used that were found to perform best using preliminary experiments.

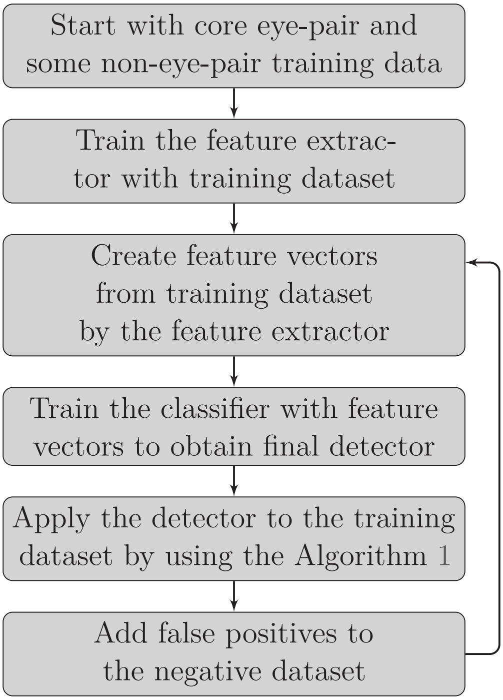
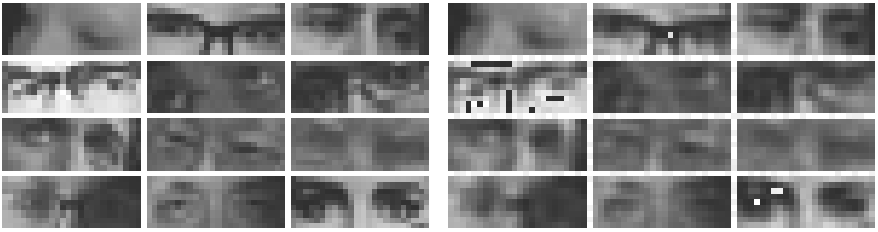

Start with core eye-pair and some non-eye-pair training data
Train the feature extractor with training dataset
Create feature vectors from training dataset by the feature extractor
Train the classifier with feature vectors to obtain final detector
Apply the detector to the training dataset by using the Algorithm 1
Add false positives to the negative dataset

Figure 3: Block diagram of the training algorithm

Hidden Activities of Linear RBM

In this scheme, the feature extractor is a two-layer linear RBM. The weights of the linear RBM are trained iteratively with training data (using positive and negative examples) to produce hidden activities as feature vectors. For training the linear RBM, 60 hidden units are used and the input image resolution is set to 24x9 pixels. Some original eye-pairs and the reconstructed eye-pairs using the linear RBM on training images are shown in Fig. 4a and in Fig. 4b, respectively. Although the reconstructed images are not perfect, they resemble the original ones quite well while reducing the dimensionality from 216 pixel values to 60 hidden unit activations. The learning rate is set to 0.035 from the start and is decreased by dividing by 1.05 for every 10 epochs. For the size of our training set, 50 epochs work well for training the linear RBM. In order to train the neural network faster, the pixel values of each gray-scale image are normalized.



(a) (b)

Figure 4: (a) Some original eye-pair images with 24x9 resolution (b). Reconstructed ones with the linear RBM.

Normalized Pixel Values

The second method is based on directly using pixel intensities. In this scheme, the feature extraction method uses a standard image resizing technique based on a linear interpolation algorithm. After resizing, the gray-scale pixel values of each image patch are normalized between 0 to 1. Since the resolution of 24x9 was shown to give noisy inputs and led to a slower detection performance, we changed the resolution to 16x6 pixels for this method.

Edge Detection Filter Output

In this scheme, the feature extraction method is based on using the hidden activities of the linear RBM which uses the DoG filter output as input. The output of the DoG filter is further smoothed with a noise reducer (despecling) before giving it to the linear RBM. For the DoG filter, the radius of the first Gaussian filter is set to 24x24, while the second filter uses a small radius of 2x2. The standard deviation of both Gaussian filters is set to 1. The image resolution of the search window frame is set to 24x9. We also use 60 hidden units for the RBM in this case.

Gabor Wavelets

In this scheme, the feature extraction method is based on using the hidden activities of the linear RBM which uses the Gabor filter output as input. For the Gabor filter, due to obtaining best performances, the wavelength and bandwidth are selected as 2 and 16 respectively. As for the orientations the angles of $0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}, \pi, \frac{5\pi}{4}, \frac{3\pi}{2}, \frac{7\pi}{4}$, are used. The aspect ratio is selected as 1 (a square). Again 60 hidden units for the RBM are used.

Eigenvalues of PCA
In this scheme, the feature extraction method is based on PCA, which creates eigenvectors as a model of the training data and the projection of the data to this model as feature values. For this method, to make a good comparison, we use the same parameters as the RBM, namely 60 eigenvalues and the resolution is set to 24x9.

2.2.1.3 *Training the SVM with Feature Vectors*

The output of the feature extraction process is used to train the SVM classifier. The radial basis function kernel is used as the kernel of the SVM. The best regularization and gamma parameters, which are two important parameters to be tuned to obtain good classification results, were selected by testing the resulting eye-pair detector on images of the ORL dataset (Samaria and Harter, 1994), which we used as a separate training set. The details about those parameters are given in the experimental section.

2.2.2 Eye-Pair Detection Method

The image including a human face with a visible eye-pair, is resized to user selected predefined scales by preserving the original width-to-height ratio. The algorithm is explained below.

2.2.2.1 *Sliding Windows Technique*

To find the eye-pair in an image, a window with predefined resolution value is slided through the image from top to down and from left to right, to extract different regions. Then the feature vector constructed by a feature extractor on each region is given to the SVM to get a classification result (the discriminative value). The region with the highest output of the SVM is assumed to contain the eye-pair.

2.2.2.2 *Using a Scale of Resolutions*

Since the faces are not in standard scales in the images, the detector assumes a range of different scales. Also because the resolution of the eye-pair search window is fixed, the detector changes the resolution of the image in which it looks for an eye-pair. For an illustration of scale changes of the eye-pair detector, see Fig. 5. The resolution of the image containing an eye-pair is rescaled such that the ratio of width of the main image to width of the search window changes from 2 to 1 with steps of 0.125 while preserving the resolution of the main image. For a detailed explanation of the method, see Algorithm 1. We move the sliding window with 2 pixels in horizontal and vertical directions. Going over a complete image with different scales and locating the eye-pair cost around 1.3 seconds with our method on an average Laptop PC.

Algorithm 1 EyePairDetection (w, inc_w, inc_x, inc_y, w_f,h_f)

```
$w$ is rescaled width of main image being scanned, $w_f \times h_f$ is resolution of detection frame
Set x and y to zero;
while $w \leq max_w$ do
    Calculate original aspect ratio: $r := \frac{h_{org}}{w_{org}}$
    Calculate rescaled height : $h := wr$
    Rescale the original image to $w \times h$ : $I_{(w,h)} := R(I_{org}, w, h)$
    while $y \leq max_y$ do
        while $x \leq max_x$ do
            Get image patch at $x$ and $y$ from main image: $I_p := I_{w,h}(x, y, w_f, h_f)$
            Process the patch with the feature extractor: $v := F(I_p)$
            Get classification value from the SVM: $d := SVM(v)$
            Store this value with $x$, $y$ and $w$ values in a list: $L \leftarrow (d, x, y, w)$
            Increment x value: $x := x + inc_x$
        end while
        Increment y value: $y := y + inc_y$
    end while
    Update width size: $w := w + inc_w$
end while
Return $x$, $y$ and $width$ with the highest discriminant value: $result := argmax\,(L)$
```

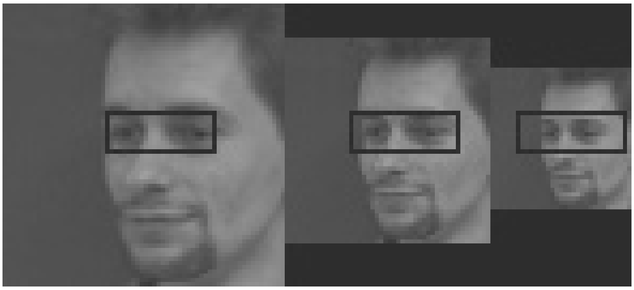


Figure 5: Searching for an eye-pair in different scales of width (54, 48, 33 pixels, respectively) with the same window frame. The best fit here is the middle one with an image width of 48 pixels. The resolution of the window frame (black rectangle) is 24x9.

2.2.3 The Single Eye Detector

In this section, the single eye detector for comparison with the eye-pair detector in terms of accuracy and speed performance is presented. Since we already explained our eye-pair detection method in detail and the detection method of the single eye detector is very similar, only the differences will be given below.

2.2.3.1 *Training the Single Eye Detector*

Collecting the Training Data

Differently from the eye-pair detector a single eye detector searches and finds eyes separately. Because of this fact, we collected around 300 eye images from the main face dataset we constructed for the eye-pair detector. For some eye examples, see Fig. 16b. Then, similarly to our eye-pair detection training method, we collected mirrored versions of the original eyes. Then we created an initial amount of non-eye images from the previously constructed non-eye-pair negative set. Next, we collected negatives by testing the detector on the face dataset we used for training the eye-pair detector and we kept on adding false positives to the negative dataset. In this way we finally aggregated 7800 training images for our single eye detector.

Image Resolution and Feature Extraction Method

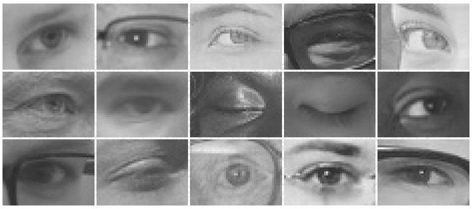


Figure 6: Sample eye images of Single Eye Detector.

As for cropping resolution we used 14x10 (hence 140 pixels) and for feature extraction we selected the linear RBM method for this task as it proved to perform best from our eye-pair detection experiments. For the linear RBM, 50 hidden units are used for training.

Detection Method of Single Eye Detector

Since the eye detector finds eyes separately, it returns always two values for the two best-matching eyes, which is different from the eye-pair detector that returns one detected eye-pair. To prevent ambiguity that the same eye with a slightly different resolution and position appears as the second best matching eye patch, we use a distance condition so that the second best matching eye which does not fulfill this is removed. This condition is given below:

$$d(c_{eye_1}, c_{eye_2}) > l_1 + l_2$$

where c_{eye_1} and c_{eye_2} are the center points of two eyes, $d(\cdot)$ is the function which computes the Euclidean distance between two points and l_i is the maximum distance from the center to the border of the i^{th} eye frame with the angle of the line which connects the center points of two eyes. This condition eliminates the second eye found too close to the best matching eye patch, since they are almost surely the same eye detected twice. Here, l_i is given by:

$$l_i = (w_{eye_i}/2) * sin(a),$$

and the center points are computed as follows:

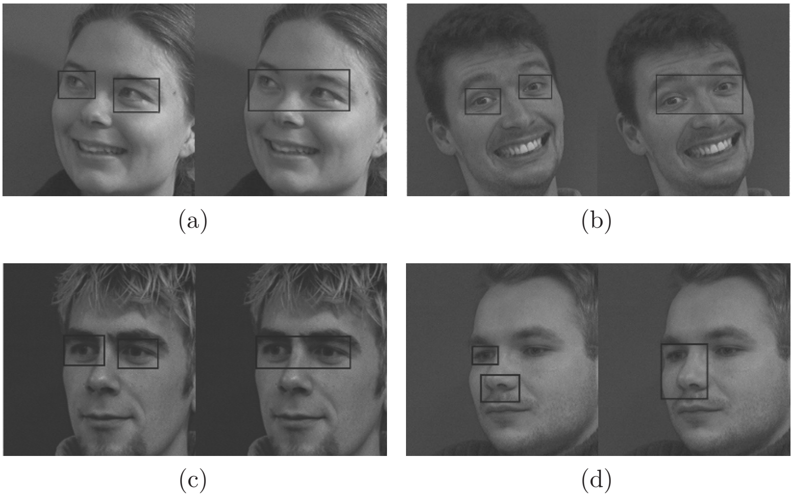
$$c_{eye_i}(x) = (x_{eye_i} + w_{eye_i}/2)$$

$$c_{eye_i}(y) = (y_{eye_i} + h_{eye_i}/2)$$

In the equation above a, the angle of the slope between two center points, is computed from m, where m is the slope of the line which connects two center points:

$$m = \frac{c_{eye_1}(y) - c_{eye_2}(y)}{c_{eye_1}(x) - c_{eye_2}(x)}$$

$$a = \arctan(m)$$


(a)

(b)

(c)

(d)

Figure 7: Illustration of how the bounding box (right parts) is computed from two single detected eyes (left parts) in the single eye detector. Examples of correctly detected eye-pairs: (a), (b), (c), incorrectly detected: (d)

Evaluation Method

To compare the single eye detector to the eye-pair detector, we created a bounding box using the information from two detected eyes. Some example bounding box pictures are given in Fig. 7.

2.3 Experimental Setup and Results

In this section the datasets that are used in the experiments, the evaluation metrics, and the results obtained by the different methods are presented. We have also compared our methods with the Viola-Jones eye-pair detection algorithm.

2.3.1 Datasets

For the tests, the ORL (Samaria and Harter, 1994), the IMM (Nordstrøm et al., 2004a), the Caltech [2] and the Indian [3] face datasets are used. The images in these datasets except the ORL dataset were not seen before for training our eye-pair detection system. The ORL dataset, on the other hand, is used to evaluate training parameters of our method. We have used all face images in these datasets and manually cropped the eye-pair regions to be able to compute the system performances. The ORL face dataset was created at AT & T labs of the University of Cambridge. It involves 400 faces obtained from 40 individuals. The IMM face database was created by the Technical University of Denmark. It contains 240 images obtained from 40 individuals. The Caltech face database was created by Markus Weber at the California Institute of Technology. It contains 450 images obtained from 27 individuals with different lighting/expressions/backgrounds. The Indian face database was created in the campus of the Indian Institute of Technology Kanpur. It contains 566 images obtained from 40 individuals. Some example images of the ORL, IMM, Caltech and Indian datasets are provided in Fig. 8 and Fig. 17. The images in the IMM, Caltech and Indian datasets were cropped manually before giving them to the eye-pair detector.

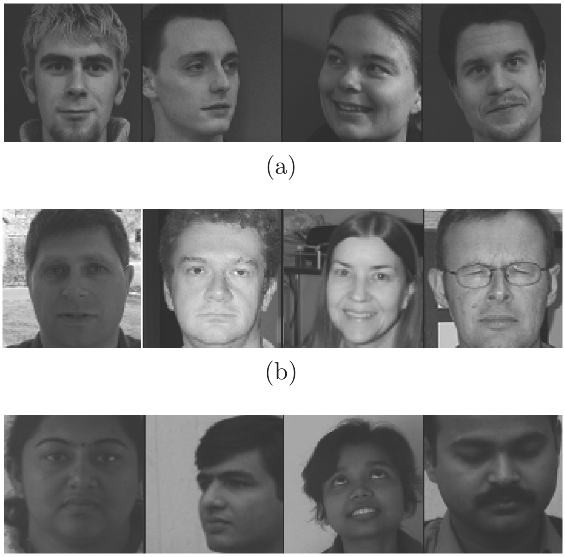
2.3.2 Evaluation

To evaluate the results of automatically detected eye-pairs, an overlapping windows ratio (OWR) metric, which calculates the fraction of matching

[2] Weber. M, Frontal Face Dataset, http://www.vision.caltech.edu/html-files/archive.html

[3] The Indian Face Database, http://vis-www.cs.umass.edu/~vidit/IndianFaceDatabase/

Figure 8: Sample images of the ORL dataset.



(a)

(b)

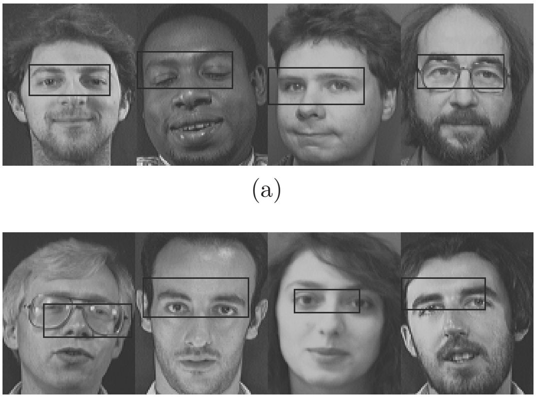
(c)

Figure 9: Sample cropped images of the test datasets: (b) IMM, (b) Caltech, (c) Indian.

pixels between automatically detected and manually cropped eye-pair regions, is used. The detection performance (OWR) is defined by:

$$\text{OWR} = \frac{r}{\sqrt{m * a}}$$

Where r is the matched pixel count, m is the pixel count of the manually annotated ('true') eye-pair region and a is the pixel count of the eye-pair region which is detected automatically by the system. The minimum OWR is 0 and the best obtainable performance is 1, when the windows have equal size and completely overlap. Some examples of face images in which detected eye-pairs have an OWR higher or lower than 0.75 are shown as rectangles in Fig. 10a and in Fig. 10b, respectively.



(a)

(b)

Figure 10: Eye-pair detection results (a) higher than 0.75 OWR, (b) lower than 0.75 OWR for images from the ORL training dataset.

We will compute the percentage of test images that have an OWR above a specific threshold. Finally, we will also report the average OWR on all test images and the standard error.

2.3.3 Results

The SVM needs two parameters to be set (C and gamma), which we optimized after extensive detection experiments on the images in the training dataset. The parameters which worked best are $\gamma = 0.4$ and $C = 6$ for RBM, DoG, PCA and Gabor, $\gamma = 0.2$ and $C = 4$ for the pixel-based method and, $C = 7$ and $\gamma = 0.5$ for the single eye detection method.

The summary of results, as percentage of correctly retrieved eye-pairs (recall) with a minimum OWR of 0.75 and average of OWR results are given in Table 1 and Table 2, respectively. Additionally, we made a comparison of speed performances of our eye-pair detector and the single eye detector. Our eye-pair detector detects the eye-pair within 1.3 seconds on average. The single eye detector can be used to detect the eye-pair within 3.6 seconds. Both methods are significantly slower than the Viola-Jones eye-pair detector, which is optimized for speed, rather than accuracy.

Table 1: Recall performance and the standard errors of Eye-pair Detection Experiments on 3 face datasets. The results are computed using all 240 faces of IMM, 450 faces of Caltech and 566 faces of the Indian dataset.

	Datasets		
Method	**IMM**	**Caltech**	**Indian**
RBM	95% ± 1.4%	97% ± 0.8%	81% ± 1.6%
Pixel	89% ± 2.0%	94% ± 1.1%	83% ± 1.6%
DoG	87% ± 2.2%	89% ± 1.5%	50% ± 2.1%
PCA	95% ± 1.4%	96% ± 0.9%	81% ± 1.6%
Gabor	94% ± 1.5%	88% ± 1.5%	87% ± 1.4%
Single Eye(RBM)	91% ± 1.8 %	91% ± 0.8%	70% ± 1.9%
Viola	80% ± 2.5%	79% ± 1.9%	69% ± 1.9%

Table 2: Average OWR performance and their standard errors of Eye-pair Detection Experiments on 3 face datasets. The results are computed using all 240 faces of IMM, 450 faces of Caltech and 566 faces of the Indian dataset.

