University of West Bohemia
Faculty of Applied Sciences
Department of Computer Science and Engineering

Doctoral Thesis

Face Recognition under Real-world Conditions

Pilsen, 2014 Ladislav Lenc

Declaration

I submit this doctoral thesis for review and defense in fulfillment of the requirements for the degree of Doctor of Philosophy at the University of West Bohemia in Pilsen, Czech Republic. I declare that this doctoral thesis is completely my own work and that I used only the cited sources.

Pilsen 9. července 2014

. . . .

Ladislav Lenc

Acknowledgements

I would like to express my thanks to Mr. Pavel Král, my supervisor, for his support and for valuable advice during my PhD studies. I would also like to thank to my family and to my wife Nora Lencová for their support and understanding during this studies. Special thanks belong to the ČTK for support and for providing the data for my research.

Abstract

This thesis deals with Automatic Face Recognition (AFR). It is focused on face recognition under real-world conditions. It means recognizing human faces extracted from ordinary photographs. The main goal of this work is proposing a complete face recognition system intended to be used by the Czech News Agency (ČTK) for automatic annotation of photographs. A small subset of the ČTK dataset is annotated by the photo-identities. The remaining part is unlabelled. The goal of the application is to label the unlabelled photographs.

The first challenging task is to prepare a gallery of known faces from the labelled images. We had to consider the fact that the database contains multiple images for each person and the number is not constant. Also the quality of the images is varying and some of them are not suitable for gallery preparation. Moreover, it is possible that one photograph contains more than one person. We thus need to choose the most representative images to ensure the quality of the dataset. Our first contribution is the proposition of an automatic corpus creation algorithm. The faces are detected, extracted from the photograph, aligned and transformed. An important step is the corpus cleaning algorithm that we developed in order to balance the number of images for each person and exclude erroneously labelled faces. Another important contribution related to the corpus creation algorithm is the creation of a novel real-world ČTK face corpus. The corpus is available for free for research purposes.

Since the performance of existing face recognition methods on real-world data is still limited, we had to propose a novel method suitable for our system. The contribution is developing of several Gabor wavelet and Scale Invariant Feature Transform (SIFT) based methods. These methods were tested and evaluated on widely used face datasets (ORL, FERET) as well as on the newly developed ČTK face corpus. Best results were obtained using the SIFT based adapted Kepenekci method. This method was integrated to the system.

Because of the limited accuracy of the face recognition on our real-world data we needed a tool that can verify the recognition results. The solution and our final contribution is proposition of a novel confidence measure approach for the face recognition. The proposed method integrates several separated approaches that determine the possibility that the recognition result is correct.

The final outcome of this work is thus a complete face recognition system capable to handle real-world photographs. Currently, discussions about the deployment of the system are under way.

Keywords: Automatic face recognition, Face recognition system, Gaussian mixture models, Gabor wavelets, Scale invariant feature transform, Speeded-up robust features, Confidence measures, Two-step supervised confidence measure, ČTK, ČTK database

Abstrakt

Tato dizertační práce se zabývá automatickým rozpoznáváním obličejů. Je zaměřena především na rozpoznávání obličejů v reálných podmínkách, což znamená, že obličeje jsou získány z běžných fotografií. Hlavním cílem práce je návrh systému pro automatickou anotaci fotografií. Tento systém bude využíván Českou tiskovou kanceláří (ČTK). Část fotografií v databázi ČTK je anotována (je přiřazena identita osoby). Úkolem systému je anotovat větší, neanotovanou část fotografií.

Prvním krokem je vytvoření galerie (korpusu) známých obličejů z anotované části databáze. Musíme zde brát v potaz, že databáze obvykle obsahuje více fotografií pro jednotlivé osoby. Navíc se počty fotografií u jednotlivých osob liší. Také kvalita snímků je různá a některé z nich nejsou pro přípravu galerie vhodné. V některých případech dokonce fotografie obsahuje více obličejů. Je proto nezbytné z množiny fotografií pro jednoho člověka vybrat nejvhodnější snímky pro přípravu modelu obličeje. Prvním přínosem práce je návrh algoritmu pro automatické vytvoření korpusu. V rámci algoritmu jsou obličeje detekovány, vyříznuty z fotografie, zarovnány a transformovány. Důležitým krokem je námi navržený algoritmus pro čištění korpusu. Ten zajistí vyrovnaný počet obrázků pro jednotlivé osoby a také vyřadí chybně anotované obličeje. Významný přínos, spojený s algoritmem pro čištění, je vytvoření nového reálného korpusu pro rozpoznávání obličejů. Tento korpus je bezplatně k dispozici pro výzkumné účely.

Protože úspěšnost existujících metod pro rozpoznávání obličejů na reálných datech je stále omezená, bylo nutné navrhnout novou metodu vhodnou pro náš systém. Naším přínosem je návrh několika metod založených na Gaborových waveletech a algoritmu Scale Invariant Feature Transform (SIFT). Metody byly testovány na běžně používaných databázích (ORL, FERET) i na nově vytvořené databázi. Nejlepších výsledků dosáhla adaptovaná Kepenekciho metoda založená na algoritmu SIFT. Tato metoda byla použita v našem systému.

Vzhledem ke stále relativně nízké úspěšnosti rozpoznávání na našich reálných datech bylo nutné najít prostředek pro zjištění, zda je výsledek rozpoznávání správný či nikoli. Řešením a naším posledním přínosem je návrh nové míry důvěry, která integruje několik samostatných metod a z nich určuje pravděpodobnost, že výsledek je správný.

Hlavním výsledkem této práce je ucelený systém pro rozpoznávání obličejů. V současné době probíhají jednání o jeho nasazení v prostředí ČTK.

Klíčová slova: Automatické rozpoznávání obličejů, Systém pro rozpoznávání obličejů, Gaussian mixture models, Gaborovy wavelety, Scale invariant feature transform, Speeded-up robust features, Míry důvěry, Dvou kroková míra důvěry, ČTK, ČTK databáze

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 From Bertillon to Face Recognition

To identify people according to their characteristic features is an old idea. It is even nearly as old as the human race itself. Already the prehistoric huntsmen painting their primitive depictions of animals made signatures by printing their palms. It is also confirmed that old Babylonians used fingerprints for identification of merchants. Other ancient nations used for example palm-prints and other forms of primitive biometrics [7].

Significant progress in this field started at the end of the 19th century. The attempts to find reliable method for people identification gave birth to a new scientific field called Biometrics. The term originated from Greek words “bio” and “metrics”. The first impulse to start the research in biometrics came from the needs of crime scene investigation domain. At the end of the 19th century, a French police officer Alphonse Bertillon introduced system for identification of criminals [8]. After the inventor, the system was also called “Bertillonage”. He used a combination of various measurements of body features and also a frontal and profile photographs. The system gave a basis for so called “mug shot” photographs which are used till now in police archives.

Later, the Bertillon system was overcome by the fingerprint classification system [9]. The system was invented already in 1892 by sir Francis Galton. The fingerprints method (modernised and computerized) is also used till today.

Iris is another part of human body used for identification. Already in the 1930s a concept of iris recognition was proposed. The automated iris recognition system were implemented after the 1990 [10].

For people, the face is the most important part of human body used for identification. Therefore, it seems to be natural to recognize people according to their faces by a computer. The era of automated face recognition started in 1960s when Woody Bledsoe developed a semi-automated face recognition system [11]. An important step forward was made by developing the Eigenfaces method [12]. Since that many other face recognition methods have emerged and the results are very promising.

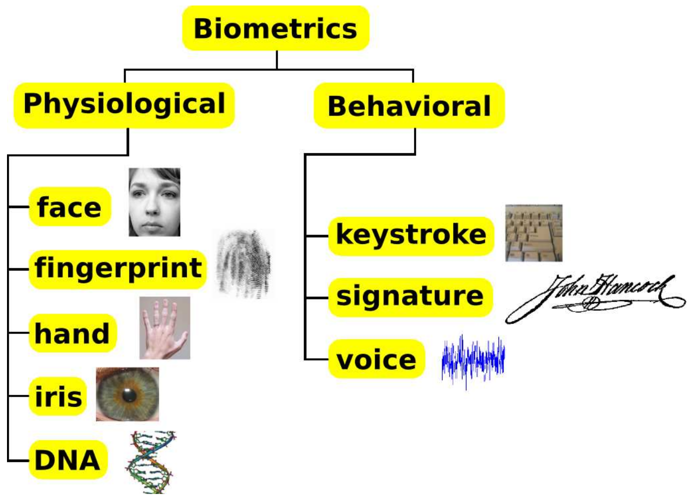

Figure 1.1: Classification of biometric methods [1]

Recently, many biometric identification methods are used. The methods can be divided into two groups: physiological and behavioural. Examples of the physiological characteristics are face, fingerprints, iris, DNA, hand geometry, etc. Behavioural characteristics are, for example gait, typing rhythm on keyboard, voice or signature. Figure 1.1 illustrates the most often used biometric methods.

Using biometrics for identification brings a lot of advantages. Contrary to using passwords or identity cards, there is no need to carry or remember something. Moreover, the card can be lost or stolen and the password forgotten.

From the users point of view, the face recognition is the most comfortable one of biometric method. The fact that no interaction of person to be identified is needed is also very important for automatic surveillance of people. The images may be taken by surveillance cameras and automatically compared with the database. This type of application is not possible using other biometrics. In the case of fingerprints people have to touch a sensor. For iris scan it is necessary to place your eye to the sensor. In the case of the voice-print the identified person is obligated to pronounce some short text, usually a keyword or key-phrase. Using face recognition, only a brief look into the camera is needed. Therefore, the potential of face recognition is really huge.

1.2 Automatic Face Recognition

Automatic Face Recognition (AFR) means automatic identification of a person from a digital image or from a video frame by a computer. This field became intensively studied in the last two decades.

Generally, the face recognition has two main tasks: identification and verification. The goal of the identification is to find the identity of an unknown person. The photograph of the unknown person is compared against the image gallery and, based upon the similarities with gallery images, the most similar one is determined. This one is the identified person. This task is very well expressed by the question "Who is the person?".

The verification task uses a different scenario. The person claims certain identity and the task is to confirm or reject it. The similarity between the face model stored in the database and the persons photograph is determined. Based upon the similarity, the identity is confirmed or rejected. We can use in this task the question "Is the person who he says he is?"

Possible applications for both face recognition tasks are really broad. Historically the first application was identifying of wanted criminals. This application has a great importance also today. Recently, the coverage of public area by surveillance cameras is massive. It is thus only a matter of time when we will be able to identify anybody in these areas. Another important application is an access control. The face can be used instead of identification cards.

Recently a few new applications emerged. One of them is already implemented in social networks. The functions can help you automatically identify your friends in uploaded photographs. Similar functions are used in photo sharing applications.

1.3 Face Recognition under Real-world Conditions

Since the 1990s, the face recognition algorithms achieve very good results. The Eigenfaces [13] were a revolutionary approach and it allowed a very reliable recognition. Still today, many approaches are based on this method or combine the Eigenfaces with other approaches. That is a very good proof of the strengths of this method. But there is a substantial limitation. The successful recognition is possible only with "high quality" images. The variances in images cause a rapid decrease of recognition rates. The most important factors influencing the recognition performance are an imprecise location of the face within the image, variances in pose (frontal image, half profile, profile) and lighting conditions [14].

Later, other methods less prone to the image quality and variances were proposed. For example approaches based upon Gabor wavelets were a significant breakthrough [15]. Another important progress consists in proposing the Scale Invariant Feature Transform (SIFT) [16] approach. These recent AFR methods are already capable to handle (to a certain extent) variances in face images.

One important issue in the AFR field consisted in the lack of a huge common and publicly available face database for a straightforward testing and comparison of the methods. Since the Facial Recognition Technology (FERET) [17] program has started this issue was sufficiently solved. The FERET protocol gave the methodology how to test the algorithms and also the large database of face images. The images are divided into several categories. They have different poses, varying face expressions and also the influence of ageing is considered. The span between acquisition of the images is often more than 1 year. This categorization allows to test the methods on various types of images and thus simulate the recognition in real conditions.

Nevertheless, in the case of totally uncontrolled conditions the recognition is still far more complicated.

In our work, we will focus on the face recognition in real conditions. We have to consider that the photographs are not taken with intention to create quality images for face recognition. Instead, we only take the photograph as is, detect the face and use it for recognition. This important task is necessary for many applications, as for instance processing of images from surveillance cameras, automatic labelling of ordinary photographs containing persons, etc.

To the best of our knowledge, only few scientific works are aimed to the totally uncontrolled environment. Therefore, this task is very challenging and deserves our particular interest.

To perform the face recognition under real-world conditions a complete face recognition system is needed. Except of the face recognition module itself, the system must contain the additional modules. The most important is the face detection one. This module provides the image preprocessing and detects and extracts the faces. The quality of the extracted faces is determining for the accuracy of the subsequent recognition step.

1.4 Motivation

The interest in the face recognition under real-world conditions came from the needs of a real application for the Czech News Agency (ČTK)[1]. ČTK disposes a large database (about 2 millions) of photographs. A certain number of photos is manually annotated (i.e. the photo identity is known). However, it is possible that one photograph contains more people. The remaining (significantly larger) part is unlabelled; the identities are thus unknown. The main task of the proposed application consists in the automatic labelling of the unlabelled photos, which were taken in the completely uncontrolled environment. Note that only few labelled images of every person are available.

Currently, the ČTK annotates the photographs manually. This is a very expensive and time demanding task. Therefore, we would like to reduce the manual work by proposing a fully automatic annotation system.

From the scientific point of view, a design and implementation of such system is very challenging. To the best of our knowledge, this task has not been sufficiently solved yet. Our second motivation thus consists in proposing, implementation and evaluation of such system.

1.5 Contributions

The main scientific contributions of this work can be summarized below:

[1]http://www.ctk.eu

- We proposed and implemented a new face corpus creation algorithm. This algorithm allows to create a face corpus from annotated pictures which may be composed of more people, some background objects, etc. The algorithm was evaluated on the lower quality real ČTK data. The first proposal of this algorithm was published in [7][2].

- We created a new facial corpus from the ČTK data designed particularly for the evaluation of automatic face recognition in real conditions. This corpus is available only for research purposes for free at `<http://home.zcu.cz/~pkral/sw/>` or upon request to the authors.

- We proposed and evaluated a novel face recognition method, called the SIFT based Kepenekci approach, which is based on the SIFT features and adapted Kepenekci matching. This algorithm significantly outperforms the other approaches on the lower quality real data. The method was published in [5] and further evaluated in [6] and [2].

- We proposed and evaluated four unsupervised confidence measures. Two of them are based on the *a posteriori* class probability. The two others are based on the characteristics of the face models. The confidence measure method was originally published in [4]. Then, we proposed and evaluated a composed two-step supervised confidence measure which combines the previously mentioned measures utilizing a multi-layer perceptron neural network. The proposed confidence measure approach allows to successfully decide whether the recognition result is correct in more than 90% of cases. The composed confidence measure method was published in [12] and [9].

- The last and the most important contribution is the design, implementation and evaluation of the fully automatic face recognition system. This system slightly outperforms the other efficient approaches from the viewpoint of face recognition accuracy. It is sufficiently robust on lower quality real data. The system is also available only for research purposes for free at `<http://home.zcu.cz/~pkral/sw/>` or upon request to the authors. The whole system is presented in a new paper that is currently under review [2].

1.6 Thesis Organization

The thesis is organized as follows. The first chapter describes the background of face recognition, our motivation and the main scientific contributions of this work. Chapter 2 summarizes the state of the art of the face recognition and the strictly related fields. Therefore, the face detection and eye detection domains are shortly described. Then the corpora used for evaluation of face recognition approaches are summarized. Chapter 4 describes the procedure used for the corpus creation

[2] The references in this section are related to the Authors publications

task and the related experiments. The following chapter details the face recognition methods that we tested during the design and implementation of our system. Experiments performed to evaluate these methods are a part of this chapter. Confidence measure methods that we proposed together with performed experiments are presented in Chapter 6. Chapter 7 describes the designed face recognition system. The last chapter gives conclusions and proposes some directions of further work.

Chapter 2

Literature Review

This chapter summarizes state-of-the-art methods in the face recognition and strictly related fields. A thorough overview of AFR methods is presented and the strengths and weaknesses of particular methods are mentioned. Some of the approaches which are important for our work are described in more detail.

To be able to implement a fully automated face recognition system also other pattern recognition methods are needed. Particularly face detection and also eye detection tasks are necessary. Therefore, we also present an overview of methods used for these tasks.

The final step of our system consists in verification of the recognition result. It means we would like to determine whether and with what probability is the result correct. For this task we will utilize a confidence measure method. We will thus also mention the most important methods in this field.

Section 2.1 describes the face recognition methods. The following Section 2.2 summarizes the face detection algorithms and Section 2.3 details eye detection approaches. Finally, Section 2.4 shortly summarizes the state of the art in the confidence measure field.

2.1 Face Recognition

2.1.1 Early Face Recognition Methods

The first attempts to recognize faces automatically were made in the 1960s. The initial methods were semi-automatic. A set of facial landmarks was manually determined and normalized measures between these landmarks were used to create the face model.

In 1966, one of the first methods was proposed by Woody Bledsoe [11]. The goal of the application was to select from a database a small subset of faces such that one of them was the wanted face. The system was not fully automated. The coordinates of important facial features were manually labelled by the operator. Examples of features are centres of pupils, eye corners, nose tip, etc. From the features coordinates, 20 distances were computed. These distances were normalized, so that they correspond to the frontal face (elimination of pose, tilt and lean variations). A vector composed of such normalized distances was used in the matching procedure. The nearest neighbour classifier was employed for recognition. This system was highly successful and could even outperform humans in some recognition tasks. Bledsoe also defined the main issue in the face recognition: *The recognition is made difficult by great variability in head rotation and tilt, lighting intensity and angle, facial expression, ageing, etc.* A similar method was designed in the 1970s by Goldstein et al. [18]. In that case, 22 features were used to describe a face.

In 1977, Takeo Kanade [19] developed an approach based on similar measurements. This method determines the feature point positions automatically. The positions are detected using edge maps, signatures and other image processing techniques.

2.1.2 Correlation Method

The simplest and most straightforward method of comparison of two images is to directly compute the correlation between them. The images are handled as one dimensional vectors of intensity values. The correlation of such vectors is used as a similarity measure. The nearest neighbour classifier is used directly in the image space. The images must be normalized to have a zero mean and unit variance. Under these conditions, the influence of light source intensity and camera characteristics are suppressed. However, this method has some substantial weaknesses:

- If the images are taken under varying lighting conditions, the corresponding points in the image space may not be tightly clustered,
- It is computationally expensive,
- Huge amount of memory storage is needed.