	Datasets		
Method	**IMM**	**Caltech**	**Indian**
RBM	0.88 ± 0.005	0.87 ± 0.005	0.79 ± 0.009
Pixel	0.86 ± 0.007	0.86 ± 0.008	0.79 ± 0.009
DoG	0.80 ± 0.015	0.83 ± 0.010	0.53 ± 0.016
PCA	0.88 ± 0.004	0.86 ± 0.005	0.78 ± 0.008
Gabor	0.86 ± 0.005	0.84 ± 0.007	0.81 ± 0.008
Single Eye(RBM)	0.85 ± 0.007	0.86 ± 0.003	0.74 ± 0.009
Viola	0.72 ± 0.020	0.71 ± 0.017	0.64 ± 0.012

As can be seen from Table 1, the linear RBM method and PCA give the best overall results. The PCA method performs very similarly to the linear RBM method and the performance differences of RBM and PCA are statistically insignificant. The pixel-based method closely follows the linear RBM and the PCA methods, except for the IMM dataset in which the pixel-based method performs significantly worse. The IMM dataset contains many rotated faces, with which the dimensionality reduction methods seem to cope better. It can also be seen that the method that uses the DoG filter performs much worse than PCA and the linear RBM method. On the Indian dataset the DoG method performs even worse than the Viola-Jones eye-pair detector. This indicates that the low-contrast and noisy nature of the images of the Indian dataset hinders the DoG filter to perform well. Furthermore, our images have a low resolution and the DoG filter cannot cope well with that. On the other hand, the method that uses a Gabor filter shows somehow varying results. It remarkably outperforms the other methods significantly for the Indian dataset. This is because the Gabor filter increases the contrast which is very helpful for this dataset. The Gabor filter with the SVM gives a close performance to the best feature methods for the IMM dataset. However, for the Caltech dataset, where the illumination properties of the images vary a lot, the Gabor filter diminishes the detection performance of the system a lot. This seems to suggest that highly illuminated images processed by the Gabor filter lose some important information. Finally, our results clearly show that the use of a single eye detector for finding an eye-pair is outperformed by the eye-pair detector when both use the RBM feature extraction method.

As can be noticed from Table 1 and Table 2, the different average OWR results and the recall performance results show a correlation. Table 2 shows that the average OWR on all datasets with our best methods is always larger than 0.78, which shows our methods reliably detect the eye-pairs. Especially on IMM and Caltech, the detection accuracies are very high.

We also show the percentage of retrieved eye-pair regions for the results of RBM, pixel-based methods and Viola-Jones detector when we let the OWR threshold increase from 0.0 to 1.0 in Fig. 11, Fig. 12, and Fig. 13, for the IMM, the Caltech and the Indian dataset, respectively.

In Fig. 11, it can be seen that on the IMM dataset, the RBM performs similarly to the pixel-based method, except for the threshold area between 0.7 and 0.9. In this area, the RBM method outperforms the pixel-based method. The Viola-Jones eye-pair detector performs much worse than these two methods. The Viola-Jones detector often fails to get close to an

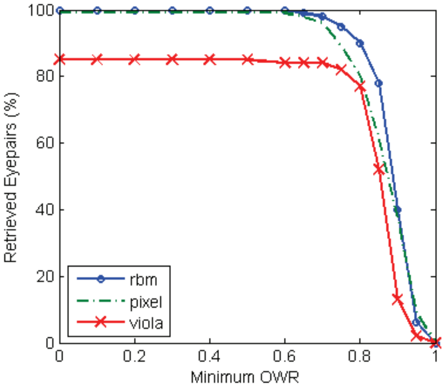


Figure 11: Recall performances of three eye-pair detection systems on the IMM dataset as a function of minimum OWR threshold

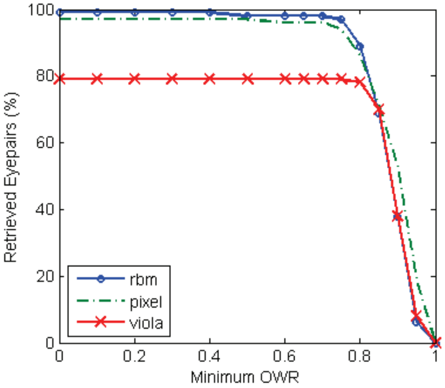


Figure 12: Recall performances of three eye-pair detection systems on the Caltech dataset as a function of minimum OWR threshold

eye-pair at all with around 18% misses, which is shown by the percentage of retrieved eye-pairs with a low OWR threshold.

In Fig. 12, it can be seen that the RBM performs better than the pixel-based method on the Caltech dataset between the thresholds of 0 and 0.9. After the 0.9 threshold, it performs a bit worse than the pixel-based method. The Viola-Jones eye-pair detector with a threshold larger than 0.9, performs similarly to the RBM method and somewhat worse than the pixel-based method. However, just as with the IMM dataset, the

Viola-Jones eye-pair detector quite often fails to find an eye-pair at all. The rough proportion of undetected eye-pairs of the Viola-Jones eye-pair detector is 20%.

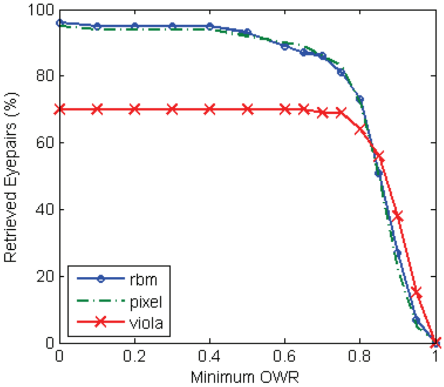


Figure 13: Recall performances of three eye-pair detection systems on the Indian dataset as a function of minimum OWR threshold

In Fig. 13, we can see that the RBM and the pixel-based method perform equally well throughout the whole threshold area on the Indian dataset. After the threshold of 0.85, they perform somewhat less well than the Viola-Jones eye-pair detector, but with lower OWR thresholds the Viola-Jones detector performs much worse than our methods. The difference of undetected eyepairs between the Viola-Jones eye-pair detector and our best method is around 27% until the OWR threshold of 0.4.

To summarize, all these results show that RBM and PCA are the best methods and perform much better than the Viola-Jones eye-pair detector. Visual inspection of images with the SVM discriminant value on the red channel revealed that the midpoint between the eyes is well detected with very rare occasions of maximum values outside of the eye-pair region.

2.4 Discussion

Eye-pair detection is an important step to align different face images for improving a face recognition application. Because of illumination effects, non-rigidity of human eyes caused by ocular muscles and eyelids and also different poses of human faces, the problem is quite difficult to solve. In this chapter we presented and compared different eye-pair detection systems which consist of different feature extraction methods and a support vector machine classifier. We explained how the methods are trained and how they are combined with a sliding window to detect an eye-pair.

The experimental results showed that the linear RBM and the use of principal component analysis give the best results. These two methods generally give more reliable results and they seem to be less sensitive to noisy and low-contrasted inputs. The use of the pixel-based method gives sometimes better results than the other methods on particular images. However, its results have higher variance, indicating that it is less reliable than the RBM and PCA methods. The use of the difference-of-Gaussians filter decreases the performance and with low-contrast images (like in the Indian dataset), its performance can be even quite bad. The Gabor filter results in much better performance levels than the other methods for low contrast and low illuminated images (Indian dataset), but loses its strength significantly for highly illuminated images (Caltech dataset).

The comparison of our application with the Viola-Jones eye-pair detector showed that the Viola-Jones eye-pair detector performs much worse on all datasets compared to the RBM, PCA, Gabor filter, and pixel-based methods. This may be explained by the low information capacity of the Haar features, which are used in the Viola-Jones framework. Finally, the performance of the single eye detector is also much worse than the performance of the eye-pair detector. This confirms our hypothesis that eye-pair detection can be more accurate, because more pixel information can be used.

We will now summarize our main findings. First, using a dimensionality reduction method such as the linear RBM or PCA improves the robustness of the system and lowers the false positive rate. Second, the size of the training dataset directly affects the system's performance. We noticed that increasing the training data with mirrored versions of non-frontal eye-pairs and shifted versions of the cropped eye-pairs was important to get to the very accurate approach proposed here. Furthermore, adding a substantial number of additional negative samples according to the false positive outputs of the system also makes the detector much more reliable and accurate. Third, we want to note that although we trained our detector for eye-pair detection, our approach can be generalized for creating any object detection method, such as face detection and pedestrian detection since no eye-specific modeling was applied in the algorithms presented here.

As future work, we are interested in improving our application even further. Our algorithm generates very few false positives, however, it is always tested on images including human faces. We also want to test the system on natural and indoor images without faces, and to tune the decision threshold in order to minimize false alarms. To increase the accuracy of our application, a combination of different detectors trained on different datasets could be useful. Furthermore, instead of using the 2-layered shallow RBM, a multi-layered deep RBM (deep architecture) might perform better. Finally, we want to use our detector to align faces in a data-mining effort and subsequently to develop a complete face recognition system.

3

IN-PLANE ROTATIONAL ALIGNMENT OF FACES BY EYE AND EYE-PAIR DETECTION

In face recognition, face rotation alignment is an important part of the recognition process. In this chapter, we present a hierarchical detector system using eye and eye-pair detectors combined with a geometrical method for calculating the in-plane angle of a face image. Two feature extraction methods, the restricted Boltzmann machine and the histogram of oriented gradients, are compared to extract feature vectors from a sliding window. Then a support vector machine is used to accurately localize the eyes. After the eye coordinates are obtained through our eye detector, the in-plane angle is estimated by calculating the arc-tangent of horizontal and vertical parts of the distance between left and right eye center points. By using this calculated in-plane angle, the face is subsequently rotationally aligned. We tested our approach on three different face datasets: IMM, Labeled Faces in the Wild (LFW) and FERET. Moreover, to compare the effect of rotational aligning on face recognition performance, we performed experiments using a face recognition method using rotationally aligned and non-aligned face images from the IMM dataset. The results show that our method calculates the in-plane rotation angle with high precision and this leads to a significant gain in face recognition performance.

This chapter was published in:

Karaaba, M.F., Surinta, O., Schomaker, L.R.B., and Wiering, M.A. (2014). In-Plane Rotational Alignment of Faces by Eye and Eye-Pair Detection. *Computer Vision Theory and Applications (VISAPP), The International Conference on,* pages 392-399.

Alignment of a face after the detection from a still image before the image is given to any face recognition algorithm has a crucial importance to obtain accurate results. In particular, rotational alignment is necessary after locating the face, since in unstructured environments the face can appear in any angle rather than frontal. There are three types of rotation angle parameters which determine the pose of a face: roll (in-plane), yaw and pitch. Since the roll angle exists in 2D (hence it is also called in-plane), aligning of it is easier than the other angle parameters. Yaw and pitch angles exist in 3D, and aligning faces which are transformed by such rotations is much harder, because the aligning method has to deal with invisible or deformed parts of the face. We here propose an in-plane alignment of a face using eye coordinates that are automatically found in a face image. In this way we aim to obtain in future work high recognition results with a face recognition algorithm, without the need to use full 3D modeling techniques.

Related Work

For aligning a face image, three general methods have been used: statistical appearance modeling methods, local features methods and geometric calculation methods.

In the first approach, two related methods called active shape models (ASM) (Cootes et al., 1995) and active appearance models (AAM) (Cootes et al., 1998) are popular where statistical information obtained from sample training data is used. The simplest of these methods is ASM. In the ASM method, one manually labels a number of facial landmarks as salient points on example faces used for training the system. These landmark points are then used to model the facial shape. Since positions of these points are correlated, the PCA method is further applied to obtain principal components describing the variances of the point distributions and to make the further calculations computationally more efficient. Since shape information is not sufficient for modeling some complex face data, the AAM method, which is an extension of ASM, has been proposed. AAM combines the shape model with texture information for improving the face recognition system. With both approaches, especially with the latter one, promising results have been obtained. Nevertheless, an intensive labeling effort to obtain all salient points in the training images is required to train these systems.

In the second approach, one uses local features by implementing a local feature extractor without examining global information. An example method for this approach, proposed recently in (Anvar et al., 2013), utilizes the scale invariant feature transform (SIFT) (Lowe, 2004) algorithm as a local feature detector. Here, only one face is labeled with two reference points (the mid-point between two eyes and the tip of the nose) and using the reference information, the rest of the training face images are described automatically using SIFT features. Then a Bayesian classifier is trained on the patches, which are composed of face and non-face SIFT patches, to eliminate non-face features. Since SIFT features include orientation information for each facial feature found, this information is used to estimate the rotation angle. However, high-quality face images, which are not available for every application field, are generally a prerequisite for the SIFT algorithm to perform accurately.

In the third approach, some landmark points localized by detectors are used to determine the correct alignment position of a face. The points used to align a face are usually central points of the eyes, and sometimes the mouth and tip of the nose. After locating these points by a corresponding detector, the face rotation angle can be estimated and the face can be rotated geometrically. In this approach, because the performance of the aligner will depend on the performance of the detectors, detector design becomes an important part of the method. There are two different approaches for detectors: the ones which are implemented using mathematical operators describing object specific information and the others which learn object specific information from sample images. While the methods using the former approach are also called shape-based models, the methods which are based on the latter approach are called appearance-based models. While the former one is faster, its performance strictly depends on the specification of the object to be found. The latter one is slower but more robust to illumination and other noise sources that exist in real-world data (Hansen and Ji, 2010).

To localize an object, using two or more layered systems has been shown to obtain a performance improvement. In (Li et al., 2010a), such an approach has been used to align faces. In that paper, a two-layered eye localization method is adopted such that in the first layer a mathematical operator named Fast Radial Symmetry Transform is implemented to find the points with high radial symmetry in a given image. After locating eye candidate points by this operator, the eye classifier of Castrillon (Castrillón-Santana et al., 2008a) is applied to eliminate false candidate points and to finally locate the eyes in a face image. After the localization, the in-plane rotation angle is estimated by using the central points of the left and right eye. In (Monzo et al., 2011), another hierarchical method is implemented. Here, in the first layer the Adaboost classifier using Haar-like features supplies many promising eye candidates to the second layer. Then the second layer implementing the histogram of oriented gradients (Dalal and Triggs, 2005) and a Support Vector Machine (SVM) is used to localize eyes.

Contributions

In this chapter, we propose a simple yet robust automatic rotational face alignment method in which the in-plane rotation angle of a face is estimated using the eye locations found by eye and eye-pair detector systems. Eyes are localized by the eye detector that searches for eyes in an eye-pair patch obtained with our previously proposed eye-pair detector (Karaaba et al., 2014). The eye detector is implemented by using a feature extractor and a classifier. The method for each detector is based on a sliding window approach. We make use of the restricted Boltzmann machine (RBM) (Hinton, 2002) and the histogram of oriented gradients (HOG) (Dalal and Triggs, 2005) to extract features from the patches belonging to the sliding window. Then the extracted features are presented to a support vector machine classifier (SVM) (Vapnik, 1998). The eye-pair detector is implemented by using an RBM and an SVM. In this chapter, we compare the effects of the HOG and the RBM to study their utility for eye detection.

After locating the eyes in a face image, the in-plane angle is calculated geometrically with the arc-tangent formula using x and y distances between the two detected eyes. Finally, the face is rotated by using that angle. We have tested our method on (subsets of) three different face datasets, namely IMM (Nordstrøm et al., 2004b), FERET (Phillips et al., 1998) and LFW (Huang et al., 2007). Our datasets contain 240, 230 and 450 face images, respectively. We have chosen to use subsets in order to save time on preparation of the datasets and on testing of the methods. We evaluate the performance of our method based on two different evaluation criteria: eye localization error and rotation error. The results show that the RBM feature extraction method performs slightly better than the HOG method on in-plane angle estimations. Moreover, we have also compared the use of rotationally aligned faces to non-aligned faces using a simple but robust face recognition system. The results of that experiment prove that rotational alignment of a face has a high impact on the recognition performance.

Outline. The rest of the chapter is organized as follows: In Section 3.1, the feature extraction technique HOG is described in detail. In Section 3.2, the eye-pair and eye detectors are described together with the method used for computing the rotation angle. In Section 3.3, the experimental platform, the evaluation methods, and the results of the experiments are presented. In Section 3.4, we conclude this chapter.

3.1 Histograms of Oriented Gradients

In this section we will explain the histogram of oriented gradients (HOG) (Dalal and Triggs, 2005), which is used as one of the feature extraction method in this research.

The histogram of oriented gradients, proposed first by (Dalal and Triggs, 2005) for pedestrian detection, is a feature extraction technique which computes the oriented gradients of an image using gradient detectors. It has been applied since then in many other object detection systems such as faces (Zhu and Ramanan, 2012) and on-road vehicles (Arróspide et al., 2013), as well as for object recognition like for recognizing faces (Déniz et al., 2011), emotions (Dahmane and Meunier, 2011) and even actions (Wang et al., 2011).

The mathematical description of the HOG is briefly presented below:

$$G_x = I(x+1,y) - I(x-1,y) \tag{5}$$

$$G_y = I(x,y+1) - I(x,y-1) \tag{6}$$

where $I(x, y)$ is the intensity of the pixel at position (x, y), and G_x and G_y are the horizontal and vertical components of the gradients, respectively.

$$M(x, y) = \sqrt{G_x^2 + G_y^2} \tag{7}$$

$$\theta_{x,y} = \tan^{-1} \frac{G_y}{G_x} \tag{8}$$

While $M(x, y)$ is the magnitude of gradients, $\theta_{x,y}$ is the angle of the gradient at the given location. There are mainly two HOG descriptor calculation methods: Circular HOG (C-HOG) and Rectangular HOG (R-HOG). In this chapter, we used the R-HOG method where the image to be processed is divided into *blocks* which are composed of pixels. For each block a separate histogram is constructed after which all histograms are concatenated to form the feature vector.

As seen from the equations, angles and magnitudes are calculated from the gradients. In the HOG descriptor angles are grouped using *orientation bins*. The *orientation bins* are used to select angles for which magnitudes of gradients are collected. The appropriate bin b_θ for some angle $\theta_{x,y}$ is computed by:

$$b_\theta = \lceil \frac{\theta_{x,y} B}{2\pi} \rceil, \quad 0 \leq \theta \leq 2\pi, \quad 0 \leq \ b_\theta \leq B \tag{9}$$

where B is the bin size.

The calculated contributions of each pixel to the appropriate bin are weighted using the magnitudes and summed up in the final histogram.

3.2 Eye and Eye-Pair Detection

Here, our novel hierarchical detector system based on eye-pair and eye detectors is explained. In this system, it is assumed that a face is detected in a picture by a face detector, therefore we focus only on the eye-pair and eye detection process before the alignment. The system is comprised of two detection layers. In the first layer, the eye-pair detector searches for an eye-pair in an image containing a face. After the eye-pair is found, the eye detector, which is in the second layer, looks for the eyes in the eye-pair region. So, the eye detector assumes its input image is an eye-pair image rather than a face image. Decreasing the search space hierarchically like described above has an advantage that false positives can be greatly reduced in number. Both detectors use a sliding window method to locate the object of their interest and use a detector frame of fixed resolution. On the other hand, an input image is rescaled in a predefined range of resolutions preserving the aspect ratio of the detector frame.

3.2.1 Training Set Construction

To train the eye-pair and eye detector, we first created a face image dataset manually by collecting images containing human faces from the Internet. Although the faces in the images we collected are in different zoom levels, we kept the face-to-image zoom ratio always bigger than 0.5 during cropping. In addition, the collected faces are in various positions and illumination levels making them useful for eye-pair and eye detection purposes in uncontrolled environments (Karaaba et al., 2014). We will now present details about the training dataset collection for the eye detector and additional dataset collection for the eye-pair detector to make it more robust to rotated faces.

Eye Detector Dataset. To construct the eye dataset, we first cropped eye regions of the faces which are around 400 in number. We then added mirrored versions of them to the eye dataset. To obtain negatives, we have used two different methods. The first one is automatic non-eye image collection using initial eye ground truth information and the second one is obtaining the negatives by testing the system with our initially trained detector. We used approximately two times more image patches (for both the positive and negative set) than for the eye-pair dataset used in (Karaaba et al., 2014).