Therefore, a practical use of this method is very problematic.

2.1.3 Eigenfaces

One of the first successful approaches to face recognition is based on the Principal Component Analysis (PCA). The name, Eigenfaces, is derived from Eigenvalues. This method was first used by Sirovich and Kirby [12] in 1987 and then in 1991 by Turk and Pentland [20]. Eigenfaces are a statistical method that takes into account the whole image as a vector. The performance of Eigenfaces is very good when images are well aligned and have approximately the same pose. Changing lighting conditions, pose variations, scale variations and other dissimilarities between images decrease the recognition rate rapidly.

The first step of the algorithm is creating a data matrix. The facial images are handled as one dimensional vectors. These vectors are created by concatenating the rows (or columns) of the image matrix. An average vector is computed from all image vectors. One row of the data matrix is then created as a difference between the face vector and the average vector. The covariance matrix is computed by multiplying the data matrix with its transposition. Subsequently, the eigen decomposition of the matrix is realized. Only certain number of eigenvectors which correspond to the largest eigenvalues is used for the face representation. Around 50 Eigenface values are sufficient. The appropriate number depends on the size of the face database. Particular eigenvectors can be seen as face components from which the face is composed. The vector defining the linear combination of eigenvectors is used as a representation of the face. Usually the nearest neighbour rule is employed for feature vectors comparison.

As mentioned above, the dissimilarities in the facial images influence the recognition rate dramatically. In order to overcome this drawback, an extensive preprocessing of input images should be realized. An essential step is to perform the histogram equalization. Then, some transformations for unifying lighting conditions should be made. Even more important is transforming the images so that they were well aligned. The face must be placed at the same position and its proportion must be unified. Also the lean of the head have to be justified so that the eyes are on the horizontal line. Transforming the face pose is also possible but is usually not realized. Fulfilling these conditions makes the algorithm highly accurate and useful. Some of the best performing commercial systems for face recognition are based on this approach.

View Based Eigenfaces

Pentland et al. [13] presented an approach based on the original Eigenfaces. This method is very interesting in two basic ideas:

- Evaluating the method on a large database,
- Addressing the problem of different viewing orientation (pose).

Contrary to the previously developed methods, this one was tested with a dataset containing several thousands of individuals. Two general methods how to extend the classic Eigenfaces in order to handle different face poses are proposed. The first one is to use several face images of one individual, each of them having different pose. Such extended eigenspace is able to encode both identity and viewing orientation. The second one is to create several eigenspaces, each of them representing one viewing orientation. In this case, the first step while identifying a new face is to determine its pose. This is done by calculating the residual description error (measuring the distance from the particular eigenspace). Then, this eigenspace is used to identify a person.

2.1.4 Independent Component Analysis

Independent Component Analysis (ICA) is used for separating a signal into subcomponents. The main goal is to find a linear combination of non-Gaussian data signals that reconstructs the original signal [21]. It is assumed that these components are statistically independent.

The ICA algorithm is usually used in signal processing for signal separation. Another application of the ICA is the feature extraction. There are two different scenarios of using independent component analysis for the face recognition [22]:

- Images are treated as random variables and pixels as observations;
- Pixels are treated as random variables whereas images as observations.

Contrary to PCA, ICA uses higher order statistics (two orders in case of PCA). ICA thus provides more powerful data representation. Authors show in [23] that ICA performs slightly better than PCA approach. The comparison is carried out on the FERET [17] dataset.

2.1.5 Fisherfaces

The Fisherfaces [24] are derived from Fisher's Linear Discriminant (FLD). Similarly to the Eigenfaces approach, the Fisherfaces project an image into another, less dimensional, space. The original dimensionality, which is given by the resolution of the images, is reduced to the number of images (number of distinct classes). The projections of facial images are then compared using some suitable similarity measure. The key point is maximization of the ratio of between-class scatter and within-class scatter. Conversely, the Eigenfaces maximize the total scatter across all images. PCA is thus significantly influenced by the variations in lighting conditions and facial expression, while this drawback is substantially reduced by the Fisherfaces approach, which should be insensitive to changing lighting conditions.

The problem of using this approach for face recognition is that the within-class scatter matrix is always singular. To overcome this problem, the following approach was proposed. The image set is first projected to a lower dimensional space so that the matrix is not singular. PCA is then used to reduce the dimensionality to $N - c$ where N is the total number of training examples and c the number of classes.

2.1.6 Kernel Methods

For both PCA and FLD based methods (Eigenfaces and Fisherfaces) also kernel versions (Kernel Principal Component Analysis (KPCA) and Kernel Fisher's Linear Discriminant (KFLD)) were proposed [25]. The kernel versions address the issue that original methods are based on second order statistics. The methods take into account dependencies among multiple pixels. It allows to capture more information important for the face representation. Both methods are tested on the ORL [26] and Yale [27] databases. The kernel methods reach higher recognition rates than the original ones.

Another interesting method based on the kernel Linear Discriminant Analysis (LDA) was proposed in [28]. The authors present a rotation and illumination invariant polynomial kernel Fisher discriminant analysis. This method combines features extracted by the Discrete Cosine Transform (DCT) and Radon Transform [29]. The significant coefficients of the DCT are used as a feature vector. Further, the kernel Fisher Linear Discriminant is applied to the vectors to increase the discrimination abilities. This approach was tested on FERET, Yale and Olivetti Research Laboratory (ORL) databases. It outperforms other methods such as PCA, KPCA and KFLD.

2.1.7 Adaptive Local Hyperplane

A novel Adaptive Local Hyperplane (ALH) classifier is proposed for the face recognition in [30]. It is an extension of the K-local Hyperplane Distance Nearest Neighbour (HKNN) [31]. ALH approximates the possibly missing instances in manifolds of particular classes by a local hyperplane. When classifying unknown vector first the K nearest neighbours are identified. Based on these K nearest neighbours the local hyperplane is constructed. The class label is assigned to the vector according to the distances between the vector and hyperplanes of each class.

The classifier is tested together with several feature extraction methods, namely 2DPCA, $(2D)^2$2PCA, 2DLDA and $(2D)^2$2LDA. The tests were performed on the ORL and Yale datasets. It is stated there that the ALH classifier outperforms all traditionally used classifiers (Nearest Neighbours, Support Vector Machines, etc.) for this testing set-up. The best recognition results are obtained when Linear Discriminant Analysis (LDA) [32] was used for feature creation.

2.1.8 Genetic Algorithms

Genetic Algorithms (GAs) were also used for the face recognition task as shown for example by Liu in [33]. The author proposes an approach called Evolutionary Pursuit (EP). It is an adaptive dictionary method. The author states that using genetic algorithms, it determines an optimal basis of human face encoding.

In this approach, the facial image is processed in lower dimensional PCA subspace. The genetic algorithm searches for optimal rotation of a basis vector. The rotations are random and the search of the optimal one is done based on a fitness function. It is reported by the authors that this method outperforms both Eigenfaces and Fisherfaces methods.

2.1.9 Trace Transform

In [34], a face recognition approach based on the Trace Transform (TT) is proposed. The Trace transform is a generalization of the Radon transform. It is invariant to image transformations (rotation, scaling and translation).

The image is first transformed into the trace transform space. Thus, the face representation is created. Further, a novel similarity measure is proposed for matching of face representations. The algorithm was tested on the AR [35] face database. This method outperforms the Eigenfaces approach on this dataset.

2.1.10 Linear Regression

An interesting approach using linear regression for the face recognition is proposed in [36]. This approach is based on the assumption that the faces from one class (one individual) are placed in one linear subspace. It assumes multiple training images for each person. Each training image is down-sampled and representing vector is created. The vectors belonging to one individual are put together to create the face model. In the classification step, the image must be also down-sampled and transformed into a vector. The recognized face should be expressed as a linear combination of the model vectors of a relevant class. The estimate is based on the least-square [37] estimation method.

The method was evaluated on the FERET, ORL and Yale datasets. It reaches interesting results on lower quality images (different facial expressions, partial occlusions, etc.).

2.1.11 Active Appearance Models

The Active Appearance Models (AAMs) were proposed for the face analysis in [38]. This approach uses a statistical model of the object shape and grey level appearance. A set of training examples is used to learn the valid shapes. The examples must be labelled, it means the facial landmarks are manually marked. Then, the algorithm tries to match the model to an image. It is done by minimizing the magnitude of the difference vector between the synthesized model and the image. The minimization is performed iteratively.

A view based version of this method was proposed in [39]. Five different models are constructed for different poses (from the left profile to the right profile). These models are sufficient to cover most variations in the face pose.

2.1.12 Neural Networks

Another group of approaches uses Neural Networks (NNs). Several NNs topologies were proposed. One of the best performing methods based on the neural networks is presented in [2]. The image is first sampled into a set of vectors. The vectors created from all labelled images are used as a training set for a Self Organizing Map (SOM). Image vectors of the recognized face are used as an input of the trained SOM. The output of the SOM is then used as an input of the classification step, which uses a convolutional network. This network has several layers and ensures some amount of invariance to the face pose and scale. Figure 2.1 shows an example of a typical convolutional network.

Another approach [40] uses the PCA algorithm for the face representation. Then, an auto-associative neural network is used in order to reduce the feature size to five dimensions. The face recognition is realized, as in the previous case, by a convolutional Multi-layer Perceptron (MLP). This approach achieves good results on a quite simple dataset with manually aligned images of 20 people with no lighting variation, rotation and tilting.

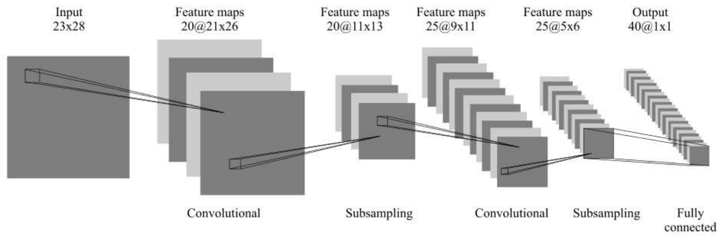

Figure 2.1: An example of the convolutional neural network [2]

The authors use in [41] also the PCA algorithm and neural networks for the face recognition. The Fisher's linear discriminant technique is used for the dimensionality reduction instead of the auto-associative neural network in the previous case. The Radial Basis Function (RBF) neural network is used as a classifier. Experimental results show that this approach achieves very good recognition accuracy and outperforms the majority of the other evaluated methods on the ORL corpus.

However, it is possible to use an MLP network directly with the face images [42]. The intensity values of the pixels are used as the input of the MLP. The main drawback of this approach is the complexity of the network and usually the amount of the training data for a correct estimation of the face models is often not sufficient. Therefore, this approach usually does not achieve interesting results.

2.1.13 Hidden Markov Models

The first method using Hidden Markov Models (HMMs) for the face recognition was proposed in [43]. The face is divided into regions (mouth, nose, eyes, etc.). These regions are then associated with the states of an HMM. The boundaries between the regions are represented by probabilistic transitions between the states. The first step of the algorithm is the image sampling. The image is thereby converted to a 1D sequence of the observations. Usually a left-right and top-bottom sampling direction is used. A square sliding window is employed. First, the image is traversed from the left to the right with a specified step. When the right border is reached, the window is shifted downwards with the same step and traverses back to the left side. This process is repeated till the bottom-right corner is reached.

An alternative technique samples the image with a rectangular window, which has the same width as the image. It is shifted downwards with a specified overlap. The HMM has 8 or 5 states respectively. The algorithm was tested on a dataset containing 5 images of each of the 24 individuals. Indicated recognition rate of this approach is 84%. For comparison, the Eigenfaces were evaluated on the same dataset and the recognition rate of 74% is reported.

Another HMM-based approach is described in [3]. It is stated there, that this method significantly reduces the computational complexity in comparison with the previous methods while the recognition rate remains the same. The image sampling is performed in the same manner as in the above mentioned method. The process

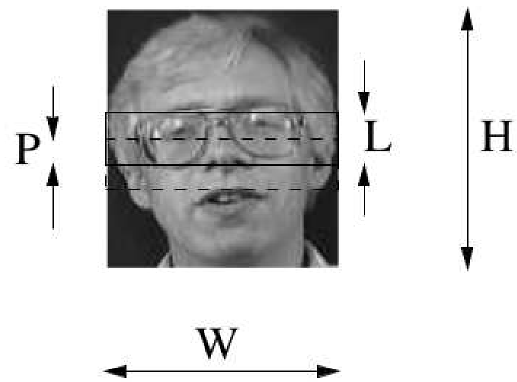

Figure 2.2: Image sampling procedure [3]

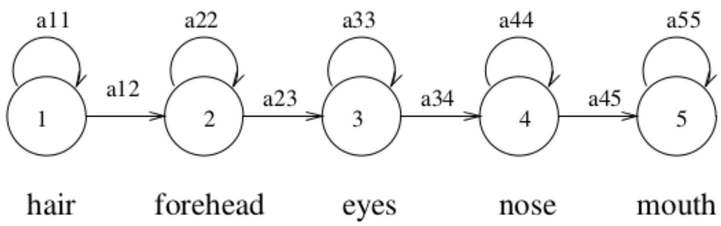

Figure 2.3: Structure of an HMM face recognition [3]

is depicted in Figure 2.2. Instead of using directly the pixel intensity values, a 2D-Discrete Cosine Transform (2D-DCT) is performed. Then, the resulting coefficients are used. The states of the proposed HMM are shown in Figure 2.3.

Another more recent use of the HMM for face recognition is presented in [44].

2.1.14 Support Vector Machines

In [45], an algorithm using Support Vector Machines (SVMs) for face classification is described. The authors propose one component based and two global methods for creation of the vectors representing the face. The SVMs are then used for classification.

The first global approach takes into account all pixel values as the input vector for an SVMs classifier. The second one uses several view-point specific classifiers. The component based method uses separate representations of the important parts of the face and classifies them individually. It is proved that the component based approach is less sensitive to image variations. Another method proposed in [46] uses an SVMs for the feature extraction. This method is derived from the linear discriminant analysis and is called SVM-based Discriminant Analysis (SVM-DA). Employing the SVMs for the feature extraction should enhance the abilities of the method in the case of recognition under uncontrolled conditions. The results on the FERET, AR and CMU-PIE [47] datasets are reported. The authors show that this approach outperforms several other LDA-based methods.

2.1.15 Cost-Sensitive Face Recognition

Zhang et al. propose in [48] an interesting concept of classification of recognition errors according to their cost. Usually when evaluating the face recognition methods, only the recognition error rate is considered. But in some applications, different types of misclassification may have different impact on the whole application performance. The term "loss of the misclassification" is defined and each type of classification error may have different loss value.

Two methods for cost-sensitive classification are proposed: mckNN and mcKLR. The authors state that the proposed methods achieve better performance than other cost-based methods.

2.1.16 Local Binary Patterns

The Local Binary Pattern (LBP) operator was first used for texture representation as presented in [49]. The LBP value in a pixel is computed from the neighbourhood of a pixel and uses the intensity of the central pixel as a threshold. The pixels are marked either 0 or 1 if the value is lower or greater than this threshold. The binary values are concatenated into one binary string and its decimal value is then used as a descriptor of the pixel. The original method uses 3×3 neighbourhood which was later extended to use neighbourhoods of various sizes.

The first application of LBP for face recognition is proposed by Ahonen et al. in [50, 51]. The face is divided into rectangular regions. In each region a histogram of the LBP values is computed. All histograms are then concatenated into one vector which is used for the face representation. A histogram intersection method or Chi square distance are used for vector comparison. A weighted LBP modification is also proposed in this work. It gives more importance to the regions around the eyes and the central part of the face. The reported recognition rate on the FERET dataset [17] reaches 93% for the original method and 97% for the weighted LBP method.

Other LBP based Methods

A modification of the original LBP approach called Dynamic Threshold Local Binary Pattern (DTLBP) is proposed in [52]. It takes into consideration the mean value of the neighbouring pixels and also the maximum contrast between the neighbouring points. It is stated there that this variation is less sensitive to the noise than the original LBP method.

Another extension of the original method is the Local Ternary Patterns (LTP) proposed in [53]. It uses three states to capture the differences between the center pixel and the neighbouring ones. Similarly to the DTLBP the LTP is less sensitive to the noise.

An interesting method which uses uniform patterns is proposed in [54]. The authors state that the histogram bin containing non-uniform patterns dominates among other bins and gives thus too much importance to this bin. Therefore they propose to assign such patterns to the closest uniform pattern. Hamming distance is used for the face comparison.

Some methods also combine other preprocessing tools with the LBP. In [55] Gabor features and LBP are combined. The Gabor features as well as the LBP features are extracted and transformed using PCA. The features are then combined and used as face representation.

Another method called Local Gabor Binary Pattern Histogram (LGBPH) [56] also combines the Gabor wavelet transform and LBP. It first filter the image with a set of Gabor filters and obtains a set of magnitude images. Then the LBP operator is applied to each of the magnitude images.

Local Derivative Patterns

A novel pattern descriptor related to LBP, called Local Derivative Pattern (LDP) is proposed in [57]. The method constructs pattern features from local derivative variations. The advantage over the previously described LBP is its higher order. It thus can represent more information than the LBP. The LDP can be applied both on original grey level images and images processed by Gabor filter. Using the LDP on Gabor filtered images should improve the recognition results. Results on several standard dataset such as FERET, CMU-PIE and Yale are reported.

2.1.17 3D Face Recognition Methods

The aim of the 3D methods is to perform the recognition of faces with any pose. One of such methods is presented in [58]. The algorithm uses linear equations to make out the face description. It should work independently on the facial pose and lighting conditions. A 3D model is used to create images of different pose and illumination from a frontal face image in [59]. Consequently the 2D recognition methods are used for recognition. The 3D methods have a great potential to outperform existing 2D methods. However, the successful implementation of the methods is still problematic. Moreover, the main drawback of this method is the computational complexity of the face fitting process. A challenging scheme is to combine the 3D and 2D approaches.

2.1.18 Methods Based on the Gabor Wavelets

A very important group of approaches is based on the Gabor Wavelet Transform (GWT). These methods were the first examples of feature based methods. The first and probably best known of them is the Elastic Bunch Graph Matching (EBGM). It brought a great success and significant improvement in the face recognition field. Another important method is the Kepenekci method.

Gabor wavelet based methods are very important for our work. Therefore we describe next the fundamental algorithm in detail.

Gabor Wavelets

Gabor filters proposed by Hungarian Dennis Gabor belong to the family of linear filters. The filter is represented by the set of coefficients in a rectangular mask. The response of the filter is obtained by convolving this mask with an image. These filters are orientation and frequency sensitive. They can be used for edge detection.

Gabor filter is basically a sinusoid modulated with a Gaussian. A basic form of a two dimensional Gabor filter is shown in Equation 2.1.

$$g(x, y; \lambda, \theta, \psi, \sigma, \gamma) = \exp\left(-\frac{\acute{x} + \gamma^2 \acute{y}^2}{2\sigma^2}\right) \cos\left(2\pi\frac{\acute{x}}{\lambda} + \psi\right) \tag{2.1}$$

where

$$\acute{x} = x \cos\theta + y \sin\theta \tag{2.2}$$

and

$$\acute{y} = -x \sin\theta + y \cos\theta \tag{2.3}$$

In this equation λ is the wavelength of the cosine factor (the frequency can be used instead of the wavelength, $f = \frac{1}{\lambda}$), θ represents the orientation of the filter and ψ is a phase offset. σ and γ are parameters of the Gaussian envelope. σ is the standard deviation of the Gaussian and γ defines the ellipticity (aspect ratio) of the function.

Alternatively, the filter can be described by Equation 2.4.

$$g(x, y; \lambda, \theta, \psi, \sigma_x, \sigma_y) = \exp\left(-\frac{1}{2}\left(\frac{\acute{x}}{\sigma_x^2} + \frac{\acute{y}}{\sigma_y^2}\right)\right) \cos\left(2\pi\frac{\acute{x}}{\lambda} + \psi\right) \tag{2.4}$$

In this equation σ_x and σ_y specify the dimensions of the Gaussian envelope along the x and y axis respectively. This form was used in our implementation of the filter with required size.

Figure 2.4 shows an example of filter kernels with different orientations, Figure 2.5 illustrates Gabor filter kernels with different wavelengths and Figure 2.6 shows Gabor filter kernels with different sizes of σ of the Gaussian. Notice that the σ value is usually proportional to the wavelength.

$$\sigma = c\lambda \tag{2.5}$$

Figure 2.7 shows an original face image and the convolution of this image with three different filters (orientations 0, 45 and 90 degrees).

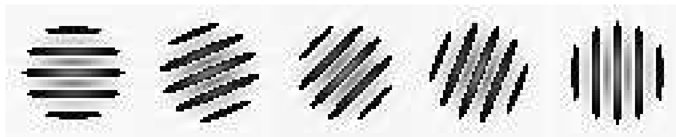

Figure 2.4: Filter kernels with 5 different orientations

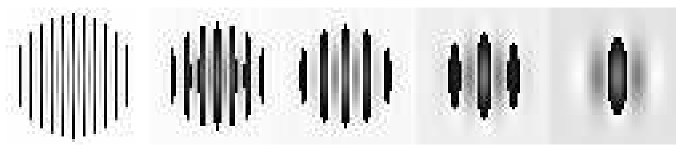

Figure 2.5: Filter kernels with 5 different wavelengths

Figure 2.6: Filter kernels with 5 different sizes (σ of the Gaussian)

Figure 2.7: Original face image (left) and the convolution with three different Gabor filters

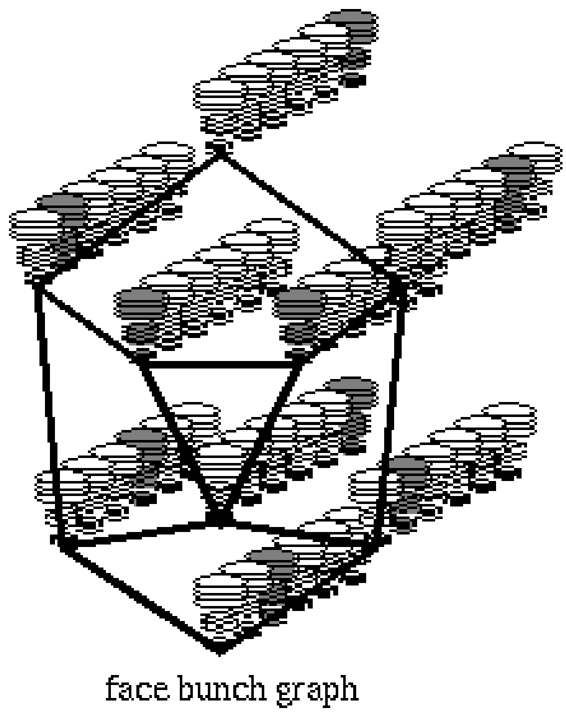

Figure 2.8: Face bunch graph [4]

Elastic Bunch Graph Matching

One of the first AFR approaches based on the Gabor Wavelets is the Elastic Bunch Graph Matching (EBGM) [60]. Initially, a set of manually labelled landmarks is presented to the algorithm. These landmarks are used as an examples to determine the landmark positions in a novel images. The Gabor wavelet convolutions (so called Jets) are computed in the landmark positions and are used for face representation. A "bunch graph" is created from these examples. Each node in the graph contains a set of Jets for one landmark across all of the images. The similarity of faces is determined from the landmark positions and jet values.

The main weakness of this approach is that a human interaction is needed, while storing new faces into the bunch graph. In the first image, the fiducial points must be labelled manually. For following images, an estimation of fiducial point positions is computed and this is manually checked and refined. This is necessary for a certain number of images. The more images are stored in the bunch graph, the less manual checking is needed. Usually, 70 face images are used for a bunch graph. This number is sufficient for automatic registration of novel images. 80 Landmark positions are used (25 landmarks + 55 interpolated points)

While recognizing an unknown image, the face graph must be constructed. This is done automatically based on the created bunch graph. The graph is then compared with face graphs stored in the gallery. The similarity of faces is determined from the landmark positions and jet values. This method is described more in detail in [4].

Other Gabor Wavelets Based Methods

In the last couple of years, several other successful approaches based on Gabor wavelets have been introduced [61]. Some approaches [15] combine the preprocessing with Gabor wavelets with well-established methods such as Eigenfaces, Fisherfaces, etc. These groups of approaches are very efficient and can handle real images because the locally created wavelet features are robust to differences in illumination, distortion, rotation and scaling in the images.

2.1.19 Kepenekci Method

This approach belongs to the group of the Gabor wavelet based methods. But it is important for our work and thus we dedicated a separate section to it. Kepenekci proposes in [62] an algorithm that outperforms the EBGM approach. Moreover, he addresses the main issue of elastic bunch graph matching, manual labelling of the landmarks. In this algorithm, landmark positions are not labelled manually, while obtained dynamically by Gabor filter responses as follows: the image is scanned using a sliding window and the maxima of Gabor filter responses within a window are identified as fiducial points. The number of fiducial points is thus not constant. The feature vectors are calculated in these points (similar as in EBGM). The similarity of two vectors is computed using the cosine similarity.

The size of the sliding window is very important for the performance of this method. It determines the number of detected fiducial points and influences its accuracy. The higher the window size is the less fiducial points are detected. On the other hand, searching larger window needs more computation time. In the comparison stage, the number of fiducial points determines the time needed.

Author states that his method outperforms significantly the Eigenfaces method on the AR [63] database. He further shows that recognition accuracy of the proposed method on the ORL corpus is about 95% and significantly higher than the results of the Eigenfaces, elastic bunch graph matching and neural networks. This approach will be described next in more detail.

Face Representation

The image is convolved with a set of Gabor filters (40 filters with different orientations and wavelengths). 40 wavelet responses R_j, where $j = 1, ..., 40$, are obtained. Each of these responses is scanned with a sliding window. Assume a square window W of size $w \times w$. All possible window positions within the response are evaluated. The center of the window, denoted (x_0, y_0), is considered to be a fiducial point iff:

$$R_j(x_0, y_0) = \max_{(x,y)\in W} R_j(x, y) \tag{2.6}$$

$$R_j(x_0, y_0) > \frac{1}{wi * hi} \sum_{x=1}^{wi} \sum_{y=1}^{hi} R_j(x, y) \tag{2.7}$$

where $j = 1, ..., 40$, wi and hi are image width and height respectively. The feature vector in point (x, y) is created as follows:

$$v(x, y) = \{x, y, R_1(x, y), ..., R_{40}(x, y)\} \tag{2.8}$$

The resulting vector thus contains information about feature point coordinates and values of Gabor responses in this point.

Face Comparison

The cosine similarity [64] is employed for vector comparison. The similarity between two vectors thus takes the values in interval $[0, 1]$. Only the last 40 positions in the vector are considered.

Let us call T a test image and G a gallery image. For each feature vector t of the face T we determine a set of relevant vectors g of the face G. Vector g is relevant iff:

$$\sqrt{(x_t - x_g)^2 + (y_t - y_g)^2} < distanceThreshold \tag{2.9}$$

and

$$S(t, g) > similarityThreshold \tag{2.10}$$

where $S(t, g)$ is the value of cosine similarity of vectors t and g.

In other words, this vector is close enough (i.e. $distanceThresholds$) and similar enough ($similarityThreshold$) to the test vector given.

If no relevant vector to vector t is found, vector t is excluded from the comparison procedure. However, the most similar vector (from the relevant vector set) is used for the face similarity computation. The overall similarity of two faces OS is computed as an average of similarities between each pair of corresponding vectors as:

$$OS_{T,G} = mean\,\{S(t, g), t \in T, g \in G\} \tag{2.11}$$

Then, the face with the most similar vector to each of the test face vectors is determined. The variable C_i says how many times the gallery face G_i was the closest to some of the vectors of the test face T. The similarity is computed as C_i/N_i where N_i is the total number of feature vectors in G_i. Weighted sum of these two similarities is used for similarity measure:

$$FS_{T,G} = \alpha OS_{T,G} + \beta \frac{C_G}{N_G} \tag{2.12}$$

The size of the sliding window is very important for the performance of this method. It determines the number of fiducial points detected and influences its accuracy. The higher the window size is the less fiducial points are detected. On the other hand, searching larger window needs more computation time. In the comparison stage, the number of fiducial points determines the time needed. The above

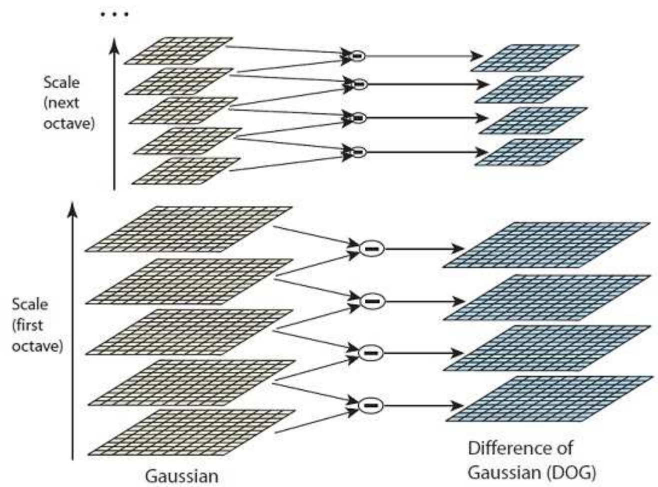

Figure 2.9: Difference of Gaussian filters at the different scales [5]

mentioned threshold *distanceThreshold* also influences the accuracy and the runtime of this method. The smaller the value of this threshold is, the less comparisons are needed and the method works faster.

2.1.20 Scale Invariant Feature Transform (SIFT)

The Scale Invariant Feature Transform (SIFT) [16] proposed by David Lowe has been also used to create the facial features. Using this method leads to high recognition accuracy. It has the ability to detect and describe local features in images. The features are invariant to image scaling, translation and rotation. The algorithm is also partly invariant to changes in illumination. The SIFT algorithm was originally developed for object recognition. The features of the reference and test images are compared using the Euclidean distance of their feature vectors. This algorithm is very efficient and it belongs, in our opinion, to one of the best performing face recognition methods. Therefore, it will be detailed next.

SIFT Algorithm

The SIFT algorithm has basically four steps: extrema detection, removal of key-points with low contrast, orientation assignment and descriptor calculation [65].

To determine the key-point locations, an image pyramid with re-sampling between each level is created. Each image is filtered by the Difference of Gaussian (DoG) filter. The filtering in several scales ensures the scale invariance. Each pixel is compared with its neighbours. Neighbours in its level as well as in the two neighbouring (lower and higher) levels are examined. If the pixel is maximum or minimum of all the neighbouring pixels, it is considered to be a potential key-point.

Figure 2.9 demonstrates the process of creation of the DoG filters at different scales [5].

Figure 2.10: Matched SIFT key-points in two different views of the same object (face)

For the resulting set of key-points their stability is determined. Locations with low contrast and unstable locations along edges are discarded.

Further, the orientation of each key-point is computed. The computation is based upon gradient orientations in the neighbourhood of the pixel. The values are weighted by the magnitudes of the gradient.

The final step is the creation of the descriptors. The computation involves the 16×16 neighbourhood of the pixel. Gradient magnitudes and orientations are computed in each point of the neighbourhood. Their values are weighted by a Gaussian. For each sub-region of size 4×4 (16 regions), the orientation histograms are created. Finally, a vector containing 128 (16×8) values is created.

Figure 2.10 shows how two images of the same object (a face) with varying scale and orientation are matched using the SIFT method.

SIFT for Face Recognition

One of the first applications of this algorithm for face recognition is proposed in [66]. The author takes the original SIFT algorithm and creates a set of descriptors (face features) for every image. The recognized face image is matched against the faces stored in the gallery. The face that has the largest number of matching features is identified as the closest one. The feature is considered to be matched if the difference between similarities of two most similar gallery features is higher than a specified threshold. The author shows that his approach significantly outperforms both the Eigenfaces and Fisherfaces methods on the ORL and Yale databases. The reported recognition rates are 96.3% and 91.7% respectively.

Another interesting approach using the SIFT features in the AFR field is presented in [65]. This method is called Fixed Key-point SIFT (FSIFT). Contrary to the previous method, the SIFT keys are fixed in predefined locations determined in the training step as follows. The key-point candidates are localized in the same manner as in the original SIFT. A clustering algorithm is then applied to this key-point candidate set. The number of clusters is set to 100. The centroids of the clusters are used as the fixed key-point locations. The number of the features thus remains constant. The distance between faces can be computed as a sum of the Euclidean distances between the corresponding features. The reported recognition rate for the Extended Yale database [27] is comparable to the previously described approaches.

2.1.21 Speeded-Up Robust Features (SURF)

Speeded-up robust features [67] is another efficient method for key-point detection and descriptor creation. The integral image [6] is utilized in this approach to speed-up the key-point detection process. The detector called by the authors a "Fast-Hessian" detector is based on the Hessian matrix[1]. Box filters are used as an approximation of second order Gaussian derivatives. Together with the integral image, it allows very fast computation. Contrary to the SIFT, no image pyramid by the sub-sampling is created. Instead, the box filters are up-scaled and applied to the original image.

To ensure the rotation invariance, the orientation is assigned to each key-point. The computation is based on the circular neighbourhood of the key-point. The resulting SURF descriptor is a vector of the length of 64.

Also an upright version of SURF (U-SURF) was proposed. It does not compute the orientation (is not orientation invariant) and simplifies and accelerates the computation process.

Recently, the SURF features were also utilized for face recognition. One of the first applications of SURF in the face recognition domain was proposed in [68]. It applies a geometrically constrained point matching scheme. It matches only key-points inside corresponding regions in the compared images which also reduces the computational coasts. A point pair with minimal distance within given rectangular surrounding of key-point is found. The difference between this distance and the distance of the second nearest key-point is calculated. If the difference is higher than a predefined threshold, the point pair is considered to be matched. As a similarity measure, a number of the matched key-points is used. If the number of matched points is lower than a predefined threshold, the matching is considered as not reliable. In this paper, several variants of SURF features are evaluated. The reported rates for the FERET database range from 95.2 to 96.5%.

In [69] another method using SURF features is described. In this paper, a grid-based feature extraction is proposed. It means that the descriptors are extracted in points on regular grid instead of using the key-point detector. Many combinations of feature extraction methods and matching schemes are evaluated. It also compares the use of SIFT and SURF features. Interesting results on AR-Face and CMU-PIE datasets are reported. Surprisingly, the best results are reached using the Upright version of SURF (U-SURF).

In [70] a SURF based face recognition approach combined with cell similarity [71] is proposed. Different cell division strategies are evaluated. Very good results on the ORL and on author's own database are reported.

[1] a square matrix of the second order partial derivatives of a given function

2.1.22 Face Recognition Systems

The main goal of our work is an implementation of a complete face recognition system. Unfortunately, as already shown above, a lot of papers presented in the face recognition domain concentrate only on the recognition task itself. Therefore, we summarize some systems where also additional steps are described.

One example of the system that addresses the issue of the imprecisely localized faces is proposed in [72]. The system compensates the face position and also solves partial occlusion and different facial expression. Only one training example per person is used.

A complete face recognition system is described in [73]. The training images are well aligned (acquired in controlled conditions) whereas the recognized images are real-world photos. The system is based on the Sparse representation and classification (SRC) [74] algorithm. It achieves very good results on the FERET database.

Another face recognition system is presented in [75]. Authors localize the face in the images and then they compute the facial features. Their face recognition algorithm is based on the EBGM, but the fiducial points are detected completely automatically. The system is evaluated on the FERET corpus. Authors show that their system has recognition scores comparable to the elastic bunch graph matching.

For additional information about the face recognition, please refer to the survey [14]. Notice that the authors of this survey mention also some commercial face recognition systems. Unfortunately, neither the system architecture nor the approaches used are usually reported. Moreover, these systems are not evaluated on the standard face datasets and it is thus impossible to compare them with our system.

2.2 Face Detection

Face detection is one of the fundamental techniques that allow human-computer interaction [76]. It is not the main goal of our work but discovering and localizing a face within an image is a necessary step before another facial analysis algorithm (e.g. face identification and verification) can be applied. Thus, the face detection algorithms are of a great importance. Therefore, we will review the most important face detection methods in this section.

The first task of a face detection algorithms is determining whether an image contains faces or not. Then, the position of the face and its size is located. What makes this task difficult are variations in size and pose of the faces and also in lighting conditions. In [77], the face detection methods are divided into 4 categories.

- Knowledge-based,
- Feature invariant,
- Template matching,
- Appearance-based.

Knowledge based methods encode human knowledge about the face. The facial features (eyes, nose) and relationships among them are described and encoded into rules. Feature invariant algorithms seek features that are invariant to pose, viewpoint and lighting conditions. These are then used for face localization. Template matching uses face patterns to describe the face. Correlations between patterns and input image in various locations are computed and used for face detection. The last type, appearance-based methods, uses a set of training images to learn the face template. The learned models are used for face detection. In the following part, two seminal methods for the face detection are described.

2.2.1 Neural Network-Based Face Detection

In [78], neural networks are used for face detection. The algorithm detects frontal faces in gray-scale images. The system has two stages. The first one uses a system of neural network-based filters. The image is filtered at several scales and the regions which may contain a face are identified. Then, an arbitrator is employed which merges results of individual filters.