Further Additions. To make the system more robust to rotated faces, we have rotated the face samples in the training sets using angles of $\pm 5^\circ$, $\pm 10^\circ$, $\pm 15^\circ$, $\pm 20^\circ$ using the initial in-plane angle of the faces computed from the manually selected eye coordinates. After this automatically cropped eye-pair and eye regions using the ground truth information of original cropped patches are added to the training set. After aggregating around 1,200 new eye-pairs, we tested the systems (eye and eye-pair detector) on the training set of face images and collected more negatives. The final amount of images in the eye-pair and eye detector datasets increased to 7,000 and 13,500, respectively.

Sample eye-pair pictures used to train the eye-pair detector (in original resolution) are shown in Figure 14. Sample eye and non-eye pictures (in original resolution) are shown in Figure 15.

To locate the eyes, the SVM is invoked on all windows of the sliding window with the appropriate feature vector extracted from the window patch, and finally the highest outputs of the SVM are selected as the locations of the eyes.

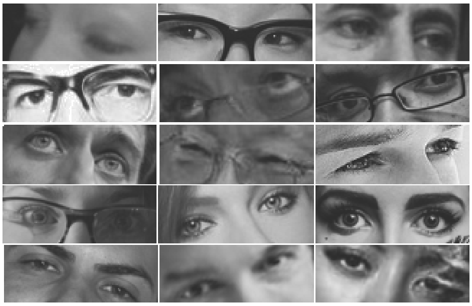
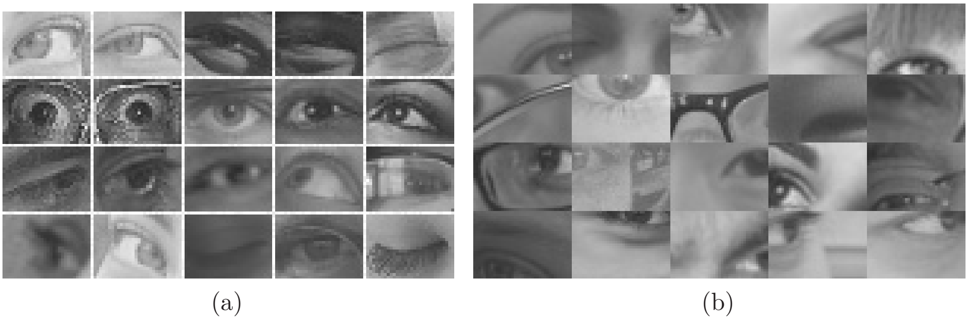


Figure 14: Sample eye-pair regions for training the eye-pair detector.



(a)

(b)

Figure 15: Sample eye (a) and non-eye (b) regions cropped from eye-pair image patches. Note that the non-eye regions may still contain eyes, but they are not very precisely located in the center.

3.2.2 Calculating the Roll Angle

After locating the two eyes, the arctangent formula is used for roll angle calculation:

$$\text{angle} = arctan(\frac{y}{x}) \tag{10}$$

Where

$$y = eye(left)_y - eye(right)_y \tag{11}$$

$$x = eye(left)_x - eye(right)_x \tag{12}$$

Where eye(left) and eye(right) denote the central points of the two eyes. In Figure 16 a graphical representation of the roll angle estimation and the face alignment method can be seen.

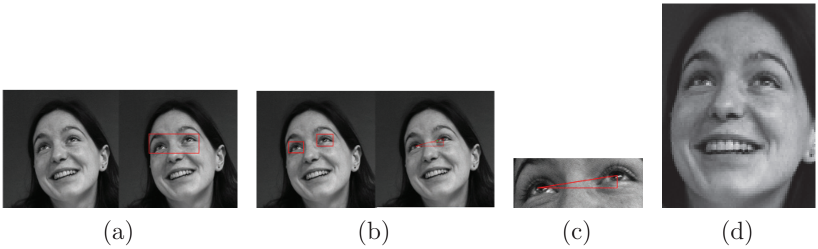


Figure 16: Rotation angle estimation stages: (a) finding eye-pair, (b) finding eyes from eye-pair, (c) computing the angle from central coordinates of eyes (17.5 ° in this example), (d) rotationally aligned face.

3.3 Experimental Setup and Results

In this section general experimental parameters, the face datasets which are used in the experiments, the formulas used for evaluation, and finally the eye detection and in-plane rotation angle estimation results are given.

In our experiments, an SVM classifier Vapnik (1998) has been employed and the RBF kernel is used as non-linear kernel due to its separability power and suitability to the datasets we used.

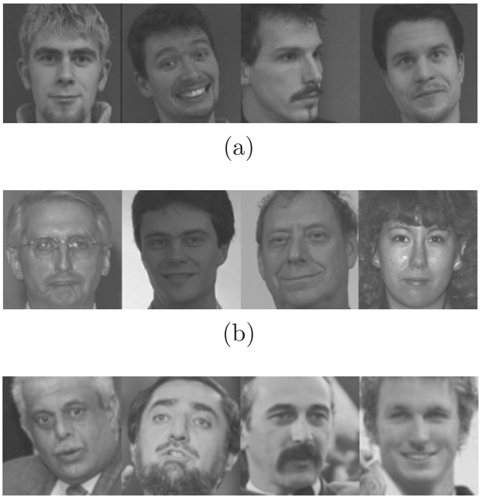
3.3.1 Experimental Parameters

For the eye-pair detector we used the same aspect ratio as in Karaaba et al. (2014). For the eye detector the ratio of a frame is selected as 1.38. The resolution used in the eye detector which uses the RBM as the feature extractor is 18×13 and it is 36×27 for the eye detector which uses HOG. We use 50 hidden units for the RBM and around 100 epochs are employed to train the model. We use a starting learning rate as 0.03 and normalized the input data between 0 to 1 before giving them to the RBM. As for HOG, we chose 4×3×6 (4×3 as block partitioning and 6 bins). According to our observations, while higher feature dimensions for HOG gave slightly better accuracy at the expense of increased computation time, lower feature dimensions gave poorer performance in comparison to the current HOG parameters.

3.3.2 Datasets

For the tests, the IMM Nordstrøm et al. (2004b), the FERET Phillips et al. (1998) and the Labeled Faces in the Wild (LFW) Huang et al. (2007) face datasets are used. We note that the images in these datasets were only used in the testing stage. The IMM face dataset belongs to the Technical University of Denmark and is composed of 240 images with 40 individuals. The FERET dataset was created by the Defense Advanced Research Projects Agency (DARPA) and the National Institute of Standards and Technology (NIST) for the purpose of testing face recognition algorithms. The full dataset is composed of 2,413 facial images with 856 individuals. We use 230 facial samples of the full dataset selected from the first 100 individual folders for our experiments. The LFW dataset is known for containing face images collected in totally unconstrained environments. It contains approximately 13,000 images of around 6,000 people. We selected alphabetically the first 450 images from this dataset. For all the selected images, we determined the rotation angles using the manually established eye coordinates. For some sample face pictures of these test datasets, see Figure 17.

Pose differences caused by yaw and roll angle changes are more prevalent in the IMM than in the FERET dataset. The LFW dataset, on the other hand, includes high variability of illumination and pose differences which makes it very challenging for computer vision algorithms.



(a)

(b)

(c)

Figure 17: Sample face images of the test datasets (b) IMM, (a) FERET and (c) LFW.

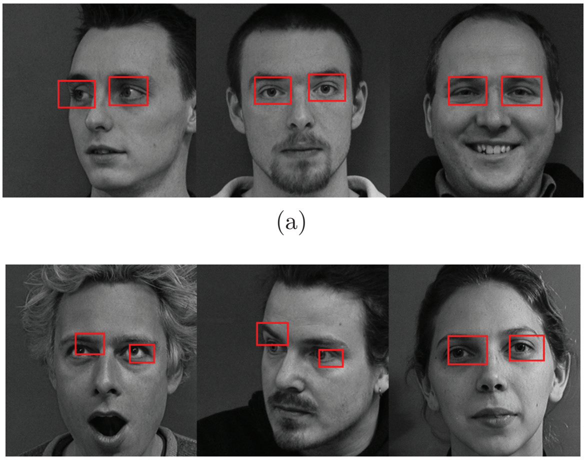
3.3.3 Evaluation Methods

We have used two evaluation methods for our face alignment method. The first one is the eye localization error which is calculated by dividing the pixel localization error by the eye-pair distance. The eye-pair distance is here the Euclidean distance between the central points of the two eyes. The localization error is calculated as follows:

$$e = \frac{d(d_{eye}, m_{eye})}{d(m_{eye_l}, m_{eye_r})} \tag{13}$$

where $d(\cdot, \cdot)$ in (13) denotes the Euclidean distance in 2D and in pixel units, d_{eye} denotes the (center) coordinates of the detected eye, m_{eye} are the coordinates of the manually cropped eye where l and r denote the left and right eyes, respectively. Some examples of face images where eyes are localized with an error lower or higher than a threshold of 0.2 are depicted as rectangles in Figure 18.

The second evaluation method is the angle estimation error which is calculated as the absolute value of the difference between manually obtained and automatically estimated angles (in degrees).



(a)

(b)

Figure 18: Eyes localized with less (a) and more (b) than a localization error of 0.2.

3.3.4 Results

In this section we will show the results using the RBM and HOG feature extraction methods with the SVM as classifier.

We first show the eye localization errors in Table 3 and the rotation angle estimation errors in Table 4. The average localization errors and rotation estimation errors were computed on the natural data without doing any additional artificial rotation. Instead we computed the average errors from all the images we selected for the datasets.

Table 3 shows the results for localizing the eyes. The two feature extraction methods perform similarly. The average localization errors are very small (much smaller than the threshold of 0.2 shown in Figure 18). This also makes the angle estimation errors in Table 4 very small, although the rotation errors are quite sensitive to small errors in the eye localization.

Table 3 also shows that, while we obtain the lowest localization errors for the IMM dataset, the performance of the method deteriorates when the method is applied to the FERET and LFW datasets. Another point is that error results on FERET are close to LFW which is known as one of the hardest datasets due to its realistic nature. The main reason for this is that although LFW possesses complex backgrounds and relatively low contrasted images, the images of FERET vary much more in illumination than the images of the other datasets (see Figure 17).

When we examine Table 4, the average rotation errors are quite small. Meanwhile, although a correlation can be seen between Table 3 and Table 4, lower position errors do not directly imply lower rotation errors. For instance, although average position error results of RBM are a bit higher than HOG results, average rotation estimation results look the opposite. This observation suggests that calculation of rotation angles are sensitive to stability of position information. In this way, we can say that the RBM feature extraction method gives more stable position information than the HOG method.

Table 3: Average Localization Error±Standard Error

Method	**Dataset**	left eye	right eye	average
RBM	**IMM**	.046±.002	.043±.002	.044±.002
	LFW	.071±.004	.069±.005	.070±.004
	FERET	.069±.009	.079±.011	.074±.01
HOG	**IMM**	.044±.006	.041±.004	.042±.005
	LFW	.066±.003	.071±.005	.069±.004
	FERET	.064±.009	.071±.01	.067±.009

The results on the LFW dataset are quite promising when compared to previous results. We only found one paper describing localization errors on LFW, in Hasan and Pal (2011) average eye localization errors on LFW are 0.081 for the left and 0.084 for the right eye. In this study, we obtained lower error rates as can be seen in Table 3.

Table 4: Average Rotation Error ±Standard Error

Method	**Dataset**	average error	successful rotations <2.5° (%)
RBM	**IMM**	1.35±.066	90.0±1.9
	LFW	2.30±.083	65.5±2.3
	FERET	2.38±.118	80.9±2.6
HOG	**IMM**	1.47±.082	80.0±2.6
	LFW	2.46±.096	63.4±2.3
	FERET	2.64±.12	76.5±2.8

As for a general comparison with other works, the survey paper Song et al. (2013) presents a lot of eye detection results obtained with many other possible methods. Our methods (using HOG and RBM feature extraction methods) outperform some of these methods, although the results of the best methods presented in Song et al. (2013) are better than the results obtained with our method. To compare to those results, we want to mention that our best method obtained 95% (96.9%) correctly detected eyes on Feret with an eye localization threshold of 0.1 (0.25), and 85.4% (99%) on LFW with a threshold of 0.1 (0.25).

We also show the plot of the average angle estimation errors in Figure 19. To construct the plot in Figure 19, we first rotated every single face image in one of the experimental datasets (IMM) to 0 ° degrees using the manually annotated coordinates of the eye centers. Then, we rotated every image from -30 ° to 30 ° in steps of 2 ° and for each angle we computed the average rotation estimation error.

The error rates of the method are lowest between -20 ° and 20 ° which corresponds to the range of angles encountered in the training set for the eye detector. Besides, a similar observation already seen in Table 3 and Table 4 about the performance of the two feature extraction methods can also be noticed here. To conclude from all of these observations, the RBM seems to better handle angle estimations than HOG.

Face Recognition. We also show the effect of rotational alignment on the performance of a face recognition system. To make this comparison,

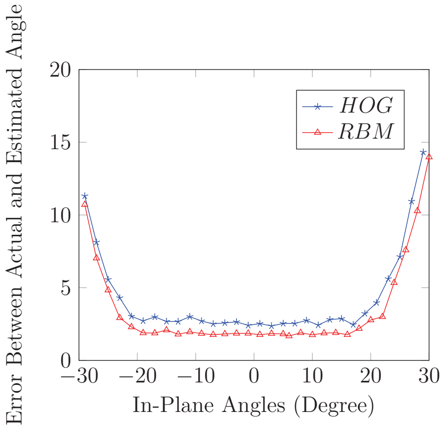


Figure 19: Angle estimation errors on the artificially rotated IMM dataset, as a function of artificial face rotation angles from -30 ° to 30 ° in steps of 2 °

we cropped all face images in the IMM dataset according to the eye coordinates. First, we created the *Non-Rotated* dataset, see Figure 20(a), by cropping using detected eye positions, without using angle information from eye positions to rotationally align the faces. In this way, the eye detection systems using HOG or RBM still operate in a slightly different way.

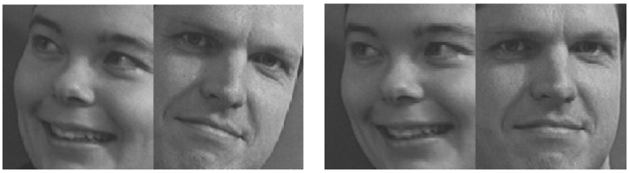
Second, we made an *Automatically Rotated* dataset, see Figure 20(b), by cropping after rotating by using the angle information using the eye positions found.

Then, we used HOG with 3×3×9 parameter settings (3×3 as block resolution and 9 bins) and 60x66 pixels resolution as the input dimension to train the face recognition system. As the IMM dataset contains 6 images per person (6×40 = 240), we selected 4 images for each class as training data and 2 for testing. Then we have in total 160 images for training and 80 images for testing. We subsequently gave the computed HOG features to an SVM *1-to-All* approach and used grid search to find the best meta-parameters to train the model. We selected HOG for this face recognition experiment particularly due to its easy training properties and its relative robustness to illumination variations. These results, however, should not be interpreted as results of an optimally working face recognition system.

With this experiment, we aim to show the influence of rotational alignment. Additionally, we examine the individual effect of each feature extraction technique used in eye detection. Table 5 shows that using automatically rotated faces gives around 6 to 8 percent improvement in recognition performance. If rotated faces are compared by eye detection technique, the use of RBM in the eye detection system gives a slightly better performance than HOG and also gives the highest overall performance.

Table 5: Face Recognition Results on IMM Dataset

	detected by RBM+SVM (%)	detected by HOG+SVM (%)
Non-Rotated	74.50	75.50
Auto. Rotated	**82.75**	81.75
Improvement	8.25	6.25



(a) (b)

Figure 20: (a) faces in original angle and (b) faces rotated using the eye coordinates found by our best performing method.

3.4 Conclusion

Face alignment is an important step to obtain good results with a face recognition system. In this chapter, we have presented a novel face alignment method based on two detectors that operate hierarchically. In this method, first the eye-pair location is found in the face image by the eye-pair detector. Then an eye detector uses the search region, which the eye-pair detector returned, to find the locations of the eyes. This location information is subsequently used to align faces by using a simple geometrical formula. For the eye detector, we also compared results of two feature extraction techniques in eye localization and rotation angle estimation. The results on three different datasets show that the RBM feature extraction technique is better at handling rotation angle estimation than HOG. This is also supported by the angle estimation error plot created by using artificially created angles. We finally examined the effect of rotational alignment in a face recognition experiment in which we compare the use of rotationally aligned and non-aligned faces in a simple face recognition system. The results show that the RBM feature extraction method gives the best angle estimation performance and this in-turn results in better performance in a face recognition system.

4 ROBUST FACE RECOGNITION WITH SINGLE SAMPLE SIZE

The Single Sample per Person Problem is a challenging problem for face recognition algorithms. Patch-based methods have obtained some promising results for this problem. In this chapter, we propose a new face recognition algorithm that is based on a combination of different histograms of oriented gradients (HOG) which we call Multi-HOG. Each member of Multi-HOG is a HOG patch that belongs to a grid structure. To recognize faces, we create a vector of distances computed by comparing train and test face images. After this, a distance calculation method is employed to calculate the final distance value between a test and a reference image. We describe here two distance calculation methods: mean of minimum distances (MMD) and a multi-layer perceptron based distance (MLPD) method. To cope with aligning difficulties, we also propose another technique that finds the most similar regions for two compared images. We call it the most similar region selection algorithm (MSRS). The regions found by MSRS are given to the algorithms we proposed. Our results show that, while MMD and MLPD contribute to obtaining much higher accuracies than the use of a single histogram of oriented gradients, combining them with the most similar region selection algorithm results in state-of-the-art performances.

This chapter was published in:

Karaaba, M.F., Surinta, O., Schomaker, L.R.B., and Wiering, M.A. (2015). Robust Face Recognition by Computing Distances From Multiple Histograms of Oriented Gradients. *Computational Intelligence in Biometrics and Identity Management (IEEE CIBIM), IEEE Symposium Series on, 2015*.

While easily performed by humans, recognizing a face is still a challenging task for computers. Face recognition has two different application fields. One is face identification, where the task is finding the real identity given a sample face image. The other one is face verification, where the task is deciding whether two faces belong to the same person. We focus in this chapter on face identification, due to its demand and popularity.

In the last decade, there has been a significant advancement as to solving the face recognition problem. Nevertheless, face recognition needs to work better for it to be widely employed in the real world. In many application fields, such as security and law enforcement applications, there are often not sufficient reference images to recognize a given test image of a person due to data collection difficulties. This is generally called the small sample size (SSS) problem (Tan et al., 2006). In many cases, even more than one image is not available which is an extreme case of the SSS problem and is named as single sample per person (SSPP).

A known fact in a face recognition task is that the differences of the face images of the same person (intra-class) can be much bigger than differences of the face images of different subjects (inter-class) due to different poses and illumination conditions (Zhang and Gao, 2009; Makwana, 2010). For example, two photos of the same person taken in different poses or illumination conditions will have a higher geometrical distance than two photos of two different people whose pose and illumination conditions are the same. Due to this fact, if there are not enough training samples, a naive face recognition method which basically relies on raw image similarities will not perform well. To overcome such a problem, various methods have been proposed over the years (Tan et al., 2006; Su et al., 2010; Hafiz et al., 2012; Lu et al., 2011; Kveton and Valko, 2013). Because of its importance, in this chapter, we also seek a solution for the SSPP problem.

Related Work

The first proposed methods for the face recognition problem, which were proven effective at their time, are appearance (holistic) based methods. Eigenfaces (Turk and Pentland, 1991) and Fisherfaces (Belhumeur et al., 1997) are the simplest and most well-known methods of these. If there is a sufficient amount of well aligned training samples, these algorithms can work well. However, aligning a face automatically is usually error prone. Also, such methods are sensitive to illumination changes because they directly process pixels.

If pixel intensities are replaced with local feature outputs, better performances can be obtained. The Gabor filter is one of the oldest local feature extractors which is used in many computer vision applications. It has also been applied to face recognition as in (Jemaa and Khanfir, 2009). The scale invariant feature transform (SIFT) (Lowe, 2004) is reported to give promising results (Bicego et al., 2006), and the histogram of oriented gradients (HOG) (Dalal and Triggs, 2005) has also been applied to the face recognition problem successfully (Albiol et al., 2008; Déniz et al., 2011). They are mostly invariant to illumination variations and, provided that there are enough properly aligned training data, they obtain good performances. Particularly HOG has been shown to get better performances than a Gabor filter if combined with an elastic matching method (Albiol et al., 2008). Nevertheless, they are not robust enough to handle the SSPP problem, since the pose variations and aligning errors skew the similarity of train and test distributions of face image data which is essential to obtain good performances.

As alignment is an important part of a face recognition application, active shape models (ASM) (Cootes et al., 1995) and active appearance models (AAM) (Cootes et al., 1998) have been proposed for robustly aligning a face. The basic idea behind the ASM is that face images of a subject can be modeled as a statistical shape model. Later, AAM was proposed on the basis that faces should not be modeled only by points but also by pixel intensities. These methods have also been extended and improved by adding texture information to the model (Kittipanya-ngam and Cootes, 2006; Zhou et al., 2008).

To tackle the SSPP problem, artificial data generation and using generic data are explored in the field. Generating artificial face samples may be effective if these samples decrease the intra-class variance of training samples as well as increase the variance of the inter-class adequately. In (Xu et al., 2014), to exploit the asymmetric nature of face appearances, mirrored images created from original samples as artificial supplementary images were added to the original data and this was reported to perform better than only using original faces. To cope with alignment and pose problems where sufficient data are not available, a generic dataset may also be beneficial. In (Su et al., 2010), generic data are used to learn a Fisher's linear discriminant (FLD) and that generic FLD model is adapted to the actual data.

Patch based methods have been popular in recent years in the face recognition research, because of their successful results. In general, instead of using a whole face image as input, patch based methods divide an input image into several patches, in grid or sliding window fashion. In (Lu et al., 2011), faces are represented as manifolds which are composed of non-overlapping patches. Then, margins for each subject pair are optimized with a reconstruction-based discriminant learning method. This obtained better performances compared to using a single manifold which is based on a whole face image.

In (Lu et al., 2013), a random walk based similarity measure is proposed to compute face similarities. For this, an in-face and an out-face network are constructed. In the in-face network, several overlapping face patch samples together with 8 neighbouring patches are used to make the network. A vector of similarity values is calculated using this network. For the out-face network, the patch locations selected for the in-face network are collected for all the training face samples. Then, the final verification process is performed by these similarity vectors for each face patch-pair. In (Yan et al., 2014), a correlation-based filter bank is constructed to capture similarities of sample images of the same person and differences of similar looking image samples of different people. There they use a grid-based partitioning to compute the patches from the images.

Another popular family of methods is based on neural networks with several layers which are called deep neural networks. Especially convolutional neural networks (CNN) for face verification are reported giving promising results. In (Taigman et al., 2014), a CNN is adopted to learn if two faces are the same or not for face verification. A very large amount of data is used (100K images of 3K subjects) to train the CNN. Besides, a 3D face alignment is employed additional to a 2D alignment before creating the data for the CNN. This alignment contributes to better performances.

In (Zhou et al., 2015), the correlation among the amount of training data, distributions of the train and the test data and the accuracy is investigated regarding to using a CNN approach as the learning algorithm. According to this, increasing the training data is not helpful after some point and the imbalanced sample amount per subject distribution (also called *the long tail effect*) has a negative impact on the performance. The CNN is becoming more popular due to its very good performance potential in computer vision problems, though it requires a large amount of training data and long training times.

Contributions

In this chapter, we propose two novel algorithms which work hierarchically to identify faces. When two input faces are given to the system, first the most similar regions are found by a distance-based search algorithm. The regions, for which the Euclidean distance computed by using HOG features is the smallest, are found by using a sliding window approach. After the best regions for two face images are located, the multi-HOG based algorithm is employed to create a vector of distances on these located regions. The vector of distances is then given to a distance computation function to obtain the real distance value before feeding it to the 1-nearest neighbour classifier (1-NN).

These functions are the mean of minimum distances (MMD) and a multi layer perceptron based distance function (MLPD). To train the MLPD, we used a generic dataset, which is composed of the IMM and the MUCT face datasets. We have tested our algorithms on two face datasets, namely FERET and LFW. The results show that our methods give better or close performances compared to state-of-the-art face recognition algorithms.

Outline. The rest of the chapter is organized as follows: In Section 4.1, the proposed face recognition algorithm is described. In Section 4.2, experimental settings and the results are presented. In Section 4.3, the conclusion and future work are given.

4.1 Proposed Face Recognition Algorithm

4.1.1 Grid-Based Multi-HOG Technique

In this chapter, we use a grid-based distance computation algorithm based on multi-HOG features.

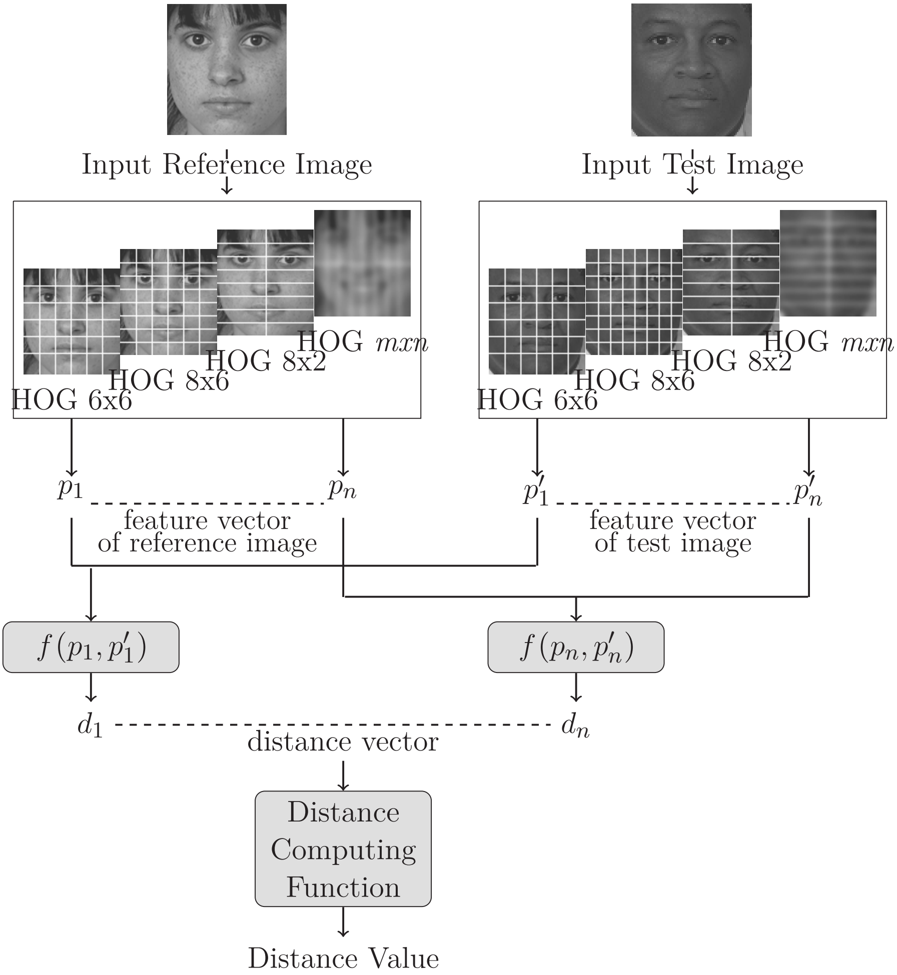


Figure 21: Method of computing the distance between two faces where $f(,)$ is the Euclidean distance function.

Distance Vector Construction from Multi-HOG Features

In the typical HOG method, the image is divided into *sub-images* which are composed of pixels. For each sub-image a separate histogram is constructed after which all histograms are concatenated and normalized to form the feature vector. In our method, on the other hand, from the same input image we create several sub-image sets each of which contains different grid dimensions. Besides, the HOG parameters for each sub-image set are not fixed to the same bin size.

Then, all of these sub-images are used to construct a distance vector which is composed of distance values computed for each image-pair. The Euclidean distance is used to calculate these values. See Fig. 21 for a graphical explanation.

Distance Computation Function

The distance vector is given to a distance computation function to be used for the final classification. After that a 1-nearest neighbor (1-NN) classifier assigns the label of the reference face which has the closest distance to the test image. We used 2 distance computation functions in our experiments: MMD and MLPD functions.

The first method, the MMD, is computed as the mean value of the selected minimum distance values of HOG blocks. First, many distances between different HOG features extracted from different patches are computed.

$$d_i = f(\mathbf{p}_{i_R}, \mathbf{p}_{i_T}), \quad i = 1 \ldots n, \quad \mathbf{p}_i \in \mathbb{R}^b \tag{14}$$

where f is the Euclidean distance function, d_i is Euclidean distance value for the ith patch, $\mathbf{p}_i$ is a patch vector obtained by HOG each of which has bin size b, n is the total number of patches, R and T represent reference and test (patch), respectively. To obtain the set of minimum distances, we use SelectMinimumDistances.

$$\mathbf{d_s} = \text{SelectMinimumDistances}(\mathbf{d}, k) \tag{15}$$

$$0 < k < n, \quad \mathbf{d} \in \mathbb{R}^n, \quad \mathbf{d_s} \in \mathbb{R}^k \tag{16}$$

where $\mathbf{d}$ is the vector of distances produced by the multiple HOG features and $\mathbf{d_s}$ is the vector of minimum distances. For the description of the $SelectMinimumDistances$, see Algorithm 2. We used this algorithm for eliminating the noise which may result from occlusions, accessories (glasses, mustache and beard) as well as facial expressions. We noticed that this was quite effective for obtaining better performances than using all the distances.

Algorithm 2 SelectMinimumDistances ($\mathbf{d}$,k)

```
$k$ is the number of minimum selected distances
Initialize md as minimum distance vector;
d is vector of main distances
while $i \leq k$ do
    find the minimum distance value: $md_i := argmin\,(\mathbf{d})$
    add the minimum distance value to md: $\mathbf{md} \leftarrow md_i$
    remove that value from original distance vector d
end while
Return md
```

Finally, the average distance value is calculated as:

$$\bar{d}_s = \frac{1}{k} \sum_{i=1}^{k} d_{s_i} \tag{17}$$

$\bar{d}_s$ is now the mean of the minimum selected distance values computed from a train and test image pair.
Let there be N reference samples in total. From this we employ basically a 1-NN approach to compute the final label belonging to a test image.

$$C = \arg\min_{c=1}^{N} \bar{d}_{s_c} \tag{18}$$

where C is the class label of the training sample, which is selected as the identity of the test face image.

In the second method, a multi-layer perceptron based neural network is employed. That neural network is trained on distance vectors created from a generic dataset and its output is set to *0* if the distance vector is composed of the face images of the same person, and *1*, otherwise. For this method, we are partly inspired by a face verification approach (Chopra et al., 2005) where the classifier is expected to determine if an image pair is composed of the same person or not.

4.1.2 Adding Mirrored Faces

The face images appear usually in different poses rather than frontal. This presents sometimes serious problems for the performance of a face recognition algorithm. Non-frontal face images can also be considered non-symmetrical. This means that taking the mirrored image of such a picture will supply a novel face image. Due to this, we employed a mirrored version of each training face image sample as a supplementary sample, similarly as in (Xu et al., 2014), and the results show that this improves the recognition performance significantly.

4.1.3 Illumination Correction

Illumination usually presents a problem for a typical face recognition algorithm since it changes the appearance which creates additional noise. Despite that HOG features are generally robust to such changes, our illumination correction method still improves the performance slightly. In our correction algorithm, average brightness and contrast of the image are adjusted according to a fixed mean and standard deviation of pixel intensities.

4.1.4 Maximum Similarity Based Region Selection

In general there are always some small errors in the face alignment process which can cause problems in the comparison stage. This is caused mainly by pose differences and ground truth errors. To handle this problem and improve the performance, we employ a search using the most similar region algorithm that finds the geometrically closest regions between the compared face pairs. This is done by computing Euclidean distances on extracted HOG features when different windows are used, and selecting the sub-images with the smallest distance. After the face regions are obtained for a face pair, these are given to the distance vector construction algorithm. As we will show, finding geometrically more similar facial regions than original ones improves the performance significantly. The pseudo-code of this algorithm is given in Algorithm 3. For a graphical illustration of the most similar regions found by the algorithm, see Fig. 22.

Algorithm 3 SearchMostSimilarRegion ($img_1, img_2, inc_x, inc_y, inc_w, inc_h$)

```
function FDIST($img_1$ , $img_2$)
    $hog_1 = getHog(img_1)$
    $hog_2 = getHog(img_2)$
                                        ▷ getHog is histogram of gradients calculator
return get root mean square of $hog_1$ and $hog_2$
end function
Set $w$ and $h$ to initial values;
Set $x$ and $y$ to zero;
function SEARCH($img_1, img_2, inc_x$, $inc_y, inc_w, inc_h$)
    while $x \leq max_x$ do
        while $y \leq max_y$ do
            while $w \leq max_w$ do
                while $h \leq max_h$ do
                    $subim_2 := subimage(img_2, x, y, w, h)$
                    $similarity := fdist(img_1, subim_2)$
                    $distances \leftarrow similarity$
                    increment $h$ value: $h := h + inc_h$
                end while
                increment $w$ value: $w := w + inc_w$
            end while
            increment $y$ value: $y := y + inc_y$
        end while
        increment $x$ value: $x := x + inc_x$
    end while
    $distance_{min} := argmax\,(distances)$
    Return $x$, $y$ and $w$ and $h$ with the minimum distance value
end function
$(x, y, w, h)@mindist := Search(img_1, img_2)$
$(x, y, w, h)@mindist := Search(img_2, img_1)$
if $dist_1 \leq dist_2$ then
    Return $img_2(x, y, w, h)$
else
    Return $img_1(x, y, w, h)$
end if
```

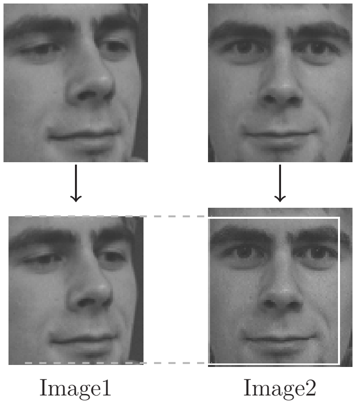


Figure 22: The white rectangle in the second image is selected as the most similar region to the first image.

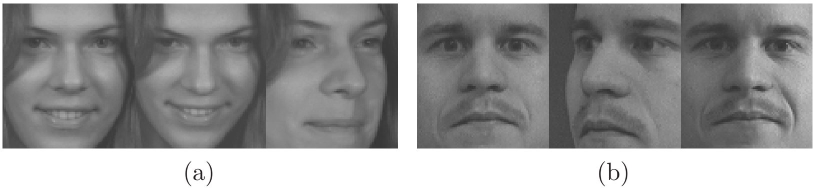

(a)
(b)

Figure 23: Sample aligned face images of one subject from the generic dataset. (a) the MUCT and (b) the IMM dataset.

4.2 Experimental Setup and Results

4.2.1 Datasets

In our experiments, we make use of 4 face datasets: 2 of them (MUCT and IMM datasets) for MLP training, and the rest (FERET and Labeled Face in the Wild (LFW)) for evaluating our methods for final accuracies.

Generic Training

The MUCT dataset was created in December 2008 at the University of Cape Town (Milborrow et al., 2010). It is composed of totally 3,755 face images of 175 individuals. The dataset is divided into 5 categories for different pose angles at which the face pictures are shot. It also has annotations (76 for each photo) for alignment purposes created mainly for experiments of active appearance models (Cootes et al., 1998).

The IMM face dataset was created by the Technical University of Denmark and contains 240 images with 40 individuals (Nordstrøm et al., 2004b). Like MUCT, it also provides annotations. Although it contains a lower amount of samples compared to MUCT, IMM has more pose variations than the former. Therefore we wanted to benefit from both datasets to optimize the parameters of our system. Sample photos of the MUCT and the IMM datasets are shown in Fig. 23.

Test Datasets

The datasets which are used to show final performances are the FERET (Phillips et al., 1998) and the LFW (Huang et al., 2007) datasets. FERET is a huge dataset, thus we selected a subset of this dataset to use in our experiments, which contains 196 subjects with 7 samples of each subject. The subset we chose includes roughly 3 challenging features which can worsen the performance of a face recognition system: illumination changes (dark and bright images), pose (left, right and frontal poses) and expressions (smiles). For example face photos of the FERET dataset which we used for experiments, see Fig. 24.

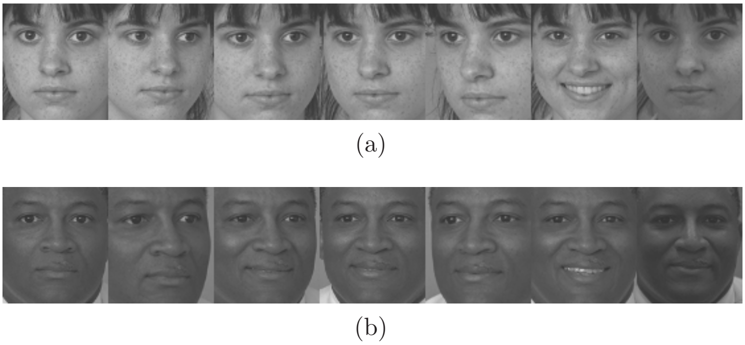
We selected from the LFW dataset 150 subjects each of which contains at least 7 samples. For example face photos of the LFW dataset, see Fig. 25. We selected these dataset configurations similarly as in (Yan et al., 2014).

For all datasets we aligned the face images by using eye coordinates as ground truth. To obtain the eye centers, we used the manual crop information provided in the dataset folder, except for the FERET dataset, from which we cropped the face images automatically by our eye and eye-pair detector (since FERET does not provide sufficient ground truth information for each image) and replaced badly cropped ones with manual crops (around 5% of them). After obtaining eye-coordinates, we followed the aligning method as presented in (Karaaba et al., 2015).

4.2.2 Parameter Tuning

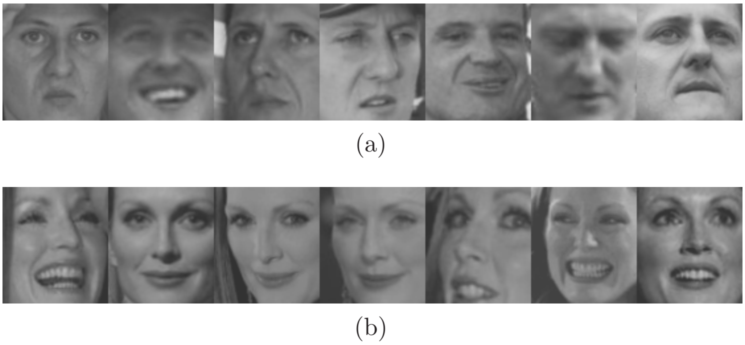
For the MMD algorithm, we choose the minimum 50% of the distances which worked best in preliminary experiments. To create data for the MLPD algorithm, we used 100 subjects from MUCT and IMM as a mixture, yielding about 750 sample pictures of faces with at least 6 samples per subject. We also added a mirrored version of each face image which accounts for 1,500 face images. The distance vector inputs that are given to the MLP are made of combinations of sample pairs. It means that the distance vector amount finally becomes $\binom{1500}{2}$ ≈1,100K. As hidden unit (hu) size, $hu = 15$ worked the best in our system.