The filters are applied on 20×20 pixel regions in the image. Their output is -1 or 1 (1 denotes presence of a face). Each possible location of such 20×20 window is scanned. This is performed at several scales. The image is gradually sub-sampled with factor 1.2 and scanned in the same manner. The key factor of this approach is the training phase of the neural networks. It is difficult to collect typical non-face examples. This is usually done by generating random images.

After the first step, multiple detections at some positions are found. Also some false detections are present. Therefore, the arbitration must merge the multiple detections and eliminate the false alarms. The reported detection rate ranges from 77,9% to 90,3% with small number of false alarms.

2.2.2 Viola-Jones Algorithm

One of the first widely used methods and de facto a standard for the face (and other objects) detection is the Viola-Jones algorithm. It was proposed in [6]. The algorithm uses boosted cascade of simple features for object detection. Figure 2.11 shows example of features used for classification.

For face representation, a new structure called *integral image* is used. It allows effective computation of facial features used for the detection. A point $[x, y]$ in the integral image is computed as a sum of all pixels intensities in the original image placed in the rectangle determined by the coordinates $[0, 0]$ and $[x, y]$.

$$ii(x, y) = \sum_{\acute{x} \leq x, \acute{y} \leq y} i(\acute{x}, \acute{y}) \tag{2.13}$$

$i(x, y)$ is the intensity in the original image and $ii(x, y)$ is the value in the integral image.

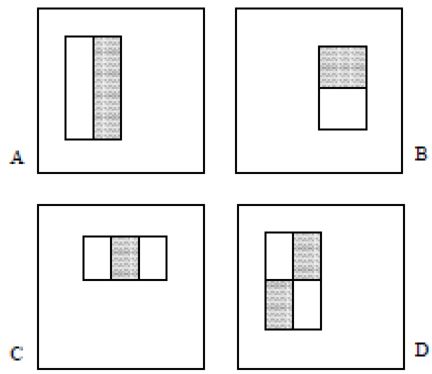

Figure 2.11: Example of features used in Viola-Jones algorithm [6]

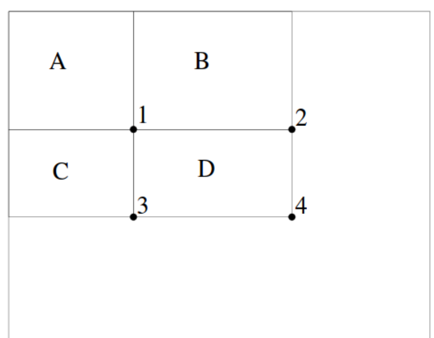

Figure 2.12: Computing sum of a rectangle using the integral image [6]

Using such image representation, sum of any rectangle in the image can be computed from 4 values. The rectangle features (usually computed from 2 to 4 rectangles) can thus be computed very quickly (see Figure 2.12).

In the next step of the algorithm a variant of AdaBoost [79] method is used to select an appropriate small set of features and to train the classifier. The goal is to choose a set of features that can form an effective classifier.

After the features determination, a cascade of classifiers is constructed. The idea is, that the smaller and more effective classifiers can reject the majority of negative sub-windows. Then, more accurate and complicated classifiers are applied to the non-rejected sub-windows. The final cascade has 38 stages and more than 6000 features are used. The algorithm has very good results and became one of the most frequently used algorithms for object detection. A great advantage of this approach is its speed.

2.3 Eye Detection

Detection of facial features is very important in many computer vision applications. Usually, features like nose, mouth and especially eyes are localized. Eyes are considered to be the most salient features in the face [80]. The position of eyes can be used to determine the rotation of the face. According to the distance between the eyes, also the size of the face can be computed. A precise localization of eyes is of great importance in our system. According to the results of the eye detection we are able to confirm the results of face detection. Moreover, we can transform the face so that it was more suitable for the face recognition algorithms.

Localization of eyes is a necessary step in the face recognition algorithms [61]. In order to identify the pose or tilt, other facial features such as nose tip or mouth are needed. Transformation based on positions of such features can substantially increase the performance of face recognition algorithms.

There are two categories of eye detection algorithms. The first group of methods is called “active” [81]. These methods use special types of infra-red illumination while acquiring the photos. It helps to localize the pupils. Active methods are very precise. Substantial drawback is the need of special light sources while photographing. The need of such special conditions during acquisition of the photos decreases the usability in many applications.

The other type of methods, passive, works with common photos. No special lighting conditions or other prerequisites are considered while taking the photographs. The passive methods can be further divided according to the type of used images. In the case of gray-level images, only intensity values can be used. Compared to the methods which use color images, there is less information. The passive methods use image gradients, projections and templates. Also Gabor wavelets can be used for this task [81]. We will focus on methods that operate with gray-scale images.

2.3.1 Eye Detection Approaches

Some of the first eye detection methods used the Hough transform [82]. The eye was modelled by a circle for the iris and an ellipse for the sclera. First, the Sobel operator was applied on the image. Then, an edge image was constructed using thresholding. Some additional constraints were used for eye positions (only the upper half of the image etc.).

Another proposed method [83] uses deformable templates to locate the eyes. The templates are able to deform, rotate and translate in order to best fit the features in the image. The eye template is described by a set of several parameters. This template is fitted to the image in the sense of minimizing the energy function. Unfortunately, the accuracy of this approach is strongly dependent on the starting position of the template. In some cases, the algorithm fails in distinguishing of eyes and eyebrows. Moreover, the algorithm is computationally expensive.

The next group of approaches [80] uses a generalized symmetry operator to find the eyes and mouth. The motivation of such an approach is the natural symmetry of human face. Special symmetry operator finds the points with high value of symmetry measure. The reported detection rate reaches 95%.

Also a well-known Eigenface [12] approach can be used to detect facial features. Instead of Eigenfaces, other eigenspaces such Eigeneyes, Eigenmouth etc. are used.

The image is searched hierarchically at different scales. Eye detection rate is around 95%.

The last mentioned type of eye detection algorithms is based on Gabor wavelets [81]. In the recent studies, this type of approach appears very often. The detection rate of these algorithms is very high.

In the next sections, some eye detection algorithms are described more in detail.

2.3.2 Linear and Non-linear Filtering

The method presented in [84] uses a linear and non-linear filtering to find the eye positions. This method involves the detection of the face in an image. It is done by thresholding and seeks a flesh-tone region. Then an approximate size and position of eyes is estimated. After the face detection phase, two possible approaches for eye detection are proposed. The first one uses linear filters based on Gabor wavelets.

The detection process is performed on gray-level images. Each eye is modelled by 4 Gabor wavelets. Instead of using the real part of the wavelet as is usual, only the imaginary part is employed. It is considered to be more suitable for finding the boundary between the sclera and the flesh region. To find the dark circular region of iris, Gaussian filter is used. For better handling the varying illumination, the filter is divided into three separate parts: Left opposing wavelets, Right opposing wavelets (Differently oriented Gabor filters for finding eye borders) and the Central Gaussian (models the iris). Facial image is separately convolved with each of these three filters. A point in the convolution result is a potential feature point if its value exceeds predefined threshold. If a point exceeds relevant thresholds in all filter responses, it is classified as a feature point. Some additional post-processing is made to eliminate false alarms. Detection rate of this method is very high but there is also a lot of false alarms.

The second approach uses a non-linear filtering. It detects eye corners in color face images. The color gives additional information. Information about normalized color distribution of the flesh tone, obtained in the face detection phase is taken into account. Also the color distribution of the sclera region is determined.

This method gives slightly lower detection rate but no false alarms are found.

2.3.3 Real Time Eye Detection

In [85], another method is presented. The algorithm uses iris geometrical information to find the candidate eye regions. The symmetry is used to find the pair of eyes. It is stated that there are no limitations in the background, skin color segmentation etc.

This work is based on a simple idea that the iris is always darker than the sclera. This fact makes the identification of iris border quite straightforward. It is counted with different radii of the iris within specified interval. The detection of circular regions is based on the Hough transform. First, the edge image is constructed. It is then convolved with the eye masks. The candidate eye locations are assumed to be the maxima of the convolution result. Once a candidate region is found, the other eye is sought in regions where it should be placed according to the symmetry of the face. A similarity between these two regions is computed. If the similarity exceeds specified value, the two regions correspond to the eyes. If not, other initial position is chosen and the procedure is repeated.

The recognition rate of 96% is reached if the eyes are open. For obvious reasons, the rate dramatically decreases when the eyes are closed. In such a case, the reported rate is only 45%.

2.3.4 Intensity Filtering and K-means

In [81], intensity filtering and k-means method are used for eye detection. The method is also based on the assumption that the pupils are darker than the eyeballs. It searches the eyes in two stages. The first stage consists of image enhancement, Gabor transform and k-means clustering. These three steps are used to roughly localize the eyes. The second stage localizes eye centres more precisely. Two special neighbourhood operators are employed in this task.

The first step of the algorithm is image contrast enhancing. The approximate skin region color (intensity) is determined first. This value is further used for the contrast enhancing.

Then, Gabor transform is applied on the enhanced image. The image is convolved with 40 Gabor filters (5 scales and 8 orientations). Only the real part of the wavelet response is considered. Out of these 40 responses, only 3 are chosen. A reference image is created as a linear combination of these 3 images.

The next step is the cluster analysis. Two square regions which should contain the eyes are estimated. The cluster analysis is performed on each of the regions. As a result, we obtain a black image with white patches. One of the patches is selected and its center is taken as an estimate of eye position.

Based on the estimated positions, pupil centre location is searched. Two rectangular regions with centres in the estimated eye positions are cut from the grayscale image. Two operators of sizes 3×3 and 5×5 are applied on these regions. The pupil centres are determined as the minima of resulting images.

Detection rates of 97% and 96% on two face databases are reported.

2.3.5 Radial Symmetry Transform

This approach is described in [80]. It is based on generalized symmetry transform. An object is considered to be symmetric if it is invariant to certain symmetry operation. The generalized symmetry transform assigns a symmetry measure to each point of the image.

The eye detection process involves preprocessing, radial symmetry transform and eye localization. The preprocessing is done by filtering the image with the 5×5 Gaussian filter. It reduces the noise and variations in lighting conditions. Then, two candidate aye regions are selected. The radial symmetry transform is applied on these candidate regions. Further, the symmetry map is filter by a 5×5 mean filter to suppress noise. Thresholding is then performed to detect high symmetry regions. The eye position is considered to be the centre of this region.

This algorithm was tested using the BioID database. The detection rate of 98,6% is reported.

2.4 Confidence Measures

Confidence Measure (CM) is used as a post-processing of the recognition to determine whether a recognition result is correct or not. The incorrectly recognized samples should be removed from the recognition set or another processing (e.g. manual correction) can be further realized. This technique is mainly used in the automatic speech processing field [86, 87, 88] and mostly based on the *posterior* class probability. However, it can be successfully used in another research areas as shown in [89] for genome maps construction, in [90] for stereo vision or in [91] for handwriting sentence recognition.

Another related confidence measure approach is proposed by Proedrou et al. in the pattern recognition task [92]. The authors use a classifier based on the nearest neighbours. Their confidence measure is based on the algorithmic theory of randomness and on transductive learning.

Unfortunately, only few work about the confidence measure in the face recognition domain exists. Li and Wechsler propose a face recognition system which integrates a confidence measure [93] in order to reject unknown individuals or to detect incorrectly recognized faces. Their confidence measure is, as in the previous case, based on the theory of randomness. The proposed approaches are validated on the FERET database.

Eickeler et al. propose and evaluate in [94] five other CMs also in the face recognition task. They use a pseudo 2-D Hidden Markov Model classifier with features created by the Discrete Cosine Transform (DCT). Three proposed confidence measures are based on the *posterior* probabilities and the other two are based on the ranking of the recognition results. Authors experimentally show that the *posterior* class probability gives better results for the recognition error detection task.

2.5 Conclusions

In this chapter we first summarized and briefly described a broad spectrum of face recognition methods. This summary is certainly not complete but it gives, in our opinion, a representative overview of existing approaches. It allowed us also to choose suitable candidates for our system. Primarily, we concentrated on methods that give good results on low quality data. From the relatively high amount of described approaches we chose the Gabor wavelet and SIFT based methods as possible candidates. These methods are thus described more thoroughly.

Then we described methods necessary for image preprocessing before the face recognition step. For both face and eye detection we consider the Viola-Jones algorithm as a good candidate.

Finally, several state of the art confidence measure methods are reviewed.

Chapter 3

Corpora

In this chapter, we summarize face datasets that are most frequently used in the face recognition field. The face datasets are necessary for testing and evaluation of face recognition algorithms. Standardized databases also allow a straightforward comparison of particular face recognition approaches. Therefore, we present a list of such databases to give an overview and to allow the reader to choose an appropriate one for his experiments. At the end of the list we describe the databases we used for evaluation of our face recognition methods.

3.1 BANCA Database

BANCA Database[1] is a large multi-modal dataset. It was created to evaluate multi-modal verification systems. Therefore, it contains also audio records in addition to the face images. The dataset intended for face recognition evaluation (called *English images*[2]) contains 6240 images of 208 individuals. The images were taken under different conditions ranging from controlled to adverse. The *png* files are of size 720×576 pixels. The *English images* dataset costs £100 for academic purposes and £200 for industry.

So called BANCA protocol is associated with the database. This protocol defines for instance the training and testing parts. This standardization allows a straightforward comparison of different methods.

Figure 3.1 shows three example images from the BANCA database. For additional information, please refer to [95].

3.2 Multi-PIE Database

The CMU Pose, Illumination and Expression (PIE) Database [3] was created at the Carnegie Mellon University. The main reason of creating such a dataset was covering

[1] http://www.ee.surrey.ac.uk/CVSSP/banca/
[2] It is named "English", because of the audio records are associated with the images
[3] http://www.multipie.org/

Figure 3.1: Three example images from BANCA image database

Figure 3.2: Three example images from the Multi-PIE image database

the differences in conditions during image acquisition. The key factors are the pose, the illumination conditions and the facial expression. Using 21 flashes, 43 different lighting conditions are simulated. Four facial expressions are captured: neutral, smile, blink and talk. 13 cameras in different positions were use to record the face in 13 different poses. The combination of these variations gives 2236 possibilities for each person. The images of 68 individuals were taken during October and December 2000. The total number of images is 40000. The image size is 640×486 pixels. The database is available for research purposes under a license from Carnegie Mellon University. The prize is \$375 for delivery in US and \$425 for international delivery.

Figure 3.2 shows three example images from the Multi-PIE database. For further information about this database, please refer to [96].

3.3 The AR Face Database

Another face dataset, called the AR Face Database [4], was created at the Univerzitat Autonòma de Barcelona. This database contains more than 4,000 color images of 126 individuals. The images are stored in the *raw* format and their size is 768×576 pixels. Only frontal faces are available. The individuals are captured under different lighting conditions and with varying expression. Another characteristic is a possible presence of glasses or scarf. The database is publicly available. Figure 3.3 shows four example images from the AR face database. For additional information, please refer to [35].

[4] http://www2.ece.ohio-state.edu/ aleix/ARdatabase.html

Figure 3.3: Four example images from the AR face database

Figure 3.4: Four example images from the Yale face database

3.4 The Yale Database

The Yale database[5] originated at the University of California, San Diego. This dataset contains several subsets of images. The main dataset contains 165 images of 11 individuals stored in *gif* format.

Yale database B contains images of 10 individuals. The images are taken under 576 different viewing conditions. 9 different poses and 64 different illuminations are used. The total amount of images is thus 5,760.

Extended version of Yale database B contains images of 28 people. 576 images for each resulting in 16,128 images in total. The size of images is 640×480 pixels. This database is freely available for research purposes. Figure 3.4 shows four example images from the Yale face database. For additional information, please see [97].

3.5 The BioID Database

The database[3] was created at the University of Erlangen. Its primary purpose is to offer a possibility to compare face detection algorithms. The images are taken under "real world" conditions and vary in the face size, illumination and background. This database contains 1,521 images of 23 individuals. The dimension of the images is 384×286 pixels. All images are stored in the *pgm* format. Each face image is associated with a text file containing eye positions. The database is available for editorial, informative or educational purposes. Figure 3.5 shows three example images from the BioID face database. Additional information could be found in [98].

[5] http://vision.ucsd.edu/content/yale-face-database

[3] http://www.bioid.com/download-center/software/bioid-face-database.html

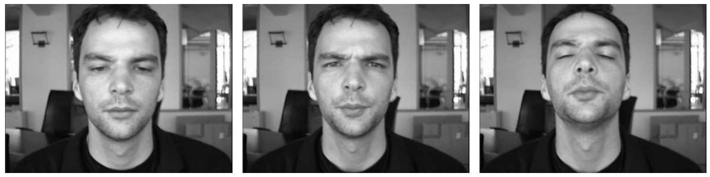

Figure 3.5: Three example images from the BioID face database

Figure 3.6: Three example images from the Labeled faces in the wild face database

3.6 Labeled Faces in the Wild

The Labeled faces in the Wild (LFW)[6] is a relatively new dataset. It was created in 2007 in order to allow testing of face recognition methods in unconstrained environment. It contains over 13,000 images of 5,749 people in resolution 250×250 pixels which were collected from the web. The images are stored in the *jpg* format. 1680 people has at least two distinct images and are used usually for face recognition experiments. The face images were detected by the Viola-Jones algorithm [6].

The database contains four different sets of images for testing. The first one is the original dataset. The other three sets are composed of images obtained using different methods of alignment. The proposed testing protocol concentrates mainly on the face verification task, it means to decide whether two images belong to the same person or not. The database is publicly available. Figure 3.6 shows three example images from the LFW face database. For further description of this database, please refer to [99].

3.7 YouTube Faces

This database [100][7] is composed of video sequences containing faces. It was created from videos collected on the YouTube. The goal was to create a dataset for comparing video based face recognition methods in uncontrolled environment.

This database contains 3,425 videos of 1,595 different people. Similarly as the LFW database a pairwise test scenario is proposed. Together with the database also a set of benchmark test is published. Moreover, descriptor encodings of faces present in the videos are provided. Figure 3.7 shows three example images from the

[6]http://vis-www.cs.umass.edu/lfw/
[7]http://www.cs.tau.ac.il/ wolf/ytfaces/

Figure 3.7: Three example images from the YouTube database

YouTube database.

3.8 The Database of Faces

This database was created at the AT & T Laboratories[1] and it was formerly known as the ORL database. For simplicity, we still refer it as the ORL database in the rest of this work. The pictures of 40 individuals were taken between April 1992 and April 1994. There are 10 pictures for each person. The size of the pictures is 92×112 pixels. Every picture contains just one face. They may vary due to three following factors: 1) time of acquisition; 2) head size and pose; 3) lighting conditions. The images have black homogeneous background.

Figure 3.8 shows four example images from the ORL database. For further description of this database, please refer to [26]. This database is publicly available.

Figure 3.8: Four example images from the ORL database

3.9 Stirling Database

This database[8] was developed at the University of Stirling. It is intended primarily for a psychological research. The database is divided into several subsets.

The most commonly used subset is named Stirling faces. It contains grayscale images of the 36 individuals (9 images for each person). The images have different orientations (frontal image, half profile and profile), for each orientation three images are available. Another difference consists of the facial expression. The size of images is 280×365 pixels and they are stored in the *gif* format. The dataset is publicly available. Figure 3.9 shows four example images from the Stirling database. For further description of this database, please refer to [62].

[1] http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html

[8] http://pics.psych.stir.ac.uk/

Figure 3.9: Four example images from the Stirling Psychological image collection

3.10 FERET Database

The FERET database[9] is connected with the Face Recognition Technology (FERET) program which started in 1993. The program was managed by the Defence Advanced Research Projects Agency (DARPA) and the National Institute of Standards and Technology (NIST). The main goal of this project was to support and motivate development of new face recognition technologies. The program consisted of the three parts:

- Sponsoring research
- Creating the FERET database
- Performing the FERET evaluations

Three phases of the program were determined. The goal of the first phase was to establish a baseline for face recognition algorithms. In the phases 2 and 3 the algorithms were developed.

The creation of the FERET database was a very important contribution of this program, because using a standardized image set for evaluation of new algorithms is crucial for further development. The original grayscale dataset was collected between December 1993 and August 1996. In 2003, DARPA released the colour version with high resolution images.

The grayscale corpus consists of 14051 8-bit images of 1,199 individuals. The resolution of the images is 256×384 pixels. The images are divided into the following categories according to the face pose: frontal, quarter-left, quarter-right, half-left, half-right, full-left and full-right, and are stored in the *tiff* format. The images are also grouped into several probe sets. The main probe sets of the frontal images are summarized in Table 3.1.

There are usually only a few seconds between the capture of the gallery-probe pairs in the *f** sets. The individuals in the *fb* set differ in facial expressions, while the images in the *fc* set differ in illumination conditions. The images in the *dup1* probe set were obtained over a three year period and the *dup2* set is a sub-set of the *dup1*. Only one image per person is available.

Figure 3.10 shows some example images from the FERET database.

[9] `<http://www.itl.nist.gov/iad/humanid/feret/feret_master.html>`

Table 3.1: Image numbers in the the main frontal probe sets of the FERET dataset

Type	**Images no.**
fa	1196
fb	1195
fc	194
dup1	722
dup2	234

Figure 3.10: Three example images of one person from the FERET face corpus (fa, fb and fc - *left* to *right*).

3.11 ČTK Database

This corpus is composed of the images of individuals in uncontrolled environment that were randomly selected from the large ČTK Photobank. All images were taken during a long time period (20 years or more). The detection and extraction of faces was realized automatically using our algorithm described in Section 4.1 and published in [101]. They were automatically resized to the size of 128×128 pixels and transformed to grayscale. The resulting corpus contains images of 638 individuals. 10 images for each person are available. Orientation, lighting conditions and background of images differ significantly. A correct face recognition on this dataset is thus very difficult.

Figure 3.11 shows examples of one individual from this corpus. This corpus is available for free for the research purposes at `<http://home.zcu.cz/~pkral/sw/>` or upon request to the authors.

3.12 Conclusions

This chapter gives an overview on existing datasets used in the face recognition domain. We mentioned the most frequently used corpora. The character of the datasets varies. Some of them are created in a controlled controlled environment whereas the others are extracted from real-world photographs. It thus allows to chose a suitable database for testing of new methods. We also introduce a newly created ČTK database. Note that for evaluation of our methods, we used four last datasets (i.e. ORL, Stirling, FERET and ČTK).

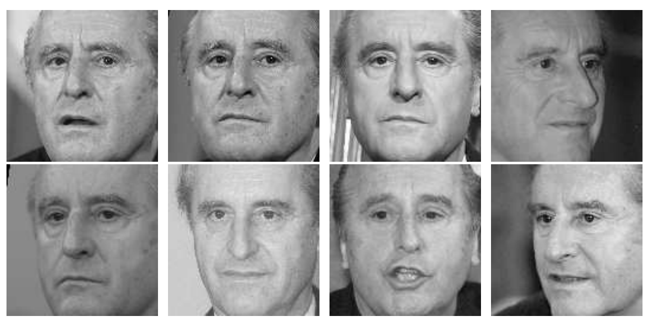

Figure 3.11: Examples of one face from the ČTK face corpus

Chapter 4

Corpus Creation

The purpose of the proposed face recognition system is labelling of photographs containing human faces from the large ČTK photographic database by person identities. There is a certain amount of photographs that are already manually labelled. The remaining photographs as well as newly acquired ones need to be annotated.

The images from the ČTK database have some properties that make the face recognition complicated. The faces have different poses, often the people are wearing helmets (e.g. hockey players) or other accessories. Also the ageing of subjects is problematic. Moreover, the time span between photographs is very high in comparison with standard face recognition datasets. Another important issue is that the photographs may contain more than one face. However, the photographs are annotated with exactly one label and another task thus consists in deciding whether the detected face is the labelled one.

To allow automatic face recognition we need a quality face corpus. Therefore, the first stage of the system is creating a face corpus from the labelled images. It consists in localizing the face within the image, extracting the face region and correction of face rotation. It is also necessary to group the images of one person (if multiple images per person are available). There is also a possibility of erroneous labelling. We thus need to check whether the image belongs to the correct person. A manual preparation of the corpus is the most precise way. However, the manual corpus creation is not practical because this task is very expensive and time consuming. Therefore, in order to minimize the human costs, an automatic corpus creation algorithm is proposed.

To the best of our knowledge there is only little work on the automatic corpus creation in the face recognition field. Automatic corpus creation methods have been developed and evaluated particularly in the speech processing domain [102, 103]. All the well known face databases have been created manually.

4.1 Proposed Algorithm

The proposed corpus creation algorithm is composed of the following tasks:

1. Face detection,

Figure 4.1: Example of one correctly detected face (left) and one incorrectly detected face by the Viola-Jones face detector

Figure 4.2: Examples of incorrectly identified faces by the Viola-Jones face detector implemented in OpenCV.

2. Identification and deletion of incorrectly detected faces,
3. Eyes detection,
4. Face rotation,
5. Image resizing and conversion to gray-scale,
6. Corpus cleaning.

4.1.1 Face Detection

We use the OpenCV library <http://opencv.willowgarage.com/wiki/> for the face detection task. It implements the Viola-Jones algorithm [6] which is based on the Haar cascade classifier and is one of the most successful face detection algorithms. Figure 4.1 shows one example of correctly and one example of incorrectly detected face.

4.1.2 Identification and Deletion of Incorrectly Detected Faces

Despite the high reliability of Viola-Jones face detection algorithm, there is certain amount of incorrectly detected faces (see Figure 4.2). Therefore, a verification of the detected faces is beneficial.

In order to avoid the manual processing, we propose a classifier for this task. It is used to classify the pictures into two classes: F (faces) and NF (non faces).

Figure 4.3: Examples of the faces with 2, 1 and 0 detected eyes (from left to right)

We chose a neural network of the type Multi-Layer Perceptron (MLP) due to its simplicity and good classification results [104].

Before the MLP classification, the images must first be parametrized. We proposed two types of image parametrization. The first one assumes that the distribution of pixel intensities of the two classes differs significantly. Therefore, we compute a histogram for each picture and we use histogram values as a feature vector.

The second method uses a sliding window. The input image is scanned by a non-overlapping window of the size 4×4 pixels. An average intensity value in the window is computed. The feature vector is then composed from average values of all window positions across the image. The vector length is thus 1024 for the 128×128 images.

In both cases, the MLP is trained on manually selected subset of faces (50 examples) and non-faces (50 examples). The MLP topology has 3 layers: the first layer has 256 input nodes (each input corresponds to one intensity value in the grayscale picture) in the first case and 1,024 in the second one. The hidden layer is composed of 10 neurons and two output nodes are used in both cases: $F \times NF$ classes.

4.1.3 Eye Detection

Many studies in the face recognition field confirm that the face quality influences significantly the face recognition accuracy [14]. The main problem are variations across pose. In our case, the database is composed of real-world photographs and therefore the poses differ significantly.

Because the recognition across poses is very difficult (and often uses separate classifiers for each pose) our aim is to obtain images as close to the frontal face as possible. A simple clue is proposed for this task. We detect the eyes within the face region. Presence of both eyes ensures that the face is more or less frontal.

The eyes are detected using the Viola-Jones algorithm (as for the face detection task). If both eyes are detected successfully the face is considered to be suitable for gallery creation. Note that the presence of both eyes does not ensure that the face is strictly frontal. Some amount of variances across the pose is still present. Figure 4.3 shows three results of the eye detection task where two, one and no eye are detected by the algorithm.

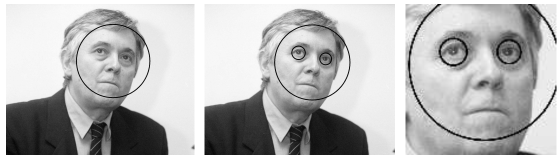

Figure 4.4: Example of face detection (left), eye detection (middle) and face rotation according to the eyes (right)

4.1.4 Face Rotation

After the previous step, only faces with reasonable pose variations are left for further processing. However, there still remain variations in face position inside the image and face rotation. Therefore, if both eyes are detected successfully, the face is rotated so that the eyes are on a horizontal line according to the detected positions. The face is further placed in the image centre. This transformation cannot completely resolve the issue of variations in pose and tilt. Nevertheless, face variations are significantly reduced. The resulting image is resized to the size of 128×128 pixels.

Figure 4.4 shows the tasks of face and eye detection and face rotation according to the eyes.

4.1.5 Corpus Cleaning

After the above described steps, we obtained a large number of face images for each individual. However, these numbers differ significantly. In our previous work [101], we proved that more training examples play a crucial role in the correct creation of face models. Nevertheless, we also showed that this number should be balanced (number of examples for each individual is comparable) and the quality of faces should be as high as possible. Further, there may be present some mislabelled images resulting from the problems described at the beginning of this chapter.

Therefore, we perform a step called "*corpus cleaning*" in order to choose a specified number of the most representative face images. We manually verified on a randomly chosen small face subset that the majority of face images is correct and the erroneous examples differ substantially from this representative set. Our algorithm used to choose only the representative faces is based on this observation.

Let S be the set of all face images extracted by the three steps described above. Let S_i be the set of n face images $I_1, .., I_n \in S_i$ representing one individual. Each face image I_j is represented by the feature vector F_j computed by the SIFT algorithm. The pseudo-code of the proposed algorithm is given below.

N = face image number/individual
K = required face image number/individual
for all $S_i \in S$ **do**
  while $N > K$ **do**
    for all $F_j \in F$ **do**

compute the face model M using the feature set $F \setminus F_j$
compute the similarity FS_j between F_j and M by the Equation 5.6
end for
compute an average value av_{FS}
compute a standard deviation sd_{FS}
compute a similarity threshold $ST = av_{FS} + \frac{sd_{FS}}{\sigma}$
if $\exists I_j : FS_j < ST$ **then**
for all $FS_k < ST$ **do**
remove the face image I_k from the face corpus
$N \leftarrow N - 1$
end for
else
remove the face image I_{min} with minimal similarity FS_{min} from the face corpus
$N \leftarrow N - 1$
end if
end while
end for

The similarity FS_j of an image I_j to a face model M created from the remaining images $I \backslash I_j$ is computed. The average av_{FS} of the similarity values and consequently the standard deviation sd_{FS} is computed. The similarity threshold ST is computed using these two values. Note that the optimal value of the constant σ is not possible to compute analytically and it is thus found experimentally on a development corpus. Images I_j with similarity FS_j lower than the threshold ST are discarded. The above described task is realized iteratively. If there is no similarity value lower than the threshold ST and there are still more images than the required number K, the images with the minimal similarities FS_{min} are further discarded. Finally the number of images/person is equal to or less than K, which is the required face image number defined at the beginning of the algorithm.

4.2 Experiments

This section summarizes the experiments performed to automatically create a quality face corpus from common photographs in the ČTK photographic database. It is assumed that multiple images for each person are available. The goal is to create face corpus suitable for creating accurate and robust face models which are then used for face recognition. The methods described in the previous section are used.

First, we created a face corpus which will serve for further experiments. It is created from a subset of images from the ČTK database containing 15,821 labelled images.

Figure 4.5 shows the structure of the automatically created face corpus by the above described algorithm. We can summarize the important information as follows:

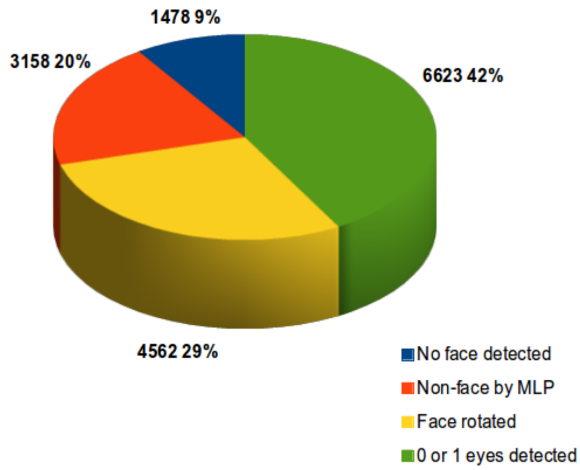

Figure 4.5: Structure of the created ČTK face corpus

- In 1,478 pictures no face was detected by the Viola Jones method,
- Another 3,158 images was marked as no faces by the MLP face $\times$ non-face classifier,
- Another 6,623 face-pictures was not rotated according to the eyes because no eye or one eye detected,
- In 4,562 faces two eyes were detected and these faces were rotated according to the eyes.

4.2.1 Face Detection, Identification and Deletion of Incorrectly Detected Faces

The purpose of the following experiment is to confirm the usefulness of the proposed methods for identification of incorrectly detected faces. We compare recognition results obtained on several subsets of the created dataset. The SIFT based adapted Kepenekci method described in Section 5.3 is used for face recognition. The subsets used for testing are described in Table 4.1. The first part of this table shows the number of individuals when the face detection step is not verified, while the second part reports the number of individuals when the verification of the face detection was used. The second column represents the number of individuals with a successful eyes detection, while the number of individuals with incorrectly detected eyes is reported in the third column.

Table 4.2 shows the face recognition accuracy on these different subsets of the ČTK dataset. We can make some interesting conclusions based on this table:

- Eye detection is very important for the face recognition (poor face recognition accuracies on the subsets with incorrectly detected eyes),

Table 4.1: Face dataset sizes in relation to: a) the number of the face examples per person; b) verification of the face detection; c) result of the eye detection

Example no. per person	Correctly detected eyes	Incorrectly detected eyes
1. Face detection without any verification		
	Number of individuals	
10	121	101
8	229	238
6	395	458
4	594	781
2. Face detection with the MLP verification		
	Number of individuals	
10	34	25
8	91	106
6	244	262
4	468	568

- The face detection using the Viola Jones method is good enough and further verification by the MLP classifier is not necessary (better recognition accuracy of the MLP verified set could be caused by the smaller number of recognized examples, only 34 individuals in the smallest set),
- Face recognition accuracy is significantly lower when the number of the recognized individuals increases and the number of training examples decreases,
- The best recognition accuracy (about 71%) is obtained on the subset with the smallest number of recognized individuals and with the highest number of training examples.

Then, we have analysed the output of the corpus creation algorithm. Based on this analysis, several conclusions were made:

- MLP removed quite a high number of false negatives (faces classified as non-faces) from further processing,
- Some faces were incorrectly rotated due to imprecise eye detection,
- Corpus still contains incorrectly detected faces.

The conclusions confirm the previously estimated facts. Therefore, we decided to exclude the MLP verification from the corpus creation process. The verification

Table 4.2: Face recognition accuracy on the different subsets in [%]

Example no. per person	Correctly detected eyes	Incorrectly detected eyes
1. Face detection without any verification		
	Face recognition accuracy in [%]	
10	56.94	19.01
8	44.60	14.45
6	34.89	12.23
4	24.28	7.74
2. Face detection with the MLP verification		
	Face recognition accuracy in [%]	
10	71.18	47.20
8	52.75	22.76
6	35.79	16.17
4	24.84	9.99

Table 4.3: Comparison of the recognition results on the raw and the cleaned version of the ČTK corpus

	Raw corpus	Clean corpus
Recognition rate [%]	30.72	64.73

step can be substituted by the eye detection and later also by the corpus cleaning algorithm. Without the MLP verification more faces are left for further processing. It can be beneficial for creating more robust face models and therefore also for recognition accuracy.

According to the described algorithm, we created a corpus from all available images. For the final version, we considered only individuals with at least 10 example images. As a result, we obtained a corpus containing images of 638 people. The total number of images is 11,095.

4.2.2 Corpus Cleaning

The last step in our corpus creation algorithm is the Corpus cleaning described in Section 4.1.5.

Table 4.3 shows differences in recognition rates on the original and the cleaned version of the ČTK corpus. Similarly as in the previous experiments the SIFT based adapted Kepenekci method is used for evaluation. This table clearly shows that the corpus cleaning algorithm is very important. The recognition rate on the cleaned corpus is nearly two times higher than on the original one.

Figure 4.6: Example images from the free version of the ČTK face corpus

The final cleaned version contains images of 638 individuals, at most 10 images for each. Total number of images is 5,698.

To further confirm the abilities of the corpus cleaning algorithm, we made an experiment with automatically selected test configuration. We selected randomly one test example for each person. Then we manually checked how many such images are incorrect (contain different person). The number of the erroneous samples in the raw corpus is significantly higher than in the cleaned version. Moreover, the cleaned version of the corpus contains only very small number of the mislabelled examples. This manual check also confirmed our assumption that the corpus cleaning step is beneficial in the whole algorithm.

Unfortunately, due to license conditions this version of the corpus cannot be used publicly. Therefore, we created a slightly restricted version of the ČTK corpus. It contains images of 561 individuals. The total number of images is 10,196. The face detection is not performed on this version of the corpus. Moreover, the face sizes differ in substantial extent. It is thus possible to use and test other face detection and face alignment algorithms on this corpus. This version of the corpus is publicly available for free upon request to the authors. Figure 4.6 shows some examples of this face corpus.

4.3 Conclusions

In this chapter, we proposed an algorithm that allows automatic creation of a face corpus. An experimental corpus was created using the algorithm. We performed a series of the face recognition experiments on this corpus using the SIFT based adapted Kepenekci method. The experiments showed that the verification of face detection using an MLP is not beneficial. The other steps are necessary for creation of a quality face corpus. As a result, we created a face corpus containing images of 638 people and with up to 10 images for each.