To test the model's performance, we used two validation sets each of which is for a corresponding test dataset. These validation datasets are


(a)

(b)

Figure 24: Sample aligned face images of two subjects from the FERET dataset.


(a)

(b)

Figure 25: Sample aligned face images of two subjects from the LFW dataset.

collected from unused parts of the training and test datasets. We then trained the MLP with 10 epochs and saved the model after each epoch. Subsequently, the models which resulted in the best accuracy on the validation datasets are selected for testing with the actual dataset. The reason we used separate validation datasets is to handle the differences of the two datasets, namely for FERET and LFW.

HOG Parameters

We used 80×88 as resolution $width \times height$ of the images. We use the notation of (w, h, b) for a HOG parameter where w is the number of columns, h is the number of rows and b is the number of bins. While the single HOG parameters are chosen as (8,8,24), the combination of HOG parameters (multi-HOG) which worked best in our experiments is (8,8,24), (6,6,24), (2,8,24), (1,11,21), (2,11,21), (8,7,24), (8,6,24), (7,8,24), (5,8,24), (6,8,24) and (7,11,21).

SearchMostSimilarRegion

Finally, the parameters used for Algorithm 3 are as follows:
For the $getHog$ function, $8 \times 8 \times 24$ is used as HOG parameter. inc_x, inc_y, inc_w, inc_h are all set to 2. While $w \times h$ (initial resolution) are initialized to 72×80, $max_w \times max_h$ (highest resolution) is set to 80×88.

Table 6: Face recognition results on the LFW dataset.

Method	Mirrored	No Mirrored
HOG	17.87±0.6	17.61±0.5
Multi HOG MMD	20.61±1.1	20.20±1.0
Multi HOG MLPD	22.34±0.5	22.00±0.6
Multi HOG MSRS-MMD	22.79±1.1	22.14±1.0
Multi HOG MSRS-MLPD	**23.49**±1.2	**22.85**±0.9
DMMA (Yan et al., 2014)	-	22.17±2.8
MS-CFB (cos) (Yan et al., 2014)	-	21.15±2.9

Table 7: Face recognition results on the FERET dataset.

Method	**Mirrored**	**No Mirrored**
HOG	46.52±1.2	39.50±1.3
Multi HOG MMD	55.94±1.0	49.00±1.0
Multi HOG MLPD	64.67±1.2	59.18±1.4
Multi HOG MSRS-MMD	64.43±0.8	57.68±0.9
Multi HOG MSRS-MLPD	**68.59**±1.0	64.68±1.3
DMMA (Lu et al., 2011)	-	65.24±2.0
MS-CFB (cos) (Yan et al., 2014)	-	**66.60**±2.1

4.2.3 Experiments and Results

In order to obtain statistically stable results, we used 10-fold cross validation in both learning (both of MMD and MLPD) and testing stages. In this way, we selected 1 example from each subject folder randomly as the training sample and the rest are used as test samples.

Table 6 and Table 7 show the results [1] (average accuracy and standard deviation). According to these results, if the methods are combined with the MSRS algorithm, our methods perform the best for both LFW and FERET. We also see that mirrored images generally improve the performance. When we use mirrored images together with our best distance functions, the results outperform the others. As easily seen from the table, for both datasets, Multi-HOG shows better results than the single HOG. It suggests that using more than one HOG feature vector captures more information related to the class of the subject. If we use one fixed HOG vector, then pose variations cause increasing intra-class distance variations. For LFW, we have better results than the results of other state-of-the-art algorithms (DMMA (Lu et al., 2011) and MS-CFB (Yan et al., 2014)). It proves the efficiency of our method. For FERET, when we use mirrored versions together with the MLPD method, we obtained the best results. When not using mirrored images, we obtain comparable results. While good results are also obtained with MMD, they are worse than the results obtained

[1] Note that results of DMMA and MS-CFB are referenced from the same source Yan et al. (2014)

with the MLPD function. From this, the usage of generic data is proven to improve accuracy considerably.

4.3 Discussion

In this chapter, we have described three novel algorithms: the most similar region search (MSRS) algorithm, distance vector construction by multiple HOG features which takes multiple HOG-based patches as input to return a distance vector, and the distance computation function which outputs the final distance value. We then introduced two kinds of distance computation functions: namely the mean of minimum distances (MMD) and the multi-layer perceptron based distance (MLPD) function.

Our results showed that using multiple HOG features together with MSRS combined with MLPD obtains the best results for the LFW dataset. For FERET, it gains very comparable results to state-of-the-art methods if no mirrored images are used. But, if the mirrored images are added, then the best results are obtained with our proposed technique. We should also point that MSRS-MLPD gives better results than MSRS-MMD, which proves the usefulness of using a generic dataset. Regarding to using mirrored images, while significant performance improvements can be seen on the FERET dataset, relatively smaller benefits are obtained on the LFW dataset.

In future work we want to improve these results further by using more distance values as well as a bag-of-words approach instead of grid-based fixed partitioning. We also consider using more layers for learning the distance function using a deep learning framework.

5

ROBUST FACE IDENTIFICATION USING HISTOGRAM OF ORIENTED GRADIENTS AND BAG-OF-WORDS

Face identification under small sample conditions is currently an active research area. In a case of very few reference samples, optimally exploiting the training data to make a model which has a low generalization error is an important challenge to create a robust face identification algorithm. In the previous chapter, we investigated and sought a solution for the more extreme case of this problem, named as single sample per person problem, by using multi-HOG features and patch-based distance computation functions. In this chapter we propose to combine the histogram of oriented gradients (HOG) and the bag of words (BOW) approach to use few training examples for robust face identification. In this HOG-BOW method, from every image many sub-images are first randomly cropped and given to the HOG feature extractor to compute many different feature vectors. Then these feature vectors are given to a K-means clustering algorithm to compute the centroids which serve as a codebook. This codebook is used by a sliding window to compute feature vectors for all training and test images. Finally, the feature vectors are fed into an L2 support vector machine to learn a linear model that will classify the test images. To show the efficiency of our method, we also experimented with two other feature extraction algorithms: HOG and the scale invariant feature transform (SIFT). All methods are compared on two well-known face image datasets with one to three training examples per person. The experimental results show that the HOG-BOW algorithm clearly outperforms the other methods.

This chapter was published in:

Karaaba, M.F., Surinta, O., Schomaker, L.R.B., and Wiering, M.A. (2016). Robust Face Identification with Small Sample Sizes using Bag of Words and Histogram of Oriented Gradients. *Computer Vision Theory and Applications (VISAPP), The International Conference on.*

FACE recognition is an important skill which we humans perform without much effort. Computers, on the other hand, still do not perform good enough to be fully trusted in real-world applications. There are two distinct application fields which are both generally called face recognition. One is face identification, in which the question is to whom a given face image belongs, the other is face verification that tries to answer the *same/not same* question given two face images. While face identification is basically a multi-class classification task and requires a reference training image dataset for identity registration, face verification is a binary classification task and does not require a reference training set containing the identity of persons. In this chapter, we focus on the face identification problem.

Face identification is an active research field due to different important possible applications and several difficulties which are not yet solved (Jafri and Arabnia, 2009). Some of these difficulties have to do with pose variances and facial expressions, which arise from the capability we have to move our head and to express ourselves with our faces. Being able to move our heads in various angles results in very different poses of the face of the same person (Zhang and Gao, 2009). If we tilt our heads clockwise or counter clockwise, a simple geometrical alignment procedure is enough to transform the face image to its frontal position. On the other hand, if we turn our head to the left, right, up or down, then without a complex 3d interpolation technique (Chu et al., 2014), geometrical normalization is very difficult, which in turn causes significant performance losses for a face recognition algorithm. Another difficulty is the non-rigidity of the face because we can change the appearance of our faces significantly (opening and closing of mouth and eye, etc). Yet another difficulty is related to occlusions which can be caused by different objects such as glasses, hands we can bring to our face, and shawls (Azeem et al., 2014).

There are many face recognition algorithms that rely on a large amount of training data to work optimally. Since more data will include more variances, the trained classifiers can generalize better to the unknown distribution of the test images. However, in a variety of application fields such as forensic research, data collection is very difficult and the obtained reference data set may not include more than a couple of images per person. This is called the small sample problem (SSP). Many research attempts target SSP (Yan et al., 2014; Lu et al., 2011; Su et al., 2010), and in this study we also propose a new algorithm to deal with few training examples for face identification.

Related Work

To cope with pose differences and alignment problems, the bag of words (BOW) method (Csurka et al., 2004), which has been successfully applied for different computer vision problems (Shekhar and Jawahar, 2012; Montazer et al., 2015), was proposed for the face recognition problem (Li et al., 2010b; Wu et al., 2012). In this method, input images are treated non-holistically by their many sub-images. These sub-images are processed by a clustering algorithm to create a codebook (the bag of words) and this codebook is then used to extract feature vectors from images which are finally given to the classifier.

Similarly to the BOW approach, in (Simonyan et al., 2013), many sub-images processed by the SIFT descriptor are used to train gaussian mixture models to compute improved Fisher vectors (Perronnin et al., 2010) for face verification. The results reported in their paper are comparable with the results of state-of-the-art face verification papers.

As for classifiers used for face recognition, k-nearest neighbour (K-NN), support vector machines (SVM) (Vapnik, 1998) and artificial neural networks (ANN) have been shown to be successful. If classifier speed is important and features from face images are selected robustly, then K-NN can be a good choice. Since no training is required for using the K-NN classifier, it is practical for fast face recognition applications, in which possibly new people are continuously added to the dataset. However, if accuracy is more important than speed, then an SVM (Wei et al., 2011) and an ANN can be preferable, even though they need retraining in case the dataset is augmented with new people and images.

Convolutional neural networks (CNNs), as a powerful feature extractor and classifier, are currently considered by researchers as one of the state-of-the-art machine learning algorithms. CNN is a special kind of multi-layer perceptron, which has many specialized layers used for feature extraction and classification. In a recent CNN based face verification study (Parkhi et al., 2015), a novel database construction and a CNN architecture are presented. Here, they construct a face database with 2.6K subjects composing of total 2.6M images from Internet, removing the duplicate images by employing a state-of-the face recognition application as well as a group of human annotators. After the database construction, they optimize a relatively simpler new CNN which integrates a combination of the most efficient features of the state-of-the-art CNNs proposed recently for face recognition.

The SVM has also several varieties. Although it was first proposed as a linear classifier, non-linear models have been proposed to classify data sets, which are not separable with the linear SVM. Another popular SVM algorithm is the L2-norm regularized SVM (L2-SVM) (Koshiba and Abe, 2003; Deng et al., 2012). It is used to tackle the problem that occurs when the size of the feature vectors is very long (e.g. more than 2000 items) which cannot be handled very efficiently by the standard SVM.

Contributions

In this chapter, as our main contribution, a bag of words (BOW) algorithm is proposed that uses feature vectors extracted with the histogram of oriented gradients (HOG) to recognize faces under small sample per person conditions (SSPP). Although the HOG and BOW algorithms are well-known algorithms, to the best of our knowledge, the combination of them is not evaluated for face recognition, especially in the case of SSPP.

In our method, a K-means clustering algorithm is used to compute the visual codebook from feature vectors extracted by HOG from many randomly cropped sub-images. Then this codebook is used to compute feature vectors from all images in the training and test set. The computed feature vectors and the labels from the training images are subsequently fed into an L2-SVM classifier to learn the model which is used to classify faces. Additionally, we compared the HOG-BOW method to two other well-known methods, namely HOG and the scale invariant feature transform (SIFT), both using the standard-SVM with rbf kernel as classifier since the feature vectors created by these methods relatively smaller than the HOG-BOW method. We performed experiments using two datasets, namely FERET (Phillips et al., 1998) and LFW (Huang et al., 2007) with one, two and three training images per person. The results show that the HOG-BOW method clearly outperforms the other methods.

Outline. The rest of the chapter is organized as follows: In Section 5.1, the face recognition algorithm which is proposed is described. In Section

5.2, experimental settings and the results are presented. In Section 5.3, the conclusion and future work are given.

5.1 Face Recognition by the HOG-BOW Method

The idea of the bag of visual words (BOW) is that, just as a text is composed of many words, an image is composed of many sub-images which resemble visual words that can be present in an image (Csurka et al., 2004). In our proposed HOG-BOW method, the bag of words model is constructed by using features extracted by HOG from sub-images, instead of directly using pixel intensities. We will now explain the codebook construction, the computation of the activity matrix of visual words on the entire image, and the final creation of the feature vector containing visual word activities per block. Note that we use the L2-norm regularized SVM as classifier, but we will not explain it because it is a well-known supervised learning algorithm.

Codebook Construction

Random cropping is used to extract a large number of sub-images (in our experiment we used 500,000 sub-images) from the training set. Then these sub-images are processed by the HOG filter and the extracted feature vectors are given to the K-means clustering algorithm that computes the centroids which serve as the visual words and make up the codebook. For the graphical illustration of the codebook construction, see Figure 26.

Creating Activity Matrix

After the codebook is constructed, the activities of all visual words are calculated per image. These activities denote the presence of different visual words in the image. For this, sub-images are obtained using a sliding window approach using a stride of 1 pixel. To compute the activities the soft assignment approach is adopted in our system. Soft assignment schemes have previously been shown to outperform hard assignment schemes where one sub-image only activates the winning cluster (or visual word). We

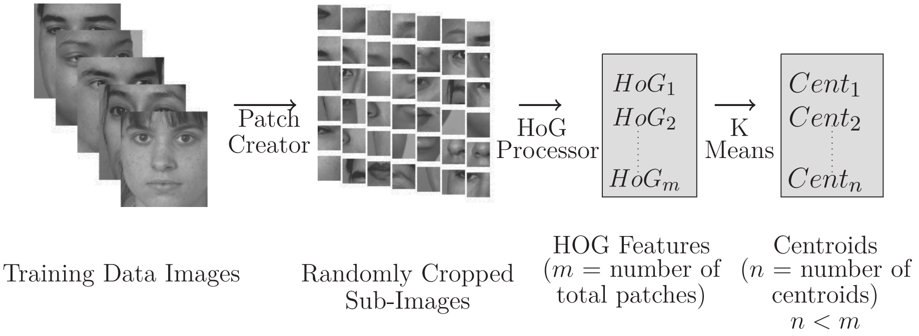

Patch Creator
HoG Processor
HoG_1
HoG_2
HoG_m
K Means
$Cent_1$
$Cent_2$
$Cent_n$
Training Data Images
Randomly Cropped Sub-Images
HOG Features (m = number of total patches)
Centroids (n = number of centroids) $n < m$

Figure 26: Graphical Depiction of the Codebook Construction in *HOG-BOW*

will now explain in detail how the activities a_{ij} of the activity matrix A are computed for a single image, where i is the cluster index, and j is the index of the sub-image (patch). Our method used the soft assignment scheme proposed in (Coates et al., 2011):

$$a_{ij} = max\{0, \bar{d} - d_{ij}\} \tag{19}$$

where $\bar{d}$ is the mean of the elements of d_{ij} and d_{ij} is the Euclidean distance between a cluster c_i and an image patch p_j:

$$d_{ij} = \|p_j - c_i\|_2 \tag{20}$$

Note that p_j is the HOG filtered sub-image vector and c_i is a cluster centroid computed from feature vectors extracted by HOG.

Image Partitioning and Feature Vector Construction

After the centroid activities are computed for each sub-image, each row of the activity matrix (which corresponds to centroid activities for all sub-images) is summed up per image block. We will use B number of blocks to partition each image and to better keep the spatial relations

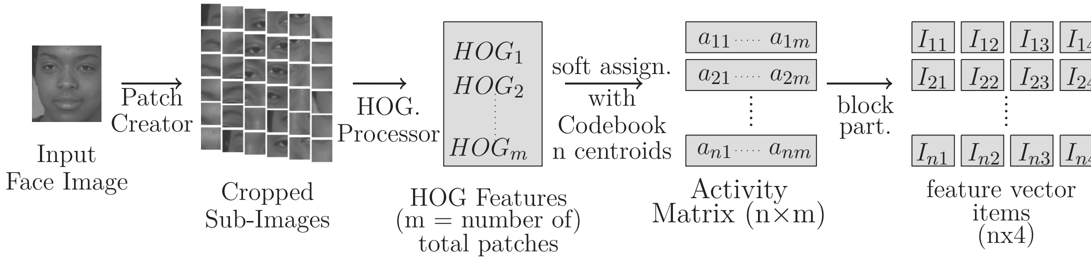


Figure 27: Graphical Illustration of the Creating Feature Vector from Codebook in *HOG-BOW*. Here I_{ij} is a vector and its size is m/4. Note that we chose 4 as the block number(B) which can be chosen differently.

between activated visual words. For this we compute visual word activities I_{ib} for each cluster i and each block b:

$$I_{ib} = \sum_{j \in block(b)} a_{ij}, b \in \{1, 2, 3, 4\} \tag{21}$$

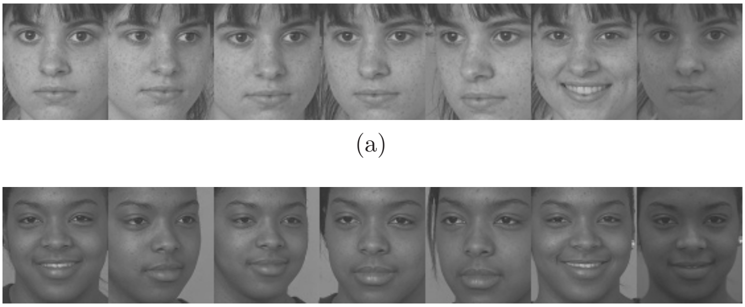
After this the size of the resulting feature vector is $B \times n$. These feature vectors are then given to a classifier. In our experiments we use 4 blocks of equal size. For the feature vector creation, see Figure 27.

5.2 Experimental Settings and Results

In this section, we first briefly explain the datasets used in the experiments, the alignment of the face images, and the selected parameters. After that the results are presented and discussed.

5.2.1 Datasets

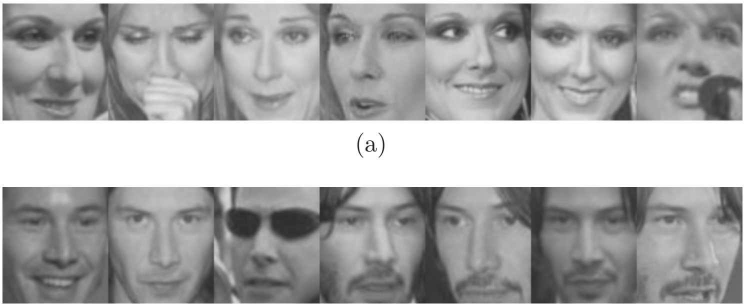
In our experiments we use two datasets, namely FERET (Phillips et al., 1998) and Labeled Faces in the Wild (LFW) (Huang et al., 2007). We



(a)

(b)

Figure 28: Sample aligned face images of two subjects from the FERET dataset.



(a)

(b)

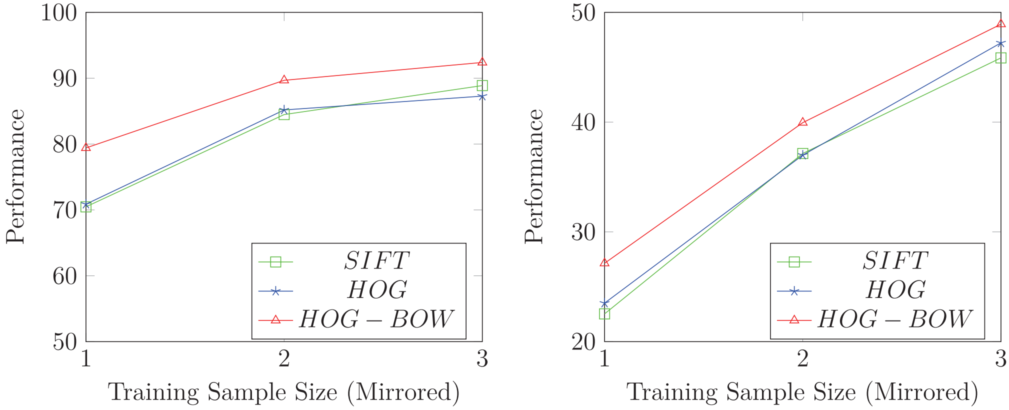
Figure 29: Sample aligned face images of two subjects from the LFW dataset.

divide each dataset into train and test sets by selecting from 1 up to 3 reference images randomly as training data and the rest is used as test data.