A restricted version of the corpus containing 561 is freely available for research purposes. In this case no face detection is applied to allow the researchers to use their own face detection algorithms. Moreover, the corpus cleaning algorithm is not applied and there are at least ten images for each person.

Chapter 5

Face Recognition

The second and fundamental step of the whole proposed system is the face recognition itself. During the evolution process of the system, several face recognition methods have been tested. This chapter first describes all proposed methods and some enhancements of the existing approaches. Section 5.5 presents the experiments we performed on these methods.

The first proposed method uses the Gaussian Mixture Models (GMMs) for classification. Then the Adapted Kepenekci method based on Gabor wavelets is described. After reaching promising results with this method, we propose to use similar matching scheme together with the Scale Invariant Feature Transform (SIFT) features which is a promising direction in the face recognition field. Finally, we propose to use a combined SIFT/SURF descriptor for face recognition.

5.1 Gaussian Mixture Models

Gaussian mixture models are often successfully used for classification in other fields (handwriting recognition, speech recognition, etc.). Therefore, we have decided to use this method also for the face recognition. Two methods based on the GMMs will be presented next.

5.1.1 SOM with a Gaussian Mixture Model

Current face recognition methods are usually composed of two steps: parametrization and classification. Parametrization is used to create a face representation. The goal is to reduce the size of the original image with minimal loss of discriminating information. The parametrized face model is then used in the classification step instead of the original image.

This method combines a Self Organizing Map (SOM) for creating a face representation and Gaussian mixture models for classification. We used the self organizing map in order to reduce the size of the feature vectors. Utilizing the SOM in the parametrization step is motivated by the method proposed in [2]. Authors also used

SOMs in the first step, while a convolutional neural network is used for classification. In our case, the second step is classification by the GMMs.

Parametrization with a SOM

Input images are represented as two-dimensional arrays of pixel intensities. We consider grayscale pictures where each pixel is represented by a single intensity value. Each image can be also seen as a one-dimensional vector of size $w * h$, where w and h are image width and height, respectively. A self organizing map is used for the dimensionality reduction . The image is first sampled. It means a set of vectors is created in the way which is described next.

The sampling procedure uses a rectangular non-overlapping sliding window which scans over the image. At each position, a vector containing intensity values of pixels, is created. The size of the created vector is $l = ww * wh$, where ww and wh are window width and height, respectively. The vectors obtained from all images are used as a training set for the self organizing map. The trained SOM (standard SOM training algorithm) is then used for image parametrization. Each input vector is associated with the closest node of the SOM and its position is used to compute the resulting parameter vector. Values of this vector are created as an average value of the node vector associated with this position. The vectors are used as an input for the classification step.

Classification with GMMs

Let us call F the set of features for one image obtained in the parametrization step, let I be the face image. We use a GMMs classifier that computes $P(F|I)$. The recognized image is then:

$$\hat{I} = \arg\max_{C} P(I|F) = \arg\max_{C} P(F|I)P(I) \tag{5.1}$$

We assume all images to be equiprobable. The prior probability $P(I)$ can thus be removed from the equation.

5.1.2 Re-sampling with Gaussian Mixture Models

An alternative way to reduce the feature space in the parametrization step is image re-sampling. Image size is reduced using the resize filters from ImageMagick library[2]. Intensity values of the resulting images are directly used as image vectors. These vectors are classified by GMMs in the same way as in the previous case.

Different resize filters and different sizes of resulting output vectors are evaluated. The first four filters are interpolated filters, while the last one, *Cubic filter*, is a Gaussian filter. The first method, the *Point filter*, determines the closest point in the original image to the new pixel position and uses its intensity value in the resized image. The *Box filter* computes an average value of pixels placed in the "box" (a rectangular window of a defined size). The next evaluated filter is a *Triangle* *filter*. This filter takes into account the distances of the pixels and uses a weighted average instead of just an average value. *Hermite filter* has similar results as the triangle filter, but produces a smoother round off in large scale enlargements. More information about resize filters is available at ImageMagick website[2].

The classification step is the same as in the previous case.

5.2 Adapted Kepenekci Method

Two previously proposed methods achieved very good recognition scores on small standard corpora (i.e. ORL dataset). However, their recognition accuracy was significantly reduced when large or real-world corpora (i.e. FERET, ČTK) are used. It is caused by the nature of these methods that belong to the group of holistic methods. More suitable for the real-world data are the feature based methods. Especially methods based on Gabor wavelets achieve impressive results. Therefore, we focused on such methods.

We decided to adapt the Kepenekci method [62] which is also based on Gabor wavelets. Several modifications due to the character of our data were needed. Three main adaptations of the method are as follows:

- Increased window size - larger sliding window for fiducial points determination,
- Composed face model - using more images to create the model,
- Removing the similarity threshold ($similarityThreshold$).

5.2.1 Face Representation

Same as in the case of the original Kepenekci method, Gabor wavelets are used for image representation. The fiducial points are determined automatically based on the Gabor filter responses. A set of 40 filters with varying wavelengths and orientations is used.

The image is convolved with all Gabor filters from the filter bank. Wavelet responses R_j, where $j = 1, ..., 40$, are obtained. Each of these responses is scanned with a sliding window. Assume a square window W of size $w \times w$. All possible window positions within the response are evaluated. The center of the window, denoted (x_0, y_0), is considered to be a fiducial point iff:

$$R_j(x_0, y_0) = \max_{(x,y) \in W} R_j(x, y) \tag{5.2}$$

$$R_j(x_0, y_0) > \frac{1}{wi * hi} \sum_{x=1}^{wi} \sum_{y=1}^{hi} R_j(x, y) \tag{5.3}$$

where $j = 1, ..., 40$, wi and hi are image width and height respectively.
The feature vector in point (x, y) is created as follows:

$$v(x,y) = \{x, y, R_1(x,y), ..., R_{40}(x,y)\} \tag{5.4}$$

The resulting vector thus contains information about feature point coordinates and values of Gabor responses in this point.

5.2.2 Composed Face Model

In our application we assume that more than one image for each person is available. Therefore, we propose so called *Composed face model* for face representation. We find fiducial points and corresponding vectors for each image as described in Section 5.2.1. Then we put all these vectors together to create a final representation.

The face model based on more images ensures better robustness of the algorithm. The variability of images used for representation is especially important in case of real-world images.

5.2.3 Face Comparison

The cosine similarity [64] is employed for vector comparison. The similarity between two vectors thus takes the values from interval $[0, 1]$. Only the last 40 positions (the first two are point coordinates) in the vector are considered.

Let us call T a test image and G a gallery image. For each feature vector t of the face T we determine a set of relevant vectors g of the face G. Vector g is relevant iff:

$$\sqrt{(x_t - x_g)^2 + (y_t - y_g)^2} < distanceThreshold \tag{5.5}$$

If no relevant vector to vector t is found, vector t is excluded from the comparison procedure. However, the most similar vector (from all relevant vectors) is used for the face similarity computation. Contrary to the original Kepenekci method we do not use the similarity threshold ($similarityThreshold$). This modification slightly reduces the computation time with no negative impact on the recognition accuracy. The overall similarity of two faces OS is computed as an average of similarities between each pair of corresponding vectors as:

$$OS_{T,G} = mean\,\{S(t,g), t \in T, g \in G\} \tag{5.6}$$

Then, the face with the most similar vector to each of the test face vectors is determined. The variable C_i says how many times the gallery face G_i was the closest to some of the vectors of the test face T. The similarity is computed as C_i/N_i where N_i is the total number of feature vectors in G_i. Weighted sum of these two similarities is used for similarity measure:

$$FS_{T,G} = \alpha OS_{T,G} + \beta \frac{C_G}{N_G} \tag{5.7}$$

The size of the sliding window is very important for the performance of this method. It determines the number of detected fiducial points and influences its accuracy. The higher the window size is the less fiducial points are detected. On the other hand, searching larger window needs more computation time. In the comparison stage, the number of fiducial points determines the time needed. The above mentioned $distanceThresholds$ also influence the accuracy and the run-time of this method. The smaller the value of this threshold is, the less comparisons are needed and the method works faster.

5.3 SIFT Based Methods

This section describes three methods based on the SIFT algorithm. The first face recognition method based on SIFT was proposed by Aly in [66] and achieved promising results on the Yale and ORL databases [66]. The SIFT features are invariant to rotation, scale and lighting conditions which is beneficial when recognizing real-world images. Therefore, we propose to use these features together with more sophisticated matching scheme.

For comparison, we describe the matching scheme proposed by Aly in [66] Further, we propose to use two matching schemes which were previously not used in combination with the SIFT features. The first one is a novel Lenc-Kral matching. The other one is the Adapted Kepenekci matching previously used with Gabor wavelets based features (described in Section 5.2).

The SIFT algorithm is described in 2.1.20. For further details please refer to [16, 5, 65]. An implementation example can be found in [105] and currently it is also implemented in the OpenCV[1] library.

5.3.1 SIFT Features Extraction

The feature points in the face are detected automatically by a part of the original SIFT algorithm. Further, the descriptor in each of the detected points is constructed.

Figure 5.1 shows the SIFT features detected in the example images from the ČTK face corpus.

5.3.2 Aly Matching

This approach computes the number of the gallery image feature vectors that are matched against the test face feature vectors. For each test feature vector the similarities to all of the gallery feature vectors are computed. The cosine similarity of two feature vectors f_1 and f_2 is computed as follows:

$$S(f_1, f_2) = \frac{f_1 \cdot f_2}{\|f_1\|\|f_2\|} \tag{5.8}$$

[1]http://opencv.org/

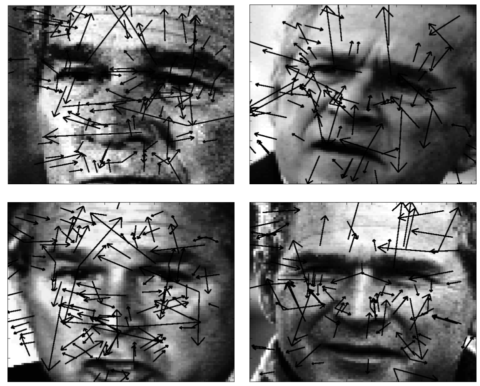

Figure 5.1: Examples of detected SIFT features with orientation.

The two most similar gallery feature vectors are determined. If the difference between these two similarities is higher than a pre-specified threshold the feature vector is considered to be matched. For each gallery face, the number of matched feature vectors is computed. The recognized face is the one with highest number of matched feature vectors.

5.3.3 Lenc-Kral Matching

The first proposed approach computes a sum of similarities between pairs of image feature vectors. For each feature vector of the test face the most similar feature vector of the gallery face is identified. The sum of the highest similarities is computed and is used as a measure of similarity between two faces.

Speaking in more mathematical terms, let T be a test image represented by m feature vectors $t_1, t_2, .., t_m$. Let G be a gallery of images composed of N images $G_1, G_2, .., G_N$. Let every gallery image G_i be represented by n_i feature vectors $g_1, g_2, .., g_{n_i}$. Similarity of two feature vectors $S(t, g)$ is computed by the cosine similarity (see Equation 5.8). For each feature vector t_i of the recognized face T we determine the most similar vector $g^j_{max_i}$ of one gallery image G_j:

$$g^j_{max_i} = \arg\max_{G_j}(S(t, g)) \tag{5.9}$$

The sum of those similarities is computed as follows:

$$D(T, G_j) = \sum_{i=1..m} g_{max_i} \tag{5.10}$$

where m is the number of test image feature vectors. The recognized face is then determined by the following equation:

$$\hat{G}_i = \arg\max_{G_i}(D(T, G_j)) \tag{5.11}$$

5.3.4 Adapted Kepenekci Matching

This approach has been initially used by Kepenekci in [62] with Gabor wavelets. Author shows that this approach exhibits high recognition accuracy. We use the adapted version described in Section 5.2. We decided to adapt this successful approach and integrate it with the SIFT.

Kepenekci combines two methods of matching and uses a weighted sum of the two values as a result. The cosine similarity is employed for vector comparison. The algorithm is thoroughly described in Section 5.2.3.

5.4 Combined SIFT/SURF Descriptor

Similarly as the SIFT also the SURF descriptors were successfully used for face recognition [69]. The performance of these methods used separately is comparable. The idea of the proposed combined SIFT/SURF descriptor is to take advantage of both methods. Combining the descriptors should bring more information to the face representation and also increase robustness.

To obtain combined SIFT/SURF representation we first detect the key-points using both methods. This results in a set of image points where the descriptors are created. The SIFT descriptor has 128 dimensions whereas the SURF descriptor has 64 dimensions. Concatenating of the two descriptors thus results in a descriptor of length 192. The descriptor values are first normalized before the concatenation.

The representation of a face is created as follows:

1. SIFT key-points detection,
2. SURF key-points detection,
3. Calculation of the SIFT and SURF descriptors in all detected key-points,
4. Normalization of the SIFT and SURF descriptors,
5. Concatenation of these two vectors in order to create a combined descriptor.

We utilized the Adapted Kepenekci matching scheme for the comparison of face representations.

5.5 Experiments

The experiments in this section are motivated by the effort to find the most suitable method for face recognition under real-world conditions. This method will be integrated into the proposed face recognition system. All above described methods were evaluated and compared. Most of these experiments are performed on three datasets: ORL, FERET and ČTK database.

5.5.1 Results of the Gaussian Mixture Models

In this section, experiments with Gaussian Mixture Models (GMMs) classifier are described. The GMM classifier is used together with two types of parametrization. The experiments are carried out on the ORL dataset. We chose the "classic" Eigenfaces method as a baseline. The recognition error rate obtained when using Eigenfaces method is 30.85% (see Table 5.1).

A cross-validation procedure is used in this case. 10% of the corpus is reserved for the test. However, for the training, we use the remaining 90% of examples from the training pool. All training examples are subsequently used. The first approach to parametrization of input images is using a self organizing map.

Parametrization by SOM

Table 5.1 shows the recognition error rates achieved with four SOM sizes. During the experiments, many SOM topologies were evaluated. However, only the four best ones are reported in this table. All recognition scores are very high and outperform the baseline significantly. Using the SOM size of 12×12 pixels, the recognition error rate is reduced to 3.25%. This table also shows that the size of the SOM has high impact on the face recognition accuracy.

Table 5.1: Automatic face recognition error rate for different parametrization/classifications methods.

Method	Error rate
1. **Eigenfaces**	30.85
2. **SOM & GMM**	
SOM size	Error rate
8x8	4.5
10x10	3.75
12x12	3.25
14x14	3.75

Two Level Dimensionality Reduction

In this experiment, we would like to evaluate the relation between the reduction of the parametrized input vector and the loss of the recognition accuracy. This means to determine the minimal feature vector length without significant decrease of performance of our system.

Another SOM is used for this additional vector reduction in a similar way as in the previous case. Several SOM topologies are evaluated. Table 5.2 shows the recognition rates achieved in this experiment. The results are not as good as in the previous case, but still significantly better than our baseline. The best recognition rate is obtained with the SOM topology 10×10 neurons with resulting vector size 42.

Table 5.2: Error rate for two level dimensionality reduction with different SOM sizes

1st level 1 SOM size	2nd level SOM size	Error rate
10x10	8x8	14.25
12x12	10x10	10.25

Parametrization by Re-sampling

In this experiment, resizing of images is used for parametrization. Table 5.3 shows recognition error rates for different filters and different image sizes. This table shows that all filters are almost comparable except the *Point filter*. The worst recognition score of this filter is probably due to its simplicity. Moreover, the lowest recognition error rate is obtained using the image size of 7×8 pixels. We can conclude that the recognition accuracy of this experiment is close to the previous one with one level SOM parametrization.

Table 5.3: Comparison of recognition error rate of different resize filters and different parametrized vector size with a GMM classifier

Filter	Point	Box	Triangle	Hermite	Cubic
2x3	55	40	39.75	38	40.75
3x4	44.75	13.5	16	14	17.75
5x6	20.75	6.25	5	5.25	6.5
6x7	18.5	3.25	3	3	4
7x8	12.5	3.25	3.25	4.5	4
8x10	7.75	2.25	3	2.75	2.5
9x11	5	2.5	2.75	2.75	2.75
10x12	5.25	3	2.75	3.5	3
11x13	5.25	3	3	3.25	3.5
13x16	3.75	3.25	3.5	4	3.75
15x18	3.75	3.75	3.5	4	3.5
17x21	4	4	3.75	4.5	3.75
19x23	4	4.5	4.75	4.25	4.5
21x26	4	3.75	3.75	4	3.75
23x28	4.25	4.5	4.25	4.75	3.75

5.5.2 Results of the Adapted Kepenekci Method

Another series of experiments was performed using the adapted Kepenekci method. This method was chosen because of its high recognition accuracy and the assumption that it might perform well in case of real-world data. We decided to do some modifications of this method in order to decrease computation costs and adapt it

[2] http://www.imagemagick.org/Usage/resize/

for the real-world data. Also the adaptation for using more examples to create the face model is used. The adaptations are described in Section 5.2. First, we determine the size of the sliding window used for fiducial points detection. The experiments are performed on the ORL and Stirling databases.

Sliding Window Size

The size of the sliding window is one of the most important parameters in the algorithm. Kepenekci [62] used the window size of 9 pixels without any justification. However, this parameter significantly influences the number of detected fiducial points and thus plays a crucial role in recognition accuracy and speed. When the size of sliding window is set to 9 pixels and the image width is 92 pixels (images in ORL database), the algorithm finds around 1500 fiducial points. Other methods, such as EBGM, show that less than 100 fiducial points is enough for a successful face recognition. The more fiducial points is used, the longer computation time for recognition is needed. We assume that the size of sliding window should depend on the image size (represented by width $\times$ height). Figure 5.2 shows the numbers of fiducial points related to different sizes of the sliding window.

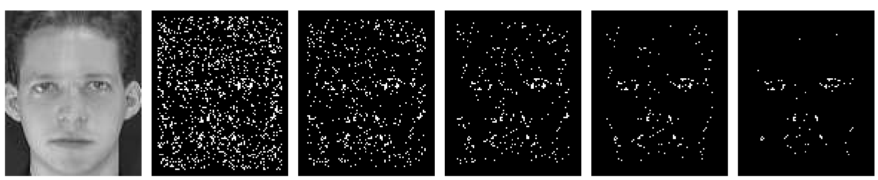

Figure 5.2: Original image and mapping of fiducial points related to the different sliding window sizes: 9 (left), 13, 17, 21 and 27 (right) pixels from ORL database.

Furthermore, the choice of fiducial points is significantly influenced by the manipulation with the sliding window close to the image borders. Kepenekci does not specify how the positions near image margins are handled.

The main goal of this experiment is thus three-fold:

1. to determine experimentally an optimal value of the sliding window size (a compromise between the recognition accuracy and computation time); and
2. to propose an optimal solution of handling the sliding window close to the image borders. We propose and evaluate two approaches:
   (a) take into account only positions where the window fits in the image. It produces fiducial points which are placed at more than $windowSize/2$ pixels from the border. This approach is hereafter called *Restricted Points*.
   (b) consider all possible positions including cases when the window partially exceeds the image border, hereafter called *All Points*.

Table 5.4: Recognition accuracy for *Restricted Points* and *All Points* approaches for different sizes of sliding window W on the ORL dataset: ACC is the recognition accuracy and T is the computational time in %. The best values reported in boldface are chosen in order to have still a good accuracy, whereas the recognition time is as low as possible.

ORL database				
W	**Restricted points**		**All points**	
	ACC in %	**T** in %	**ACC** in %	**T** in %
9	88.6	100	89.4	100
13	87.5	27.8	88.9	35.6
17	**86.1**	**10.3**	89.4	16.5
21	83.9	4.3	**88.6**	**9**
27	80.6	1.6	88.3	4.6
33	74.2	0.75	86.9	2.7
41	67.2	0.3	87.8	1.6
45	65	0.2	86.4	1.3

Table 5.5: Recognition accuracy for *Restricted Points* and *All Points* approaches for different sizes of sliding window W on the Stirling dataset: ACC is the recognition accuracy and T is the computational time in %. The best values reported in boldface are chosen in order to have still a good accuracy, whereas the recognition time is as low as possible.

Stirling database				
W	**Restricted points**		**All points**	
	ACC in %	**T** in %	**ACC** in %	**T** in %
29	100	100	100	100
35	98.4	51.3	98.4	53.8
43	98.4	24	98.4	27.6
53	98.4	11.5	98.4	13.8
65	**98.4**	**6.1**	**98.4**	**7.2**
79	98.4	3.1	96.9	3.8
95	96.9	1.5	96.9	2.2
113	92.2	0.6	98.4	1.2

Table 5.4 evaluates two above proposed approaches for different sizes of sliding window on the ORL dataset while Table 5.5 does the same for the Stirling dataset.

In the target application both recognition accuracy and computational time are important. Therefore, we have to take into consideration both these quantities when determining the optimal value of the sliding window size. Our goal is to have a good accuracy, whereas the recognition time is as low as possible. We started with the lowest value of the sliding window size and determined the "gain" when the size is increased. The "gain" denoted G means lower recognition time and higher accuracy. Therefore, a decreased recognition rate adds a negative number. It is computed according to the following equation:

$$G_i = -w_{ACC}(ACC_{i-1} - ACC_i) + w_T(T_{i-1} - T_i) \tag{5.12}$$

where w_{ACC} and w_T are weights adjusted to 0.9 and 0.1 respectively because the accuracy is significantly more important. i is the order of the evaluated sliding window size. As an optimal value we chose the first one for which the "gain" is lower than 1.

According to this criteria, the optimal values for the ORL database are 17 for the restricted points and 21 for all points. For the Stirling database, the optimal value is 65 in both cases.

The tables also show that the best performing approach to handling the window positions near the image border is the *All points* one.

Distance Threshold

Another parameter that plays important role in the accuracy of the Kepenekci method is the *distanceThreshold*. It determines the spatial distance within which we compare fiducial points of compared images. As well as the window size it should be proportional to the face size. We assume that the value of this threshold is proportional to the face size. We thus used only the Stirling dataset in this experiment. A suitable threshold for other datasets is then determined according to the size of the images. Table 5.6 shows recognition accuracy of this experiment.

Table 5.6: Recognition accuracy for different values of *distanceThreshold* on the Stirling database. The optimal value is highlighted. Window size 65 is chosen according to the Table 5.5.

	Window size				
distanceThreshold	**43**	**53**	**65**	**79**	**95**
6	90,63	85,9	78,1	75	70,3
10	98,4	98,4	95,3	96,9	96,9
14	98,4	98,4	98,4	96,9	96,9
18	98,4	98,4	98,4	98,4	98,4
22	100	98,4	98,4	98,4	98,4
26	100	100	98,4	98,4	98,4
30	100	100	98,4	100	98,4
34	100	100	**100**	100	98,4
38	100	100	100	100	98,4

Assuming that the window size is 65 points, we can suggest the *distanceThreshold* value of 34 points (about one half of the sliding window size).

Modification of the Face Representation

In most Gabor wavelet based approaches only one gallery image for each person representation is considered. We suggest employing more training images in order to increase the performance and robustness of the algorithm. This is motivated by the fact that face images in the ČTK database differ in significant extent. One particular image (chosen randomly) cannot thus represent well one individual and the additional images will certainly add more information into the face model.

Therefore, we proposed a so called "Composed face". Feature vectors are first extracted from all training images. The "Composed face" is then created as a set of all feature vectors of one individual. Test faces are matched against this composed face model.

As mentioned in the previous section, we use a larger sliding window than in the original method. We thus achieve a lower number of fiducial points per image. The composed gallery face representation from all training images has similar number of vectors as one face in the original method. The computational time will be thus close to the original algorithm, while the recognition rate should improve.

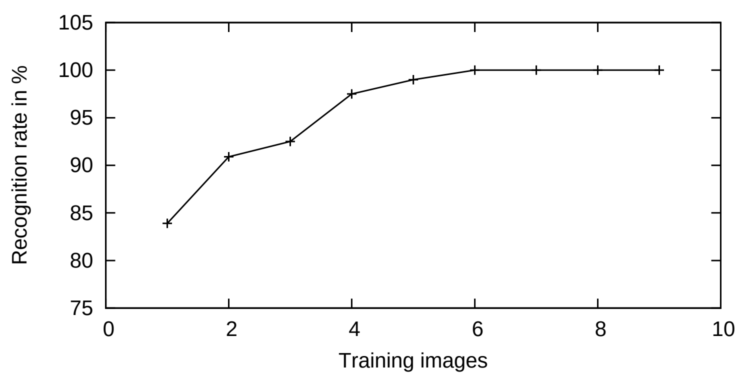

Figure 5.3: Face recognition rate for different number of training images on the ORL database: the X-axis represents the number of training images and the Y-axis plots the face recognition rate.

Figure 5.3 shows the face recognition rate for different numbers of training images on the ORL corpus. This experiment confirms our assumption, that using more training images significantly increases the recognition rate. Moreover, six training examples is enough for obtaining a 100% recognition rate that represents an increase of recognition rate by 17% in absolute value (over the approach with one training image).

To evaluate the capabilities of the "Composed face" representation we used also the ČTK corpus. In this case the task is more challenging because of the use of lower quality images. The main goal of this experiment is thus to evaluate the robustness of the modified Kepenekci algorithm on a real-world ČTK dataset. Figure 5.4 shows face recognition rates for different numbers of training images on the ČTK corpus. This experiment shows that using more training images is necessary to obtain an interesting recognition rate, and that using seven training examples represents an increase of recognition rate by about 35% in absolute value (over the original Kepenecki approach when one training example is used).

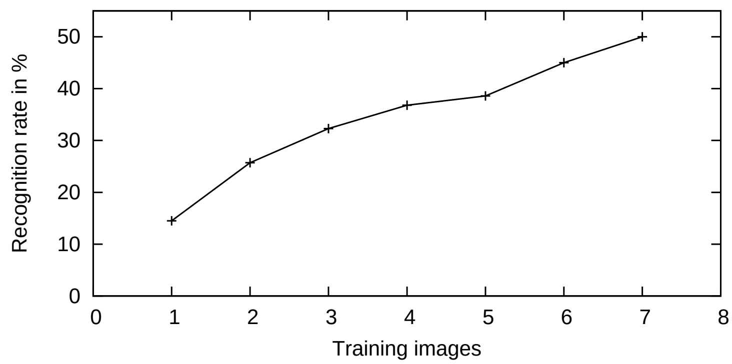

Figure 5.4: Face recognition rates in relation to different number of training images on the automatically created ČTK corpus: the X-axis represents the number of training images and the Y-axis plots the face recognition rate.

Modification of the Face Matching

In the original Kepenekci algorithm (see Section 2.1.19), a threshold called similarity threshold ($similarityThreshold$) is used for the face matching. We assume that this threshold does not play a role in the recognition accuracy and it is thus possible to simplify the algorithm by removing it. It can only slightly speed-up the recognition itself. Moreover, the computational time for determining the threshold according to the proposed algorithm is very high.

When this threshold is not used, the computational time is 2.71 s and the recognition rate is 100% (tested on the Intel Core i5-2300, 2.80GHz, 16GB RAM). This fact and results in Figure 5.5 confirm our assumption that the recognition rate is not influenced by this threshold and that speed up of the matching process is negligible when the threshold is used. Omitting the threshold thus results in lower total time because we do not need to determine the threshold value which is computationally expensive. Therefore, we state that it is not necessary to use the threshold. The reported recognition rate is obtained on the Stirling database.

5.5.3 Results of the SIFT Based Methods

This section presents experiments with proposed methods based on the SIFT algorithm, which is used to create the face representation. The face representation is the same for both proposed methods. We compare the results of the Lenc-Kral matching and the Adapted Kepenekci matching.

These experiments were motivated by the fact that the previously evaluated approaches do not have the sufficient recognition accuracy on a real-world database.

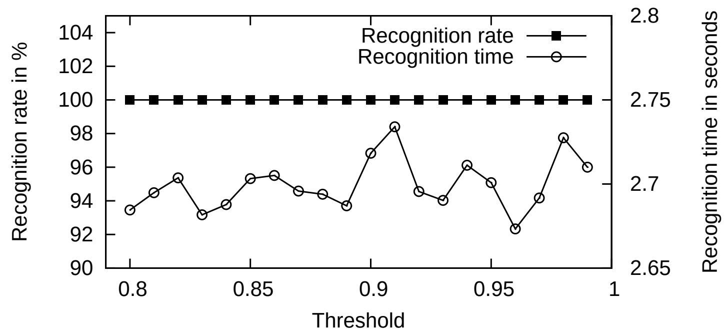

Figure 5.5: Face recognition rates and times needed for recognition of one face for different values of *similarityThreshold*.

All experiments were performed on three datasets: the ORL dataset, FERET and the ČTK corpus. We used the successful Aly method (see Section 5.3.2) as a baseline. We also compare the results with our previously proposed Adapted Kepenekci method based on Gabor wavelets.

Matching Schemes Comparison

The matching scheme proposed by Aly achieves promising results. Nevertheless we performed a series of experiments to compare the results with the newly proposed Lenc-Kral and Adapted Kepenekci matching schemes. We compared the results on the ORL dataset and on a subset of the ČTK dataset containing 63 individuals and 8 images for each person.

The size of the training set is gradually increased from 1 image/person to $N-1$ images/person (N is the total number of images/person). We used 9 different set-ups for the ORL dataset and 7 set-ups for the ČTK dataset. To allow a straightforward comparison of these methods, we evaluated each set-up with three previously described matching schemes.

Table 5.7 shows the recognition rates on different test set-ups for the ORL dataset. This table shows that the scores of the proposed Lenc-Kral approach are significantly higher than the original Aly method especially where not enough training examples available. The second proposed approach (SIFT based Kepenekci method) have slightly better recognition accuracy than both other approaches.

Table 5.8 shows the recognition accuracy of the experiments on a subset of the ČTK corpus. The recognition accuracy is significantly lower than in the case of the ORL database. This is caused mainly by the uncontrolled nature of photographs in the ČTK dataset.

This table also shows that both proposed methods significantly outperform the baseline Aly approach for all training examples in all cases. The Adapted Kepenekci matching scheme proved to be very efficient for face recognition. Therefore, we decided to use this scheme for further experiments.

Table 5.7: Recognition rate of the different matching schemes for the ORL dataset according to the different training set size

Matching scheme	**Aly**	**Lenc-Kral**	**Kepenekci**
Training Set	**Recognition rate (%)**		
1 of 10	61.25	78.75	80.56
2 of 10	78.72	88.24	90.15
3 of 10	85.36	92.46	94.24
4 of 10	88.83	95.67	97.25
5 of 10	92.42	96.75	97.92
6 of 10	95.27	97.86	97.86
7 of 10	96.88	98.65	98.65
8 of 10	98.36	98.86	99.17
9 of 10	99.00	99.00	99.25

Table 5.8: Recognition rate of the different matching schemes for the ČTK corpus according to the different training set size

Matching scheme	**Aly**	**Lenc-Kral**	**Kepenekci**
Training Set	**Recognition rate (%)**		
1 of 8	9.78	12.95	19.73
2 of 8	14.18	19.11	27.78
3 of 8	16.90	24.29	31.75
4 of 8	20.40	28.89	37.10
5 of 8	22.93	31.92	41.18
6 of 8	24.12	34.27	43.85
7 of 8	25.79	36.71	46.63

Comparison with Adapted Kepenkci Method

The following series of experiments was carried out to compare the abilities of the SIFT based adapted Kepenekci method (the best previously performing approach) with Adapted Kepenekci method (based on Gabor wavelets). First, we compare this two methods on the ORL dataset.

Table 5.9: Recognition results on the ORL database

Method	**Kepenekci**	**SIFT**
Training Ex.	**Recognition rate [%]**	
9 of 10	99.50	100
8 of 10	99.58	99.86
7 of 10	98.75	100
6 of 10	98.39	99.82
5 of 10	97.42	99.50
4 of 10	96.00	99.17
3 of 10	93.48	97.41
2 of 10	90.63	91.88
1 of 10	78.89	81.52

Table 5.9 shows the recognition rate of both methods on the ORL database when different number of training images (1-9) is used and the rest of images is used for testing. This table shows that the recognition scores of both approaches are comparable on the "small" and "artificial" corpus. Moreover, the obtained recognition rate is close to 100% when more examples is used. This experiment has been performed in order to show that both approaches perform well in the laboratory conditions on a small corpus.

For comparison, an experiment on a sub-set of the ČTK dataset was performed. The sub-set contains 37 individuals and 10 example images for each individual.

Table 5.10: Recognition results on the ČTK database

Method	**Kepenekci**	**SIFT**
Training Ex.	**Recognition rate [%]**	
9 of 10	60.81	72.70
8 of 10	57.66	69.07
7 of 10	53.83	65.20
6 of 10	50.10	62.36
5 of 10	47.12	57.21
4 of 10	42.88	51.17
3 of 10	37.55	44.21
2 of 10	32.09	34.57
1 of 10	24.62	22.37

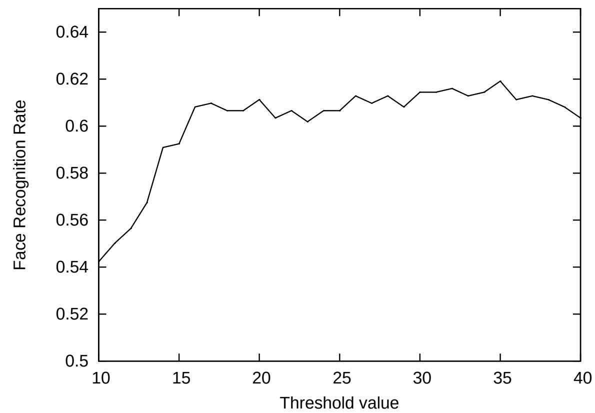

Figure 5.6: Optimal distance threshold value estimation for the ČTK corpus

Table 5.10 shows recognition results of both methods on this corpus. It is apparent that in the case of non-controlled images the difference between the performance of the two methods is much higher. The SIFT based adapted Kepenekci method outperforms significantly the Gabor based adapted Kepenekci approach. Therefore, we conclude that using the SIFT features is more suitable for real-world data than the Gabor wavelet based features.

5.5.4 Results of the SIFT based Adapted Kepenekci Method

In the previous section, we experimentally determined that the best performing method is the SIFT based adapted Kepenekci method. In following experiments we want to tune up the method for the use on real-world data from the ČTK corpus.

Parameter Optimization for the ČTK Corpus

The images in the ČTK corpus differ significantly from the images in the other databases. Therefore, an optimization of our method for this corpus is beneficial.

The most important parameter is the *distance threshold* (see Equation 2.9) which significantly influences recognition accuracy.

This threshold defines the surrounding region to a given key-point. Only the vectors in this region are used for comparison in the recognition step. The optimal value of this threshold is strictly related to the size and quality of the recognized images. Unfortunately, it is not possible to analytically set its optimal value. Therefore, this value will be set experimentally on a development corpus as demonstrated in Figure 5.6. The evaluated values are in the interval $]0; 50]$. However, for better clarity of this figure, the reported interval is reduced by 10 from both sides.

The highest obtained recognition rate is 61.91% when the threshold value is set at 35. Therefore, this value will be used in the following experiments.

Experiments with High Number of Individuals

The following experiments are undertaken in order to show the performance of this method on real data. We evaluate the method on different subsets of the ČTK database with varying number of individuals and number of images for each individual. We used a version of the ČTK database where the cleaning algorithm was not used.

Table 5.11: Recognition results of the SIFT based adapted Kepenekci approach on the ČTK database using different number of training examples and different amount of individuals

Training Ex.	**Database size (individuals #)**	**Recognition rate [%]**
9 of 10	37	72.70
8 of 9	88	56.70
7 of 8	194	49.94
6 of 7	367	41.92
5 of 6	595	33.51
4 of 5	841	27.90
3 of 4	1065	21.31

Table 5.11 details the relation among the recognition accuracy, the number of the individuals and the number of the training examples. This table shows that the recognition rate decreases significantly when the number of individuals increases and the number of training images decreases.

From the results of the previous experiment, it is obvious that the number of training examples is very important for the face recognition accuracy. Unfortunately, we do not know, whether the second parameter (number of individuals) influences the face recognition in the same manner. Therefore, we realized the following experiment that shows the face recognition accuracy for a constant number of training examples (equal to 3). Table 5.12 presents recognition results when different number of individuals is used. The number of images is set to 4 in all cases.

Table 5.12: Face recognition accuracy on different subsets of the ČTK database with 4 face examples per person in [%]

Number of Individuals	**Recognition Accuracy [%]**
34	32.35
91	27.75
244	23.77
468	24.84

Table 5.13: Recognition results on the cleaned version (with 638 individuals) of the ČTK corpus using three different evaluation metrics

Evaluation metric	Recognition rate [%]
1. SIFT based adapted Kepenekci method	
Correct	**64.73**
Correct in 5 most similar	74.76
Correct in 10 most similar	78.37
2. Adapted Kepenekci method	
Correct	48.75
Correct in 5 most similar	57.21
Correct in 10 most similar	60.50

5.5.5 Results on the Final Large ČTK Corpus

This experiment demonstrates the performance of the SIFT based adapted Kepenekci method on the large real-world ČTK corpus. The final cleaned version of the ČTK corpus described in Chapter 4 (with 638 individuals) is used. We used the Adapted Kepenekci method for comparison.