We selected a subset of this dataset to use in our experiments, in total 196 subjects are used with 7 face samples per subject. This subset has basically 3 features: illumination, pose and expression variances which present challenges for the performance of a typical face recognition system. For example face photos of FERET, see Figure 28.

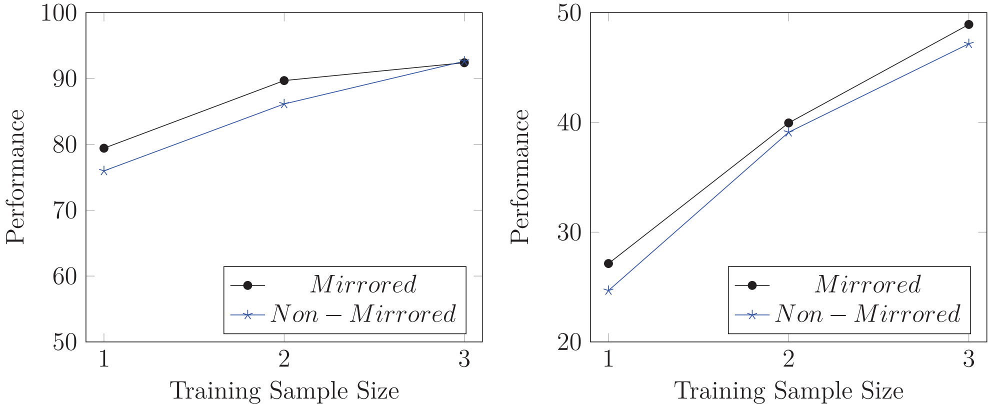
As for the LFW dataset, in the experiments, we have selected 150 subjects each of which contains at least 7 samples. For example face photos of the LFW dataset, see Figure 29.

For both datasets, we adopted a similar experimental setup as described in (Yan et al., 2014).



(a) (b)

Figure 30: Average recognition performance of different methods versus different number of training samples per person on the FERET (a) and LFW (b) databases.



(a) FERET dataset (b) LFW dataset

Figure 31: Average recognition performance of HOG-BOW method with and without mirrored data versus different number of training samples per person on the FERET (a) and LFW (b) dataset.

5.2.2 Alignment

We use an eye-coordinate based 2D alignment for all the face images before the experiments. In this method, eye centers are used to compute the roll angle of the face. Then the face is rotated to roll-normalized position as described in (Karaaba et al., 2015). All eye coordinates are obtained from the dataset directories, except for some images (of each subject) of the FERET dataset for which we used an automatic alignment algorithm.

5.2.3 Selected Parameters

In this section, we will present the selected parameters that worked best in our experiments. For all the train and test images, we use 80×88 as the image resolution. For SIFT, we used 40×44 as the patch size which corresponds to 4 sub-images for each face image. Then for each sub-image by applying the standard SIFT algorithm, we obtained a feature vector with size $(128 \times 4) = 512$.

For HOG, 10×11 is used as the patch size ($8 \times 8 = 64$ patches) and the number of bins is chosen as 24. Hence 8×8×24 is used and the size of the feature vector is 1,536.

For HOG-BOW, 600 centroids are used. For the FERET dataset, 15×15 is selected as the patch size and for the LFW dataset we selected 20×20 as the patch size, which worked better for LFW. The reason different patch sizes were found to work best can be due to differences in the resolution of the two datasets. For both datasets, 4 block partitions are used resulting in a feature vector with size $(600 \times 4) = 2400$. For HOG-BOW method a linear L2-norm regularized SVM is used, for which the C parameter is tuned using cross validation. For the other methods, standard-SVM used, for which C and γ parameters are optimized with cross validation and grid-search.

5.2.4 Experiments and Results

In our experiments, 10-fold cross validation is used. We randomly select $(t = 1, 2, 3)$ samples for each subject from the training set and the rest of the samples is used as the test data. It should be noted that in (Yan et al., 2014), 20-fold cross validation is employed.

Tables 8, 9, and 10 show the results [1]

[1] Note that results of DMMA and MS-CFB are referenced from the same source (Yan et al., 2014)

Table 8: Face Recognition Results on FERET and LFW ($t = 1$)

Methods	FERET		LFW	
	Mirrored	No Mirrored	Mirrored	No Mirrored
HOG	70.87±1.3	57.62±0.7	23.51±0.6	23.73±0.8
SIFT	70.47±1.2	56.51±1.2	22.53±1.0	21.56±0.9
HOG-BOW	**79.41**±3.3	**75.97**±1.1	**27.14**±1.0	**24.68**±0.8
DMMA (Yan et al., 2014)	-	65.24±2.0	-	22.17±2.8
MS-CFB (Yan et al., 2014)	-	66.60±2.1	-	21.15±2.9

Table 9: Face Recognition Results on FERET and LFW ($t = 2$)

Methods	FERET		LFW	
	Mirrored	No Mirrored	Mirrored	No Mirrored
HOG	85.18±0.7	77.78±1.3	36.99±1.2	37.25±1.0
SIFT	84.48±0.8	75.75±0.8	37.14±1.0	36.14±1.1
HOG-BOW	**89.68**±0.6	**86.13**±1.3	**39.95**±1.3	**39.10**±1.1
MS-CFB (Yan et al., 2014)	-	80.60±1.4	-	37.17±1.8

Table 10: Face Recognition Results on FERET and LFW ($t = 3$)

Methods	FERET		LFW	
	Mirrored	No Mirrored	Mirrored	No Mirrored
HOG	87.28±0.8	86.44±0.9	47.22±1.6	**48.25**±1.2
SIFT	88.88±0.6	85.93±1.1	45.85±1.3	46.02±1.3
HOG-BOW	**92.39**±0.6	**92.62**±0.8	**48.92**±1.6	47.16±0.7
MS-CFB (Yan et al., 2014)	-	84.72±1.3	-	43.10±1.5

(average accuracy and standard deviation) on FERET and LFW for $t = 1$, $t = 2$ and $t = 3$, respectively. The results show that the HOG-BOW method obtains the best performances for both datasets, except for LFW without mirrored images with $t = 3$. Especially when the available training data is the smallest in number, the HOG-BOW method shows a significant performance gain (9% and 18% for FERET, and 4% and 1% for LFW for the mirrored and non-mirrored case respectively) compared to the HOG method, which performs second best. The average performance gain over all 12 experimental results of HOG-BOW compared to HOG is slightly more than 5%.

As for the mirrored image samples, a significant performance improvement is obtained for the FERET dataset, especially where $t = 1$. The improvement becomes smaller when more original training data is provided. For instance, while the performance difference is only around 1% for $t = 3$ for almost all the methods, for $t = 1$ this is 4% for the HOG-BOW method and even 13% for the HOG and SIFT methods. This shows that mirrored data sampling is a powerful way to boost the face identification performance for the FERET dataset when there are only one or two training examples per person. On the other hand, for the LFW dataset, mirrored images, except for the HOG-BOW method, do not provide any significant performance gains and even decrease the performance in some cases (e.g. the HOG method with $t = 1$). This might be due to the nature of the LFW dataset where low resolution, occlusions and a high-degree of pose differences are prevalent.

The HOG-BOW method also significantly outperforms two state-of-the-art face recognition algorithms for the non-mirrored case with few training examples. These methods are the multi-subregion based correlation filter bank (MS-CFB) (Yan et al., 2014) with the cosine similarity metric and discriminative multi-manifold analysis (DMMA) (Lu et al., 2011), which were specially designed for face recognition problems with few examples.

We also show two additional figures drawn from the results to obtain more insights. The first one is the comparison of the methods in relation to the training sample size, see Figure 30. The second one is to see the performance effect when mirrored data is added, see Figure 31. As can be seen from the method comparison figures, the HOG-BOW method is always better than the other methods for each training data size if the images are mirrored and its performance stays a large margin above the performances of the other methods. Figure 31 shows that adding mirrored data helps to increase the performance of HOG-BOW the most when the training data size is the smallest ($t = 1$), although in most cases it improves the results.

5.3 Conclusion

In this paper, we described a new face identification algorithm, namely a bag of visual words using extracted features of histogram of oriented gradients (HOG-BOW) with L2-SVM as classifier. This method is designed to cope with small sample sizes in the training set, which is a challenge for obtaining good performances. We compared the HOG-BOW method with two other algorithms: the scale invariant feature transform (SIFT) and HOG, both with a standard SVM as classifier.

We have shown the effectiveness of the HOG-BOW method over the others. On the FERET dataset, for instance, it performs much better than the other methods for all the different selected small sample sizes of the training set. On the LFW dataset, except for $t = 3$ with the non-mirrored case, it also performs significantly better than the other methods. We also compared our results with two state-of-the-art face recognition algorithms by following similar dataset selections. From the results it can be seen that, HOG-BOW obtains state-of-the-art performances for face recognition with few training examples.

In future work, we plan to work on more datasets and we will further optimize the parameters of HOG-BOW to obtain higher accuracies. We are interested to use local binary patterns or features extracted with pre-trained convolutional neural networks (Krizhevsky et al., 2012) instead of HOG as the feature extraction scheme, and combine them with the bag of words approach. Finally, we want to experiment with other clustering algorithms which may work better than simple K-means clustering.

6 DISCUSSION

> Everything has its beauty
> but not everyone sees it.
>
> Confucius

Face recognition and its preprocessing steps facial feature localization and face alignment have gathered a lot of attention from researchers. This may be due to the potential usage possibilities as well as the remaining challenges which need to be solved.

Face recognition has the advantage of its usability without disturbing the person whose face is the input. This may be a useful feature to recognize a person remotely under special security conditions. Apart from the security, face recognition has some potential near future applications such as home or hospital environments. In these places mobile robots can recognize the faces of people in a home environment or can recognize the faces of patients which makes the jobs of hospital personnel easier.

On the other hand, it is not yet easily handled due to the different issues such as pose, lighting and expression variances as well as occlusions and the aging factor that cause it still a challenge for a machine; all of which, accordingly, necessitates more research in this field.

In this dissertation, we focused on these 3 problems: Eye-pair detection for accurate face localization, face alignment using hierarchical eye and eye-pair detection to alleviate some of the pose problems and face recognition under single and small sample conditions.

The rest of the chapter is organized as follows: In Section 6.1, our proposed solutions to these problems as well as the answers to our research questions are briefly explained and in Section 6.2 we discuss possible future directions.

6.1 Concluding Remarks

Eye-pair Detection

In Chapter 2, we proposed the detection of eye-pairs for making a better face localization. Using the eye-pair is our answer to the research question of *how a face can accurately be localized after the face detection process*. The next research question is *what the best feature extractor for the eye-pair detector is*. To answer this question, we investigated the effects of several feature extractors and compared their performances for the eye-pair detection problem.

Now we explain our findings: We have used 5 feature extraction methods, namely a restricted Boltzmann machine (RBM), Gabor+RBM (RBM with the output of Gabor filters as the input), pixel intensities, DoG+RBM (RBM with the output of difference of Gaussians as the input), principal component analysis (PCA). According to the results, in general a single layered RBM with the linear layer outperforms other methods such as pixel intensities and Gabor+RBM. The RBM, furthermore, slightly outperforms PCA. For the Indian dataset, which is the most challenging dataset in our experiments due to its low-contrasted and low-illuminated images, the Gabor filter outperformed other feature extraction methods. The same method for other datasets performed poorer than others, however. Although the pixel intensities method, which uses only the normalized pixel intensities to construct a feature vector, provides meaningful results, it is significantly outperformed by RBM and PCA (6% for recall, 2% for OWR accuracy).

The final research question for this study is *what the best eye-pair detection method is*. As an answer to this question, we have shown that the detection of the pair of eyes using the eye-pair rectangle performs more accurately (%10 for recall and %3 for OWR accuracy) and (2.5 times) faster than the detection of eyes using a single eye rectangle because an eye-pair simply carries more information than a single eye.

Eye Detection for Alignment

In Chapter 3, we aimed to normalize faces from 2D rotated positions to frontal for seeking an answer to the research question *in which way a face can be robustly aligned.*

Our solution to this question is using the centers of the eyes detected in an eye-pair found by our eye-pair detector in a face. These centers provide an angle which can be used to align a face. We then showed that it improved the accuracy of a face recognition algorithm if rotationally normalized faces are given instead of original ones. In addition, another research question is *which feature extraction method can perform best for the problem of eye detection.* We empirically showed that the RBM neural network obtained better accuracies than the histogram of oriented gradients (HOG) descriptor in terms of rotation angle estimation errors.

We also have observed that adding artificial data such as adding mirrored and rotated versions for each eye image also contributes to performance gain. We also learned that while an RBM is better suited to low-resolution images, HOG handles higher resolution images without much performance loss. Since HOG is based on convolution kernel from which the gradients are computed for a histogram, images in lower resolution seem to be less convenient for these computations. At the same time, higher resolution images contain more noisy information which an RBM (due to its dense structure) cannot efficiently handle.

Face Recognition in Single Sample Conditions

In Chapter 4, we investigated one of our main research questions: *how to make a robust face recognition algorithm with only a single reference image?* Our proposed method to this problem is using a combination of HOG descriptors which we call the Multi-HOG method. We described our 2 novel distance computation functions, namely the mean of minimum distances (MMD) and a multi-layer perceptron based distance function (MLPD). We also combined these functions with the maximum similarity based region selection function (MSRS) which attempts to find the closest region in one face image against the other face image given a face image pair. Our findings in summary are as follows:

Multi-HOG obtained better performances than a single-HOG descriptor. This may be explained in this way: When the discrimination capacity of feature vectors increases, their separation becomes easier. This is especially helpful in cases where training examples per class (subject) are very scarce while the total number of classes are high in number. However, not all the distance values are used for final distance computation of the Multi-HOG algorithm. Without MMD or MLPD, lower performances are obtained. We observed that when selecting a subset of distances which have the smallest euclidean distances (as in MMD), it provides more robust results. This could likely be due to removing the distances resulting from occlusion and high pose differences. On the other hand, when an MLP based classifier is trained on an additional dataset for similarity learning, the accuracy of Multi-HOG is much better. This proves the usefulness of using a generic dataset for single sample conditions.

Face Recognition in Small Sample Conditions

In Chapter 5, we sought a solution to the other main research problem: *how a face recognition system performs accurately under very limited reference data*. Our proposed method to this problem is exploiting the bag-of-words (BOW) concept, which is actually borrowed from the natural language processing field, combined with the HOG descriptor with which the method is finally called HOG-BOW.

In this method, randomly cropped image patches of training samples, after being processed by the HOG descriptor, are given to the K-means clustering algorithm to create a codebook. This codebook is then used to create feature vectors from both train and test data. After that training feature vectors are fed into the L2-support vector machine (SVM) classifier to create models for classification. In our experimental settings, we used only 1, 2 and 3 samples as training data per subject and the rest was selected as test data. We also experimented with the same train and test samples with single HOG and SIFT descriptors combined with a standard SVM with the radial basis function kernel. The results have shown that the HOG-BOW method outperformed the other state-of-the-art methods.

6.2 Future Directions

For our eye-pair and eye detection methods, convolutional neural networks (CNN) can be used instead of a combination of RBM and SVM or HOG and SVM to make a faster detector and/or more accurate one if more data can be supplied. In this case, the resulting detector system can also be used in a real-time face and eye tracking system (In fact we are currently working on a real-time robust face identification framework). Alternatively, a multi-layer RBM can also be applied as a feature extractor (by excluding the classification unit) or as a full classifier. As an alternative to an SVM, the multi-layer SVM (ML-SVM) (Wiering and Schomaker, 2014), which has more than one layer and is more powerful in terms of accuracy compared to the standard SVM algorithm, can be used as a classifier.

For our Multi-HOG based face recognition method, pose prediction can be integrated to improve the method's performance. Also, as we stated in the previous paragraph, CNN or multiple layer RBMs can be used for face similarity learning which could possibly replace the MLPD algorithm implemented in our method as well as the HOG descriptor if a CNN is used. Another improvement can be using a higher resolution (more than 80×88 used in our experiments) for images given as input to Multi-HOG distance functions as well as for the MSRS (most similar region selection) algorithm.

For the HOG-BOW method, we have made use of the k-means clustering algorithm to compute the centroids which are used to create a codebook. It can be improved in two ways: Firstly, instead of k-means, k-means++ can be employed, such that, initial centroids as facial features can be given specifically such as patches of mouth, eyes and forehead as initial information. Since clustering performance of k-means depends on initial centroids, its performance can be boosted by the method mentioned above. Secondly, another clustering algorithm can be used instead of k-means or k-means++ (e.g. Local Learning based Clustering (Wu and Schölkopf, 2006)). To increase the discriminative power of feature vectors, Multi-HOG features instead of single HOG can be used. Codebook creation can take longer in this case, since feature vector sizes will increase depending on the number of the additional HOG filters. HOG can also be replaced with a more efficient feature extractor, such as pretrained convolution layers of a CNN (then it may be named CNN-BOW). In this case, however, more data might be needed to obtain meaningful performances.

In conclusion, we believe that all of the future work mentioned here will be helpful to develop a better face recognition system.

SUMMARY

Face recognition is an attractive identity recognition method. Although face recognition is not yet fundamentally solved, it is drawing more attention not only from researchers, but also from private as well as state-owned companies because of its advantages over other biometric techniques such as finger print, iris and speaker recognition.

The merit of face recognition lies in the easiness of its deployment and its application: One does not need a very expensive device to compare the face of a person to others in a face database and also does not need any assistance from a person to show his/her face for recognition. Besides, this is also an important task for many robotic applications. For example a robot which is used to help patients can register the face of a patient without extra human intervention.

Nevertheless, face recognition still remains a scientific problem mainly because intra-class variances usually are of a comparable magnitude to the inter-class variances. For instance face images of two different people taken with the same pose or same illumination condition look more similar than two face images of the same person taken with different aforementioned conditions. This fact challenges the performance of a face recognition algorithm even further under small sample conditions (*i.e., when not many examples are given per identity*) as well as under small resolutions (*i.e., less than* $200x200$).

Face recognition has two distinctive application fields: Face identification and face verification. While for face identification the task is finding the correct identity given a sample face of a subject by using a database with N facial identities, face verification is defined as determining whether two face images represent the same person. In this dissertation we study face identification for the small sample problem.

The processing steps of identifying a face can be outlined as face detection and localization, face alignment and face identification. In face localization, the purpose is to find the location of a face precisely. This can be called an after-process performed following the face detection.

(*although, sometimes, detection and localization can be seen intertwined.*). In this research, a face is localized by precise detection of the eye-pair returned from a face detector. Face localization is followed usually by a face alignment process whose task is normalizing the face with respect to 2D/3D rotation angles and illumination values. This is an important step for a meaningful performance gain. We used eye centers for aligning the face. After this step, localized and aligned faces are given to a face identification algorithm. The face identification algorithm uses the input to compare to the other faces registered in a database of interest and returns the candidate identity with the highest classification likelihood value. In this thesis, we seek a solution to the problems described above under the constraints such as low resolution, small sample sizes, faces in arbitrary poses and illumination conditions. In the following paragraphs, we summarize each chapter from this thesis.

***Chapter 1:* Introduction**

A general introduction for face recognition followed by the objectives and the contributions regarding to this dissertation is given. In the general introduction, first, preprocessing steps for face recognition, namely, face detection, localization and alignment are briefly described. Next, the problem of face recognition is defined and several state-of-the-art methods in the literature are shortly reviewed. In the objectives, the reasons of conducting this research are explained. After that our contributions of this research are summarized and this is followed by a short overview of the dissertation.

***Chapter 2:* Eye-Pair Detection for Face Localization**

Eye-pair detection, in which one captures both eyes in a single rectangle, has not much been researched except the eye-pair detector developed by using the Viola and Jones method. Therefore, in this chapter, we propose novel eye-pair detection methods based on some state-of-the-art feature extractors and a support vector machine (SVM) as a classifier. These feature extractors are image filters such as the difference of Gaussians (DoG), the Gabor filter, the linear restricted Boltzmann machine (RBM) and principal component analysis (PCA) as well as normalized pixel intensity values. We also compare these methods to the eye-pair detector of the Viola-Jones method. We use a sliding window approach to obtain several samples any of which can contain an eye-pair in an image

containing a face. These samples are given to a feature extractor which further passes them to an SVM which classifies a sample for whether it contains an eye-pair. We performed our experiments for the purpose of comparing the power of feature extractor methods on three benchmark datasets: IMM, Caltech and Indian. The results show that the linear RBM obtains the best accuracy for most cases followed by the PCA and pixel intensity methods. Moreover, all of our methods perform better than the Viola-Jones eye-pair detector (except the DoG method on the Indian dataset). Besides we also compare our eye-pair detector which detects an eye-pair in a rectangle with the eye-pair detector which detects a pair of eyes in two stages (a single eye detector). Finding the eye-pair within a rectangle in one stage not only results in a more accurate detection result (average 6% better recall) but also runs faster (approximately 3 times).

Chapter 3: **Rotational Alignment by Eye and Eye-Pair Detectors**
This chapter presents a hierarchical eye detection system that first obtains the location of the eye-pair after which the center of each eye is located. In this method, many samples are obtained by using a sliding window. These samples are later given to a feature extractor to construct feature vectors which are subsequently provided to an SVM with a radial basis function kernel to select the best eye candidate. After the centers of the eyes are obtained, the 2D (in-plane) rotation angle is computed which is subsequently used to align the face. We used two feature extraction methods in this chapter: the RBM and the histograms of oriented gradients (HOG) and two test data sets: a subset of FERET and the Labeled Faces in the Wild (LFW). According to the rotation angle evaluation results, our approaches perform very accurately. We also compared the effect of rotational alignment on face recognition. To do this, the IMM dataset is used and according to the results aligning faces rotationally before giving them to the face recognition algorithm improves the recognition accuracy with 6 to 8 percent.

Chapter 4: **Face Recognition by Multiple HOG and Distance Computation**
In this chapter a distance based face identification algorithm which uses multiple histograms of oriented gradients (Multi-HOG) as a feature extractor for the single sample problem is introduced. In this algorithm, a

distance computation function is employed to obtain distances for each face pair. We use two distance computation functions: mean of minimum distances (MMD) and a multi-layer perceptron based distance (MLPD) function. After features are extracted from the Multi-HOG filter, they are given to a distance computation function which outputs it to a 1-nearest neighbor classifier for the final classification. An MLP similarity classifier is trained on two generic datasets namely IMM and MUCT which returns 0 if a face pair is composed of the same subject and returns 1 otherwise. Besides, to alleviate the aligning errors, we propose the most similar region selection algorithm (MSRS) which seeks to find the closest regions given a face pair and returns the region coordinates to the face identification algorithm. We evaluated our methods on two test datasets: a subset of FERET and LFW. The results of these methods are compared with a single HOG and some state-of-the-art face identification algorithms. According to these comparisons, our method performs better than the other methods: our best results are 2% on FERET and 1% on LFW higher both utilizing mirrored samples.

Chapter 5: **Face Identification with HOG-BOW Method**
This chapter presents a novel face identification algorithm we called HOG-BOW which combines the histogram of oriented gradients (HOG) and the bag of words (BOW) approach to utilize few training examples. In this method, first a codebook is constructed by the k-means clustering method which computes centroids from many sub-images cropped from every training image. The train and test feature vectors are constructed subsequently using the centroids. These vectors are then given to the L2-SVM for classification. The datasets we used for tests are FERET and LFW similar as the previous chapter. This proposed method is compared with HOG and SIFT feature descriptors and the HOG-BOW method outperformed both of these methods significantly.

Discussion Chapter
This is the last chapter where concluding remarks and future directions are given. In the concluding remarks section, our findings in relation with each chapter as well as our answers to the research questions are explained. In the future directions, we discuss about how our proposed methods can be improved by making use of the deep learning paradigm.

BIBLIOGRAPHY

Abate, A. F., Nappi, M., Riccio, D., and Sabatino, G. (2007). 2D and 3D Face Recognition: A Survey. *Pattern Recogn. Lett.*, 28(14):1885–1906.

Abdel-Kader, R. F., Atta, R., and El-Shakhabe, S. (2014). An efficient eye detection and tracking system based on particle swarm optimization and adaptive block-matching search algorithm. *Engineering Applications of Artificial Intelligence*, 31:90–100.

Afifi, A., E.A.Zanaty, and Ghoniemy, S. (2013). improving the classification accuracy using support vector machines (svms) with new kernel. *Journal of Global Research in Computer Science*, 4(2):1–7.

Albiol, A., Monzo, D., Martin, A., Sastre, J., and Albiol, A. (2008). Face recognition using HOG-EBGM. *Pattern Recognition Letters*, 29(10):1537 – 1543.

Ando, T. and Moshnyaga, V. G. (2013). A low complexity algorithm for eye detection and tracking in energy-constrained applications. In *Communications, Signal Processing, and their Applications (ICCSPA), 2013 1st International Conference on*, pages 1–4.

Anvar, S., Yau, W.-Y., Nandakumar, K., and Teoh, E. K. (2013). Estimating In-Plane Rotation Angle for Face Images from Multi-Poses. In *Computational Intelligence in Biometrics and Identity Management (CIBIM), 2013 IEEE Workshop on*, pages 52–57.

Arróspide, J., Salgado, L., and Camplani, M. (2013). Image-based on-road vehicle detection using cost-effective histograms of oriented gradients. *Journal of Visual Communication and Image Representation*, 24(7):1182–1190.

Azeem, A., Sharif, M., Raza, M., and Murtaza, M. (2014). A survey: face recognition techniques under partial occlusion. *Int. Arab J. Inf. Technol.*, 11(1):1–10.

Basu, M. (2002). Gaussian-based edge-detection methods - A Survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C Applications*, 32(3):252–260.

Bay, H., Ess, A., Tuytelaars, T., and Van Gool, L. (2008). Speeded-up robust features. *Computer Vision and Image Understanding*, 110(3):346–359.

Belhumeur, P., Hespanha, J., and Kriegman, D. (1997). Eigenfaces vs. Fisherfaces: recognition using class specific linear projection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(7):711–720.

Berg, T. and Belhumeur, P. N. (2012). Tom-vs-pete classifiers and identity-preserving alignment for face verification. In *British Machine Vision Conference, BMVC 2012, Surrey, UK, September 3-7, 2012*, pages 1–11.

Bicego, M., Lagorio, A., Grosso, E., and Tistarelli, M. (2006). On the use of SIFT features for face authentication. In *Computer Vision and Pattern Recognition Workshop, 2006. CVPRW '06. Conference on*, pages 35–35.

Boser, B. E., Guyon, I., and Vapnik, V. (1992). A Training Algorithm for Optimal Margin Classifiers. In Haussler, D., editor, *Conference on Learning Theory*, pages 144–152. ACM.

Bowyer, K. W., Hollingsworth, K. P., and Flynn, P. J. (2013). A survey of iris biometrics research: 2008-2010.

Brey, P. (2004). Ethical aspects of facial recognition systems in public places. *Journal of Information, Communication and Ethics in Society*, 2(2):97–109.

Cao, X., Wipf, D., Wen, F., and Duan, G. (2013). A practical transfer learning algorithm for face verification. pages 3208–3215. International Conference on Computer Vision (ICCV).

Castrillón-Santana, M., Déniz-Suárez, O., Antón-Canalís, L., and Lorenzo-Navarro, J. (2008a). Face and Facial Feature Detection Evaluation. In *Third International Conference on Computer Vision Theory and Applications, VISAPP08*, pages 167–172.

Castrillón-Santana, M., O. Déniz-Suárez, L. A.-C., and Lorenzo-Navarro, J. (2008b). Face and Facial Feature Detection Evaluation. In *Third International Conference on Computer Vision Theory and Applications, VISAPP08*, pages 167–172.

Chen, D., Cao, X., Wen, F., and Sun, J. (2013). Blessing of dimensionality: High dimensional feature and its efficient compression for face verification. In *CVPR 2013*.

Chen, S.-C., Wu, C.-H., 0001, S.-Y. L., and Hung, Y.-P. (2012). 2d face alignment and pose estimation based on 3d facial models. In *ICME*, pages 128–133. IEEE Computer Society.

Chopra, S., Hadsell, R., and LeCun, Y. (2005). Learning a similarity metric discriminatively, with application to face verification. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01*, CVPR '05, pages 539–546, Washington, DC, USA. IEEE Computer Society.

Chu, B., Romdhani, S., and Chen, L. (2014). 3d-aided face recognition robust to expression and pose variations. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 1907–1914.

Coates, A., Ng, A. Y., and Lee, H. (2011). An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2011, Fort Lauderdale, USA, April 11-13, 2011*, pages 215–223.

Cootes, T. F., Edwards, G. J., and Taylor, C. J. (1998). Active Appearance Models. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 484–498. Springer.

Cootes, T. F., Taylor, C. J., Cooper, D. H., and Graham, J. (1995). Active Shape Models - Their Training and Application. *Computer Vision and Image Understanding*, 61(1):38–59.

Cristianini, N. and Shawe-Taylor, J. (2010). *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods.* Cambridge University Press.

Csurka, G., Dance, C. R., Fan, L., Willamowski, J., and Bray, C. (2004). Visual categorization with bags of keypoints. In *In Workshop on Statistical Learning in Computer Vision, ECCV*, pages 1–22.

Dahmane, M. and Meunier, J. (2011). Emotion recognition using dynamic grid-based HoG features. In *Automatic Face Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on*, pages 884–888.

Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893.

Deng, N., Tian, Y., and Zhang, C. (2012). *Support Vector Machines: Optimization Based Theory, Algorithms, and Extensions*. Chapman & Hall/CRC, 1st edition.

Déniz, O., Bueno, G., Salido, J., and la Torre, F. D. (2011). Face recognition using histograms of oriented gradients. *Pattern Recognition Letters*, 32(12):1598–1603.

Freund, Y. and Schapire, R. E. (1995). A decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. In *Proceedings of the Second European Conference on Computational Learning Theory*, EuroCOLT '95, pages 23–37, London, UK. Springer-Verlag.

Freund, Y. and Schapire, R. E. (1999). A short introduction to boosting. In *In Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, pages 1401–1406. Morgan Kaufmann.

Geebelen, D., Suykens, J., and Vandewalle, J. (2012). Reducing the Number of Support Vectors of SVM Classifiers Using the Smoothed Separable Case Approximation. *IEEE Transactions on Neural Networks and Learning Systems*, 23(4):682 –688.

Gui, J., Sun, Z., Jia, W., Hu, R.-X., Lei, Y.-K., and Ji, S. (2012). Discriminant sparse neighborhood preserving embedding for face recognition. *Pattern Recognition*, 45(8):2884–2893.

Hafiz, F., Shafie, A. A., and Mustafah, Y. M. (2012). Face recognition from single sample per person by learning of generic discriminant vectors. *Procedia Engineering*, 41(0):465 – 472. International Symposium on Robotics and Intelligent Sensors 2012 (IRIS 2012).

Hansen, D. W. and Ji, Q. (2010). In the Eye of the Beholder: A Survey of Models for Eyes and Gaze. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 32(3):478–500.

Hasan, M. K. and Pal, C. J. (2011). Improving Alignment of Faces for Recognition. In *Robotic and Sensors Environments*, pages 249–254. IEEE.

Hinton, G. E. (2002). Training Products of Experts by Minimizing Contrastive Divergence. *Neural Computation*, 14(8):1771–1800.

Hinton, G. E., Osindero, S., and Teh, Y.-W. (2006). A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554.

Huang, D.-Y., Lin, T.-W., Hu, W.-C., and Chen, M.-S. (2011). Eye Detection Based on Skin Color Analysis with Different Poses under Varying Illumination Environment. In Watada, J., Chung, P.-C., Lin, J.-M., Shieh, C.-S., and Pan, J.-S., editors, *International Conference on Genetic and Evolutionary Computing*, pages 252–255. IEEE.

Huang, G. B., Ramesh, M., Berg, T., and Learned-Miller, E. (2007). Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst.

Huang, W. and Mariani, R. (2000). Face Detection and Precise Eyes Location. In *International Conference on Pattern Recognition*, volume 4, pages 722–727, Los Alamitos, CA, USA. IEEE Computer Society.

Ilbeygi, M. and Shah-Hosseini, H. (2012). A novel fuzzy facial expression recognition system based on facial feature extraction from color face images. *Engineering Applications of Artificial Intelligence*, 25(1):130–146.

Jafri, R. and Arabnia, H. R. (2009). A survey of face recognition techniques. *JIPS*, 5(2):41–68.

Jain, A. K. and Klare, B. (2012). Face matching and retrieval in forensics applications. *IEEE MultiMedia*, 19:20–28.

Jain, A. K. and Kumar, A. (2012). Chapter 3 biometric recognition: An overview. *IEEE Trans. on Circuits and Systems for Video Technology.*

Jain, A. K. and Maltoni, D. (2003). *Handbook of Fingerprint Recognition.* Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Jain, L. C., Halici, U., Hayashi, I., Lee, S., and Tsutsui, S. (1999). *Intelligent biometric techniques in fingerprint and face recognition*, volume 10. CRC press.

Jemaa, Y. B. and Khanfir, S. (2009). Automatic local Gabor features extraction for face recognition. *International Journal of Computer Science and Information Security (IJCSIS))*, 3(1).

Jiang, X. and Lai, J. (2015). Sparse and dense hybrid representation via dictionary decomposition for face recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(5):1067–1079.

Joachims, T. (1999). Making large-Scale SVM Learning Practical. In *Advances in Kernel Methods - Support Vector Learning*, pages 169–184. MIT Press.

Kalbkhani, H., Shayesteh, M. G., and Mousavi, S. M. (2013). Efficient algorithms for detection of face, eye and eye state. *Computer Vision, IET*, 7(3):184–200.

Karaaba, M. F., Surinta, O., Schomaker, L. R. B., and Wiering, M. A. (2015). In-plane rotational alignment of faces by eye and eye-pair detection. In *Proceedings of the 10th International Conference on Computer Vision Theory and Applications*, pages 392–399.

Karaaba, M. F., Wiering, M. A., and Schomaker, L. (2014). Machine Learning for Multi-View Eye-Pair Detection. *Engineering Applications of Artificial Intelligence*, 33(0):69 – 79.

Kawaguchi, T., Hidaka, D., and Rizon, M. (2000). Detection of Eyes from Human Faces by the Hough Transform and Separability Filter. In *International Conference on Image Processing*, pages 49–52.

Kim, D. and Dahyot, R. (2008). Face Components Detection Using SURF Descriptors and SVMs. In *Proceedings of the 2008 International Machine Vision and Image Processing Conference*, IMVIP '08, pages 51–56, Washington, DC, USA. IEEE Computer Society.

Kittipanya-ngam, P. and Cootes, T. (2006). The effect of texture representations on AAM performance. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 2, pages 328–331.

Koshiba, Y. and Abe, S. (2003). Comparison of L1 and L2 Support Vector Machines. In *Neural Networks, 2003. Proceedings of the International Joint Conference on*, volume 3, pages 2054–2059 vol.3.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Pereira, F., Burges, C., Bottou, L., and Weinberger, K., editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc.

Kroon, B., Maas, S., Boughorbel, S., and Hanjalic, A. (2009). Eye localization in low and standard definition content with application to face matching. *Computer Vision and Image Understanding*, 113(8):921–933.

Kveton, B. and Valko, M. (2013). Learning from a single labeled face and a stream of unlabeled data. In *Automatic Face and Gesture Recognition (FG), 2013 10th IEEE International Conference and Workshops on*, pages 1–8.

Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pages 2278–2324.

Li, H., Wang, P., and Shen, C. (2010a). Robust face recognition via accurate face alignment and sparse representation. In *Digital Image Computing: Techniques and Applications (DICTA), 2010 International Conference on*, pages 262–269.

Li, Z., Imai, J., and Kaneko, M. (2010b). Robust face recognition using block-based bag of words. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 1285–1288.

Liao, S., Jain, A. K., and Li, S. Z. (2013). Partial face recognition: Alignment-free approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(5):1193–1205.

Lin, S.-H., Kung, S.-Y., and Lin, L.-J. (1997). Face Recognition/Detection by Probabilistic Decision-Based Neural Network. *IEEE Transactions on Neural Networks*, 8(1):114–132.

Lowe, D. G. (2004). Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60:91–110.

Lu, C. and Tang, X. (2014). Surpassing human-level face verification performance on LFW with gaussianface. volume abs/1404.3840.

Lu, C., Zhao, D., and Tang, X. (2013). Face recognition using face patch networks. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 3288–3295.

Lu, J., Tan, Y.-P., and Wang, G. (2011). Discriminative multi-manifold analysis for face recognition from a single training sample per person. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1943–1950.

Maio, D. and Maltoni, D. (2000). Real-time Face Location On Gray-Scale Static Images. *Pattern Recognition*, 33(9):1525–1539.

Makwana, R. M. (2010). Illumination invariant face recognition: A survey of passive methods. *Procedia Computer Science*, 2(0):101 – 110. Proceedings of the International Conference and Exhibition on Biometrics Technology.

Milborrow, S., Morkel, J., and Nicolls, F. (2010). The MUCT Landmarked Face Database. *Pattern Recognition Association of South Africa*. `http://www.milbo.org/muct`.

Montazer, G., Soltanshahi, M., and Giveki, D. (2015). Extended bag of visual words for face detection. *Advances in Computational Intelligence*, 9094:503–510.

Monzo, D., Albiol, A., Sastre, J., and Albiol, A. (2011). Precise eye localization using HOG descriptors. *Machine Vision and Applications*, 22(3):471–480.

Motwani, R. C., Motwani, M. C., Frederick, D., and Harris, C. (2004). Eye Detection using Wavelets and ANN. In *Proceedings of Global Signal Processing Conferences & Expos for the Industry (GSPx)*.

Nordstrøm, M. M., Larsen, M., Sierakowski, J., and Stegmann, M. B. (2004a). The IMM face database - an annotated dataset of 240 face images. Technical report, Informatics and Mathematical Modelling, Technical University of Denmark, DTU, Richard Petersens Plads, Building 321, DK-2800 Kgs. Lyngby.

Nordstrøm, M. M., Larsen, M., Sierakowski, J., and Stegmann, M. B. (2004b). The IMM face database - an annotated dataset of 240 face images. Technical report, Informatics and Mathematical Modelling, Technical University of Denmark, DTU.

Parkhi, O. M., Vedaldi, A., and Zisserman, A. (2015). Deep face recognition. In *Proceedings of the British Machine Vision Conference (BMVC)*.

Perronnin, F., Sánchez, J., and Mensink, T. (2010). Improving the fisher kernel for large-scale image classification. In *Proceedings of the 11th European Conference on Computer Vision: Part IV*, ECCV'10, pages 143–156, Berlin, Heidelberg. Springer-Verlag.

Phillips, P. J., Wechsler, H., Huang, J., and Rauss, P. (1998). The FERET database and evaluation procedure for face recognition algorithms. *Image and Vision Computing*, 16(5):295–306.

Reed, S., Sohn, K., Zhang, Y., and Lee, H. (2014). Learning to disentangle factors of variation with manifold interaction. In *Proceedings of The 31st International Conference on Machine Learning*.

Samaria, F. and Harter, A. (1994). Parameterisation of A Stochastic Model for Human Face Identification. In *Applications of Computer Vision, 1994., Proceedings of the Second IEEE Workshop on*, pages 138 –142.

Saquib, Z., Salam, N., Nair, R., Pandey, N., and Joshi, A. (2010). A survey on automatic speaker recognition systems. pages 134–145. Springer Berlin Heidelberg.

Shekhar, R. and Jawahar, C. (2012). Word image retrieval using bag of visual words. In *Document Analysis Systems (DAS), 2012 10th IAPR International Workshop on*, pages 297–301.

Simonyan, K., Parkhi, O. M., Vedaldi, A., and Zisserman, A. (2013). Fisher Vector Faces in the Wild. In *British Machine Vision Conference*.

Sirohey, S. A. and Rosenfeld, A. (2001). Eye detection in a face image using linear and nonlinear filters. *Pattern Recognition*, 34(7):1367–1391.

Smolensky, P. (1986). Parallel Distributed Processing: Explorations in the Microstructure of Cognition. In Rumelhart, D. E. and McClelland, J. L., editors, *Information Processing in Dynamical Systems: Foundations of Harmony Theory*, volume 1, pages 194–281. MIT Press, Cambridge, MA, USA.

Song, F., Tan, X., Chen, S., and Zhou, Z.-H. (2013). A literature survey on robust and efficient eye localization in real-life scenarios. *Pattern Recognition*, 46(12):3157 – 3173.

Su, Y., Shan, S., Chen, X., and Gao, W. (2010). Adaptive generic learning for face recognition from a single sample per person. In *Computer Vision and Pattern Recognition, (CVPR) 2010 The Twenty-Third IEEE Conference on*, pages 2699–2706.

Sun, Y., Wang, X., and Tang, X. (2014). Deep learning face representation from predicting 10,000 classes.

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

Tan, X., Chen, S., Zhou, Z.-H., and Zhang, F. (2006). Face recognition from a single image per person: A survey. *Pattern Recognition*, 39(9):1725–1745.

ter Haar, F. B. and Veltkamp, R. C. (2008). 3d face model fitting for recognition. In *Computer Vision–ECCV 2008*, pages 652–664. Springer.

Turk, M. and Pentland, A. (1991). Eigenfaces for recognition. *Journal of cognitive neuroscience*, 3(1):71–86.

Vapnik, V. (1998). *Statistical Learning Theory*. Wiley.

Vijayalaxmi and Rao, P. S. (2012). Eye detection using Gabor Filter and SVM. In *Intelligent Systems Design and Applications (ISDA), 12th International Conference on*, pages 880–883.

Viola, P. and Jones, M. J. (2004). Robust real-time face detection. *Int. J. Comput. Vision*, 57(2):137–154.

Wagner, A., Wright, J., Ganesh, A., Zhou, Z., Mobahi, H., and Ma, Y. (2012). Toward a practical face recognition system: Robust alignment and illumination by sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(2):372–386.

Wang, H., Klaser, A., Schmid, C., and Liu, C.-L. (2011). Action recognition by dense trajectories. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3169–3176.

Wei, J., Jian-qi, Z., and Xiang, Z. (2011). Face recognition method based on support vector machine and particle swarm optimization. *Expert Systems with Applications*, 38(4):4390 – 4393.

Wiering, M. A. and Schomaker, L. R. (2014). Multi-layer support vector machines. *Regularization, Optimization, Kernels, and Support Vector Machines*.

Wright, J., Yang, A. Y., Ganesh, A., Sastry, S. S., and Ma, Y. (2009). Robust face recognition via sparse representation. *IEEE Transactions Pattern Analysis Machine Intelligence*, 31(2):210–227.

Wu, M. and Schölkopf, B. (2006). A local learning approach for clustering. In *Advances in neural information processing systems*, pages 1529–1536.

Wu, Y.-S., Liu, H.-S., Ju, G.-H., Lee, T.-W., and Chiu, Y.-L. (2012). Using the visual words based on affine-sift descriptors for face recognition. In *Signal Information Processing Association Annual Summit and Conference (APSIPA ASC), 2012 Asia-Pacific*, pages 1–5.

Xiao, J., Baker, S., Matthews, I., and Kanade, T. (2004). Real-time combined 2d+ 3d active appearance models. In *CVPR (2)*, pages 535–542.

Xu, C., Wang, Y., Tan, T., and Quan, L. (2004). Depth vs. intensity: which is more important for face recognition? In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 1, pages 342–345 Vol.1.

Xu, Y., Li, X., Yang, J., and Zhang, D. (2014). Integrate the original face image and its mirror image for face recognition. *Neurocomputing*, 131(0):191 – 199.

Yan, Y., Wang, H., and Suter, D. (2014). Multi-subregion based correlation filter bank for robust face recognition. *Pattern Recognition*, 47(11):3487 – 3501.

Yang, M.-H., Kriegman, D. J., and Ahuja, N. (2002). Detecting faces in images: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(1):34–58.

Yin, Q., Tang, X., and Sun, J. (2011). An associate-predict model for face recognition. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 497–504.

You-jia, F., Jian-wei, L., and Ru-xi, X. (2010). Robust Eye Localization on Multi-View Face in Complex Background Based on SVM Algorithm. In *International Symposium on Information Engineering & Electronic Commerce*, pages 1–5.

Zhang, X. and Gao, Y. (2009). Face recognition across pose: A review. *Pattern Recognition*, 42(11):2876 – 2896.

Zhou, E., Cao, Z., and Yin, Q. (2015). Naive-deep face recognition: Touching the limit of LFW benchmark or not? *CoRR*, abs/1501.04690.

Zhou, M., Wang, Y., Feng, X., and Wang, X. (2008). A robust texture preprocessing for aam. In *Computer Science and Software Engineering, 2008 International Conference on*, volume 2, pages 919–922.

Zhu, X. and Ramanan, D. (2012). Face detection, pose estimation, and landmark localization in the wild. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2879–2886.

Zhu, Z., Luo, P., Wang, X., and Tang, X. (2014). Deep learning multi-view representation for face recognition. *CoRR*, abs/1406.6947.

Zhuang, L., Chan, T., Yang, A. Y., Sastry, S. S., and Ma, Y. (2014). Sparse illumination learning and transfer for single-sample face recognition with image corruption and misalignment. *Computing Research Repository CoRR*, abs/1402.1879.

AUTHOR PUBLICATIONS

- **Karaaba, M.F.**, Surinta, O., Schomaker, L.R.B., and Wiering, M.A. (2016). Robust Face Identification with Small Sample Sizes using Bag of Words and Histograms of Oriented Gradients Visapp, Conference for Computer Vision Theory and Applications.
- **Karaaba, M.F.**, Surinta, O., Schomaker, L.R.B., and Wiering, M.A. (2015). Robust Face Recognition by Computing Distances From Multiple Histograms of Oriented Gradients In Computational Intelligence in Biometrics and Identity Management (IEEE CIBIM), IEEE Symposium Series.
- **Karaaba, M.F.**, Surinta, O., Schomaker, L.R.B., and Wiering, M.A. (2014). In-Plane Rotational Alignment of Faces by Eye and Eye-Pair Detection. In Computer Vision, Imaging and Computer Graphics Theory and Applications (VISAAP), The 10th International Joint Conference.
- **Karaaba, M.F.**, Schomaker, L.R.B., and Wiering, M.A. (2014). Machine Learning for Multi-View Eye-Pair Detection. Engineering Applications of Artificial Intelligence.
- van de Wolfshaar J., **Karaaba M.F.**, and Wiering M.A. (2015). Deep Convolutional Neural Networks and Support Vector Machines for Gender Recognition, IEEE International Symposium on Computational Intelligence in Biometrics and Identity Management.
- Surinta, O., **Karaaba, M.F.**, Schomaker, L.R.B., and Wiering, M.A. (2015). Recognition of Handwritten Characters using Local Gradient Feature Descriptors. Engineering Applications of Artificial Intelligence.
- Surinta, O., **Karaaba, M.F.**, Mishar, T.K., Schomaker, L.R.B., and Wiering, M.A. (2015). Recognizing Handwritten Characters with Local Descriptors and Bags of Visual Words. in Engineering

Applications of Neural Networks (EANN), The 16th International Conference.

- Surinta, O., Holkamp, M., **Karaaba, M.F.**, van Oosten, J.P., Schomaker, L.R.B., and Wiering, M.A. (2014). A* Path Planning for Line Segmentation of Handwritten Documents. In Frontiers in Handwriting Recognition (ICFHR), The 14th International Conference.

SAMENVATTING

Gezichtsherkenning is een aantrekkelijke methode voor identiteitsherkenning. Hoewel gezichtsherkenning op fundamenteel niveau nog niet is opgelost, trekt het niet alleen de aandacht van meer onderzoekers, maar ook van bedrijven in de commerciële en de overheidssector, vanwege de voordelen boven andere biometrische technieken zoals vingerafdruk-, iris- en stemherkenning.

Het voordeel van gezichtsherkenning ligt in het gemak waarmee het kan worden geïnstalleerd en gebruikt: er is geen kostbaar apparatuur nodig om iemands gezicht te vergelijken met anderen in een gezichtendatabase en er is ook niemand nodig om een persoon te helpen om zijn of haar gezicht te laten zien. Daarnaast is dit ook een belangrijke taak voor veel robottoepassingen. Een robot die gebruikt wordt om patiënten te helpen kan bijvoorbeeld het gezicht van een patiënt registreren zonder extra menselijke interventie.

Desondanks blijft gezichtsherkenning een wetenschappelijk probleem, vooral omdat variantie binnen klassen meestal vergelijkbaar is in grootte met de variantie tussen klassen. Zo lijken afbeeldingen van de gezichten van twee verschillende mensen, genomen in dezelfde pose of zelfde belichting, meer op elkaar dan twee afbeeldingen van het gezicht van dezelfde persoon onder verschillende omstandigheden. Dit feit bemoeilijkt het functioneren van gezichtsherkenningsalgoritmes in het bijzonder bij kleine steekproeven (d.w.z. wanneer er weinig voorbeelden zijn per identiteit) en lage resoluties (d.w.z. minder dan 200x200).

Gezichtsherkenning heeft twee verschillende toepassingsvelden: gezichtsidentificatie en gezichtsverificatie. Bij gezichtsidentificatie is de taak om de correcte identiteit te bepalen van een gegeven gezicht door gebruik te maken van een database met N gezichtsidentiteiten, terwijl gezichtsverificatie gedefinieerd is als het bepalen of twee gezichtsafbeeldingen behoren bij dezelfde persoon. In dit proefschrift bestuderen we gezichtsidentificatie met het probleem van een kleine steekproefgrootte.

De stappen die worden doorlopen bij het identificeren van een gezicht kunnen worden beschreven als gezichtsdetectie en -lokalisatie, gezichtsuitlijning en gezichtsidentificatie. In gezichtslokalisatie is het doel om de locatie van een gezicht precies te bepalen. Dit kan ook een na-proces worden genoemd dat wordt uitgevoerd na gezichtsdetectie (hoewel detectie en lokalisatie soms als verweven kunnen worden gezien).

In dit onderzoek wordt een gezicht gelokaliseerd door precieze detectie van het oog-paar dat volgt uit een gezichtsdetector. Gezichtslokalisatie wordt normaliter gevolgd door een gezichtsuitlijningsproces dat tot taak heeft om het gezicht te normaliseren in termen van 2D/3D rotatiehoeken en belichtingswaarden. Dit is een belangrijke stap om de prestatie te verbeteren. We hebben oogcentra gebruikt om het gezicht uit te lijnen. Na deze stap worden gelokaliseerde en uitgelijnde gezichten doorgegeven aan een algoritme voor gezichtsidentificatie. Het gezichtsidentificatie-algoritme vergelijkt dit met andere gezichten uit een database en geeft de kandidaatsidentiteit met de hoogste classificatiewaarschijnlijkheid. In dit proefschrift zoeken we naar een oplossing voor de eerdergenoemde problemen beperkingen zoals lage resolutie, kleine steekproefgrootte, en gezichten in willekeurige poses en belichting. In de volgende paragrafen vatten we elk van de hoofdstukken in dit proefschrift samen.

Chapter 1: **Introductie**

Dit hoofdstuk geeft een algemene introductie voor gezichtsherkenning, gevolgd door de doelstelling en de bijdrage van deze dissertatie. In de algemene introductie worden allereerst de preprocessingsstappen voor gezichtsherkenning, namelijk gezichtsdetectie, -lokalisatie en -uitlijning kort beschreven. Vervolgens wordt het probleem van gezichtsherkenning gedefinieerd en een aantal van de nieuwste methodes uit de literatuur gepresenteerd. In de doelstellingen worden de redenen uitgelegd voor het uitvoeren van het huidige onderzoek. Daarna worden de bijdrages van dit onderzoek samengevat, en dit wordt gevolgd door een kort overzicht van de dissertatie.

Chapter 2: **Oog-paar detectie voor gezichtslokalisatie**

Oog-paar detectie, waarin beide ogen in één rechthoek worden gevangen, is weinig onderzocht buiten de oog-paar-detector ontwikkeld met behulp van de Viola en Jones methode. In dit hoofdstuk stellen we daarom een nieuwe methode voor de detectie van oog-paren voor, gebaseerd op de nieuwste feature-extractors en een support vector machine (SVM) voor classificatie. Deze feature-extractors zijn afbeeldingsfilters zoals verschil-van-Gaussians (DoG), de Gaborfilter, de lineair beperkte Boltzmann machine (RBM), principale-componentenanalyse (PCA) en genormaliseerde pixelintensiteitswaarden. We vergelijken deze methodes ook met de oog-paar-detector gebaseerd op de Viola-Jones methode. We gebruiken een sliding-window methode om meerdere voorbeelden te verkrijgen waarvan elke afbeelding van een gezicht een oog-paar kan bevatten. Deze voorbeelden worden aan een feature-extractor doorgegeven die ze verder doorgeeft naar een SVM die een voorbeeld classificeert op basis van de aanwezigheid van een oog-paar. We hebben experimenten uitgevoerd om de prestatie van verschillende feature-extractors te vergelijken op drie gestandaardiseerde datasets: IMM, Caltech en Indian. De resultaten laten zien dat de lineaire RBM in de meeste gevallen de hoogste nauwkeurigheid heeft, gevolgd door de PCA- en de pixelintensiteitmethode. Bovendien presteren al onze methoden beter dan de Viola-Jones oog-paar-detector (behalve de DoG-methode op de Indian dataset). Daarnaast vergelijken we ook onze oog-paar-detector die oog-paren in een rechthoek vangt met de oog-paar-detector die oogparen in twee stappen detecteert (een detector voor individuele ogen). Het bepalen van oog-paren in een rechthoek in één stap resulteert niet alleen in nauwkeurigere detectieresultaten (gemiddeld 6% betere herkenning), maar blijkt ook sneller te zijn (ongeveer 3 keer sneller).

Chapter 3: Rotatie-uitlijning met behulp van oog- en oog-paar-detectors

Dit hoofdstuk presenteert een hiërarchisch oogdetectiesysteem dat eerst de locatie van het oog-paar bepaalt waarna het centrum van elk oog wordt gelokaliseerd. In deze methode worden meerdere voorbeelden geconstrueerd door een sliding-windowmethode. Deze voorbeelden worden doorgegeven aan een feature-extractor om featurevectoren te construeren die vervolgens aan een SVM worden gegeven met een radial basis function kernel om de beste oogkandidaat te selecteren. Nadat de centra van de ogen zijn verkregen, wordt de 2D (in-plane) rotatiehoek berekend die vervolgens wordt gebruikt om het gezicht uit te lijnen. We gebruiken twee feature-extractiemethoden in dit hoofdstuk, de RBM en de histogram van georiënteerde gradiënten (HOG), en twee testdatasets: een subset van FERET en de Labeled Faces in the Wild (LFW). Volgens de resultaten van de rotatiehoekevaluatie zijn onze methodes zeer nauwkeurig. We hebben ook het effect van rotatie-uitlijning op gezichtsherkenning vergeleken. De resultaten op basis van de IMM dataset laten zien dat het uitlijnen van gezichten door middel van rotatie voordat het gezichtsherkenningalgoritme wordt toegepast de herkenningsnauwkeurigheid vergroot met 6 tot 8 procent.

Chapter 4: **Gezichtsherkenning door meerdere HOG en afstandsberekening**

In dit hoofdstuk wordt een gezichtsherkenningsalgoritme gebaseerd op afstand geïntroduceerd, dat gebruik maakt van meerdere histogrammen van georiënteerde gradiënten (Multi-HOG) als een feature-extractiemethode voor de individuele afstandsberekening. Dit algoritme maakt gebruik van een afstandsberekeningsfunctie om afstanden voor elk paar gezichten te berekenen. We gebruiken twee afstandsmaten: het gemiddelde van minimale afstanden (MMD) en een afstand op basis van een meerlaagsperceptron (MLPD). Features verkregen van het Multi-HOG filter worden doorgegeven aan een afstandsberekeningsfunctie die zijn uitvoer doorgeeft aan een 1-nearest neighbour classificator voor de uiteindelijke classificatie. Een MLP overeenkomstclassificator is getraind op twee generieke datasets, namelijk IMM en MUCT, die 0 teruggeeft als een paar gezichten behoren tot dezelfde persoon en 1 teruggeeft in andere gevallen. Om uitlijningsfouten te verhelpen introduceren we daarnaast het meest-overeenkomstige-gebieds-selectie-algoritme (MSRS), dat de meest dichtstbijzijnde gebieden van een paar gezichten bepaalt en de coördinaten van deze gebieden teruggeeft aan het gezichtsidentificatie-algoritme. We hebben onze methodes geëvalueerd op twee testdatasets: een subset van FERET en LFW. De resultaten van deze methode zijn vergeleken met een enkele HOG en een aantal van de nieuwste gezichtsidentificatie-algoritmes. Hieruit blijkt dat onze methode beter presteert dan andere methodes: onze beste resultaten zijn 2% hoger op FERET en 1% op LFW, waarbij in beide gevallen gespiegelde leervoorbeelden zijn gebruikt.

Chapter 5: **Gezichtsidentificatie met de HOG-BOW methode**

Dit hoofdstuk presenteert een nieuw algoritme voor gezichtsidentificatie genaamd HOG-BOW, dat histogrammen van georiënteerde gradiënten (HOG) combineert met bag-of-words (BOW) om weinig trainingsvoorbeelden te gebruiken. In deze methode wordt eerst een codeboek geconstrueerd door de k-means-clustermethode dat de centra berekent van vele deelafbeeldingen, die uit elke trainingsafbeelding zijn gesneden. Deze vectoren worden dan doorgegeven aan L2-SVM voor classificatie. De testdatasets die we hebben gebruikt zijn FERET en LFW, net als in het vorige hoofdstuk. Deze methode is vergeleken met HOG en SIFT featurebeschrijvers. De HOG-BOW methode presteerde significant beter dan beide methodes.

Discussiehoofdstuk
Dit is het laatste hoofdstuk waarin concluderende opmerkingen en mogelijke richtingen voor toekomstig onderzoek worden beschreven. In de sectie met concluderende opmerkingen worden onze bevindingen in relatie tot elk hoofdstuk uitgelegd en antwoord gegeven op de onderzoeksvragen. Daarnaast bespreken we hoe onze methodes zouden kunnen worden uitgebreid door gebruik te maken van deep learning.