From the viewpoint of the application of our system, we would like to identify whether a correctly recognized face belongs to the N best recognized ones. If this is the case, we can propose to a user choosing the correctly recognized face, for example, from a drop-down list.

Therefore, three different evaluation metrics are described next. The *Correct* metric is a classical recognition rate computed as $\frac{correct}{all}$ where *correct* represents the number of correctly recognized faces and *all* is the number of all recognized faces. The second metric, called *Correct in 5 most similar*, considers a correct recognition result when a correct face belongs to the set of the *five* most similar faces. The last metric is similar to the previous one, while a correct result is considered when a correct face belongs to the set of the *ten* most similar (*best*) recognized faces.

The first section of Table 5.13 presents the recognition rates of the SIFT based adapted Kepenekci method. In the second section, results of the Adapted Kepenekci method based on Gabor wavelets are presented. This experiment further confirms significantly better performance of the SIFT features in comparison with Gabor wavelet based features on the large real-world corpus.

Results of the SURF Descriptor Integration

This experiment was carried out with the combined SIFT/SURF descriptor. We would like to compare its results with the previously evaluated approaches.

Table 5.14 compares the previously proposed SIFT based adapted Kepenekci method, the SURF based Kepenekci method, U-SURF based Kepenekci method and the proposed combined SIFT/SURF descriptor. Note that SURF and U-SURF based Kepenekci methods are approaches that use SURF or U-SURF features with the adapted Kepenekci matching.

This experiment shows that the proposed combined SIFT/SURF descriptor with adapted Kepenekci matching method outperforms the other approaches. The difference between recognition error rate of the “best” proposed approach and the second best one is 7% in relative value.

Table 5.14: Recognition results of some most efficient AFR methods on the ČTK dataset

Method	**ACC [%]**
SIFT based Kepenekci	64.73
SURF based Kepenekci	61.13
U-SURF based Kepenekci	60.34
Combined SIFT/SURF	**65.83**

5.5.6 Recognition Results on the FERET Database

To allow straightforward comparison of our methods with other state of the art methods we made experiments on the FERET database. Table 5.15 compares and evaluates our proposed approaches with other very efficient methods on a large subset of the FERET corpus. The *fa* set is used for training, while the *fb* set is for testing. The motivation of the experiments is to show the performance of the method on larger corpus. Moreover the character of images contained in the FERET dataset is less controlled than in case of ORL which is beneficial for our aim - recognition on large corpus of uncontrolled images.

The first reported method is an implementation of the EBGM algorithm by Bolme [4]. The second algorithm is proposed by Ahonen in [50]. This approach is based on Local binary patterns. Wagner et al. propose in [73] another efficient approach, based on the Sparse Representation and Classification (SRC) algorithm, whose score is reported next. The results of the novel approach based on linear regresion proposed by Naseem et al. [36] are shown in the following line of this table. Recognition accuracy of the novel Discriminative Multi-Manifold Analysis (DMMA) method (Lu) [106] is presented next. The sixth method is introduced by Kepenekci in [62] and is based on the Gabor wavelets. The last two approaches are the proposed implementations of the SIFT based adapted Kepenekci method and the combined SIFT/SURF descriptor used with adapted Kepenekci matching.

The table shows that the recognition rates of all approaches are very good and close to one another. This experiment also shows that our SIFT based Kepenekci method slightly outperforms the other approaches and the combined SIFT/SURF descriptor reaches even better performance.

5.6 Conclusions

The first proposed method is based on the Gaussian mixture models. The experiments on the controlled ORL dataset showed that the recognition abilities of this

Table 5.15: Recognition results of various very efficient AFR methods on the FERET dataset

No.	Method	ACC [%]
1.	EBGM (Bolme)	96.4
2.	Local binary patterns (Ahonen)	97.0
3.	SRC algorithm (Wagner)	95.2
4.	Linear regression (Naseem)	93.5
5.	DMMA (Lu)	93.0
6.	Kepenekci method (Kepenekci)	96.3
7.	**SIFT based Kepenekci method (Lenc)**	**97.3**
8.	**Combined SIFT/SURF (Lenc)**	**98.4**

method are very good on the small dataset recorded in laboratory condition. Using several tens face images acquired in controlled conditions the method outperforms many other successful methods reaching the recognition error rate of 3.25%. However, from the nature of the method, its usability for real-world data is limited. The experiments on a subset of the ČTK dataset of the comparable size with ORL revealed that the recognition accuracy decreases significantly when using real-world images. Therefore, we decided to test another face recognition methods and try to find more suitable one for our purposes.

The next candidate was the Adapted Kepenekci method. We adapted the original Kepenekci method to use more training images to create a face model(so called *composed face*) and proposed to use larger sliding window in the fiducial point determination step. We also omitted the similarity threshold used in the original method. First, we made experiments on the ORL and Stirling datasets. The results were close to 100% for these controlled datasets. We also experimented with the parameters of the method in order to find the best values for particular datasets. We proved that the parameters depend mainly on the size of recognized images. We have shown that this approach is better in all cases than the previous one especially using real world photographs from the ČTK dataset. Using 7 out of 8 images for training, we achieved recognition rate of 50%. It is a significant improvement over the GMM, however the accuracy is still too low for our application.

Therefore, we proposed a method using the same matching scheme as the Adapted Kepenekci method. However, this method uses more suitable SIFT features for face representation. A thorough comparison of these two methods was performed and it proved that the SIFT based method outperforms the Adapted Kepenekci method. Especially in the case of real-world photographs, the differences are apparent. Another comparison was made using the FERET database. The SIFT based method also brought better results than the compared methods. Its obtained recognition rate on the ČTK dataset is 64.73% which can be sufficient for our face recognition system. However, we would like still to improve this result.

Therefore, we proposed a method which combines two descriptors to create the face representation. The combination of SIFT and SURF features increases the representational ability of the face models. Also in this case, the matching scheme used in the Adapted Kepenekci method is used. This method achieved higher recognition rates on both FERET and ČTK datasets.

Chapter 6

Confidence Measures

The motivation to use Confidence Measures (CM) is the evaluation of face recognition results. Based upon suitable metrics that characterize the recognition result or the quality of a face model, we would like to decide whether the recognition result is correct or not. As in many other papers [107, 88, 94], we propose two confidence measures based on the estimation of the *posterior* class probability. Further, we propose two methods based on predictor features. Finally, all partial approaches will be integrated into one robust supervised confidence measure method.

6.1 Posterior Class Probability Approaches

Let $P(F|C)$ be the output of the classifier, where C is the recognized face class and F represents the face features. The values $P(F|C)$ are normalized to compute the *posterior* class probabilities as follows:

$$P(C|F) = \frac{P(F|C).P(C)}{\sum_{I \in \mathcal{FIM}} P(F|I).P(I)} \tag{6.1}$$

$\mathcal{FIM}$ represents the set of all individuals and $P(C)$ denotes the *prior* probability of the individual's (face) class C.

We propose two different approaches. In the first approach, called ***absolute confidence value***, only faces $\hat{C}$ complying with

$$\hat{C} = \arg\max_{C}(P(C|F)) \tag{6.2}$$

$$P(\hat{C}|F) > T \tag{6.3}$$

are considered as being recognized correctly.

The second approach, called ***relative confidence value***, computes the difference between the *best* score and the *second best* one by the following equation:

$$P\Delta = P(\hat{C}|F) - \max_{C \neq \hat{C}}(P(C|F)) \tag{6.4}$$

Only the faces with $P\Delta > T$ are accepted. This approach aims to identify the "dominant" faces among all the other candidates. T is the acceptance threshold and its optimal value is adjusted experimentally.

6.2 Predictor Feature Approaches

This type of approaches uses the features with a maximal discriminability between the correct and incorrect classes to classify the recognition results. Two measures are proposed next.

The first one is based on the number of vectors in the model with the highest output value during the recognition task (i.e. the recognized face model). The number of vectors is given by the results of the SIFT algorithm. A face model with a high number of vectors is more general and it can be more likely identified as a good one. Conversely, a few vector face model is more accurate. Therefore, when this model is chosen as a good one (the highest output value) we assume that it is very probable that the recognition is correct.

Let V be the number of vectors in the face model and let T be the acceptance threshold. Only the faces where $V < T$ are accepted. The optimal value of the threshold T will be set experimentally. This measure is hereafter called the ***vector number*** approach.

The second measure uses a standard deviation of the similarities among images in the recognized face model. Let the recognized model M be composed of the images $I_1, I_2, ..., I_N$. The S measure is defined as follows:

$$S = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(FS_{I_i,M\setminus I_i} - \mu)^2} \tag{6.5}$$

where $FS_{I_i,M\setminus I_i}$ is the similarity (see Equation 5.7) of the image I_i and a model $M \setminus I_i$ created from the remaining images from model M and μ is computed by the following equation:

$$\mu = \frac{1}{N}\sum_{i=1}^{N} FS_{I_i,M\setminus I_i} \tag{6.6}$$

Similarly as in the case of the ***vector number*** measure we suppose that higher standard deviation characterizes a more general face model and vice versa. Therefore, only the recognition results where $S < T$ are accepted. The optimal value of the acceptance threshold T will be set experimentally. This measure is hereafter called the ***standard deviation*** approach.

6.3 Composed Supervised Approach

Let R_k be the score obtained by a partial unsupervised measure k described above and let variable H determines whether the face image is classified correctly or not. A Multi-layer Perceptron (MLP) which models the *posterior* probability $P(H|R_1, .., R_N)$ is used to combine all partial measures in a supervised way. Note that the variable N represents the number of measures to combine

In order to identify the best performing topology, several combinations and MLP configurations are built and evaluated. The MLP topologies will be described in detail in the experimental section.

6.4 Experiments

In this section, we present experiments that we made on the recognition results with the goal to find a suitable confidence measure.

6.4.1 Discriminability of the Proposed Measures

First, we evaluate two methods based on the posterior probability: *absolute confidence measure* and *relative confidence measure*. Then, the proposed predictor features are evaluated: *vector number* and *standard deviation*. Further we evaluate a composed supervised confidence measure using the separate measures as an input of an MLP classifier.

We create two histograms for every measure in order to analyse the distribution of the correctly and incorrectly classified faces. The reported output densities of the measures are based on the 638 values (the number of individuals in the corpus). Note that all output values are normalized to the interval [0..1].

Figure 6.1 shows the output densities of the correctly and incorrectly classified faces when the *absolute confidence value* measure is used. These histograms show that the majority of the correctly recognized face examples has higher output values than the incorrectly recognized ones. This fact confirms our assumption that the first measure is suitable for our task and should be useful to be integrated to the whole composed method.

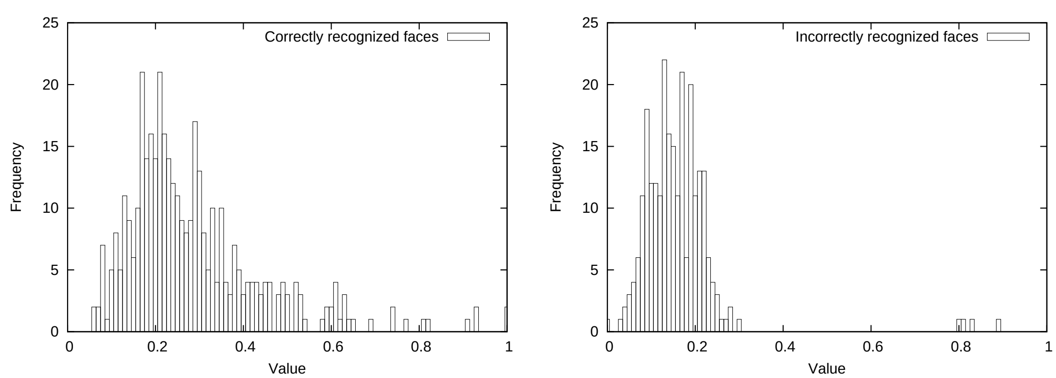

Figure 6.1: Histograms of the correctly (left) and incorrectly (right) classified faces using the *absolute confidence value* measure

Figure 6.2 plots the output densities when the *relative confidence value* measure is used. These histograms show clearly that the discriminability of this measure is better than the previous one. Almost all correctly recognized face examples have higher output values than the incorrectly recognized samples. Therefore, this measure should be suitable for our task and we decided to combine it with the other ones by an MLP. Moreover, we assume that this measure used separately outperforms the previously proposed one.

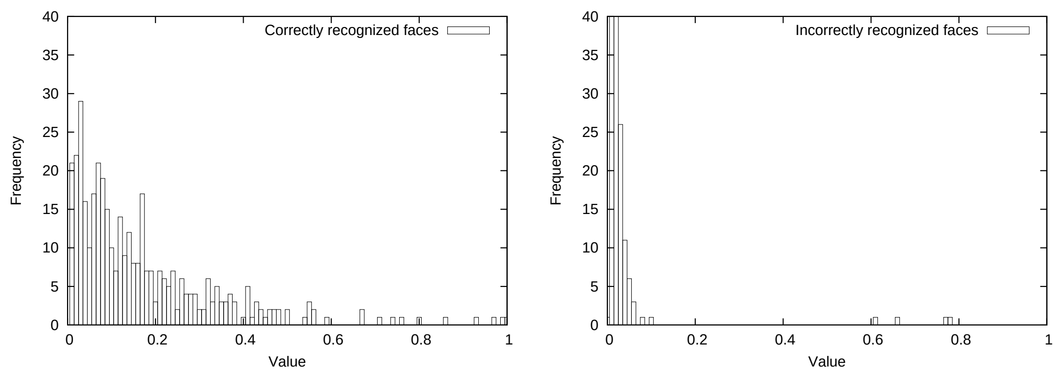

Figure 6.2: Histograms of the correctly (left) and incorrectly (right) classified faces using the *relative confidence value* measure

Figure 6.3 depicts the output densities when the *vector number* measure is used. These histograms show that this measure is less discriminant than the two ones presented previously. However, the correctly recognized examples have slightly inferior output values than the incorrect ones. This fact confirms our assumption (see Sec. 6.2) that the confidence of a few vector model is high. We assume that this measure will bring poor results if used separately. However, it can add some further information in combination with the other approaches. Therefore, we decided to integrate it into the whole composed approach.

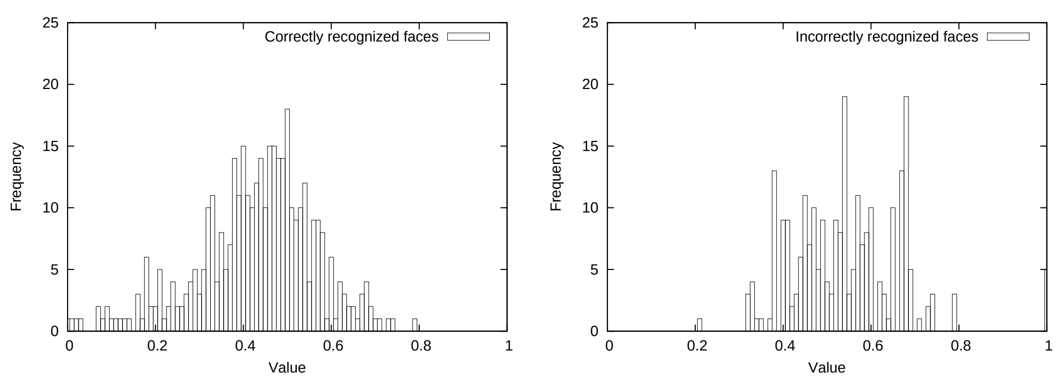

Figure 6.3: Histograms of the correctly (left) and incorrectly (right) classified faces using the *vector number* measure

The output densities of the last *standard deviation* measure are reported in Figure 6.4. The discriminability of these two histograms is limited and it is difficult to propose some conclusions about this measure. However, we decided to use this measure in further experiments and verify its usefulness experimentally.

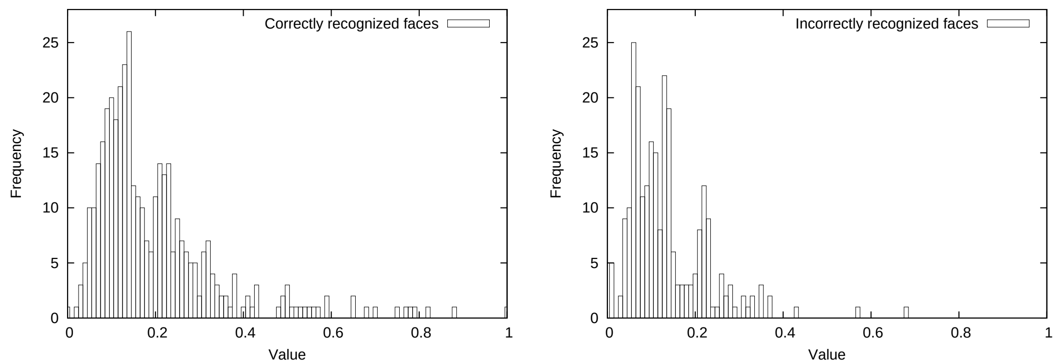

Figure 6.4: Histograms of the correctly (letf) and incorrectly (right) classified faces using the *standard deviation* measure

To summarize:

- *Relative confidence value (rel)* measure is the best proposed one,
- *Absolute confidence value (abs)* method has also very good separation abilities,
- *Vector number (vect)* measure can bring some complementary information for our task,
- Contribution of the *standard deviation (sd)* measure is questionable and must be confirmed experimentally.

6.4.2 Accuracy of the Separate Measures

In the second experiment we would like to show the performance of the above described measures used separately without any combination. As in many other articles in the confidence measure field, we will use the Receiver Operating Characteristic (ROC) curve [108] for evaluation of this experiment. This curve clearly shows the relationship between the true positive and false positive rates for the different *acceptance* threshold.

Figure 6.5 shows the results of the separately used *absolute confidence value*, *relative confidence value*, *vector number* and *standard deviation* measures. This experiment shows that the *relative confidence value* method significantly outperforms all other approaches.

We can further deduce that our assumption in the fourth proposed measure was not correct. Based on this experiment we can consider that the dependence between the value of the standard deviation and the correctly recognized faces is reversed. We modify the definition of such measure as follows: only the faces where $S > T$ are accepted.

After this modification we can conclude that all proposed measures are suitable for our task in order to identify incorrectly recognized faces. Note that the corrected version of the ROC curve of the fourth *standard deviation* measure is reported in this figure with the *modified sd* caption.

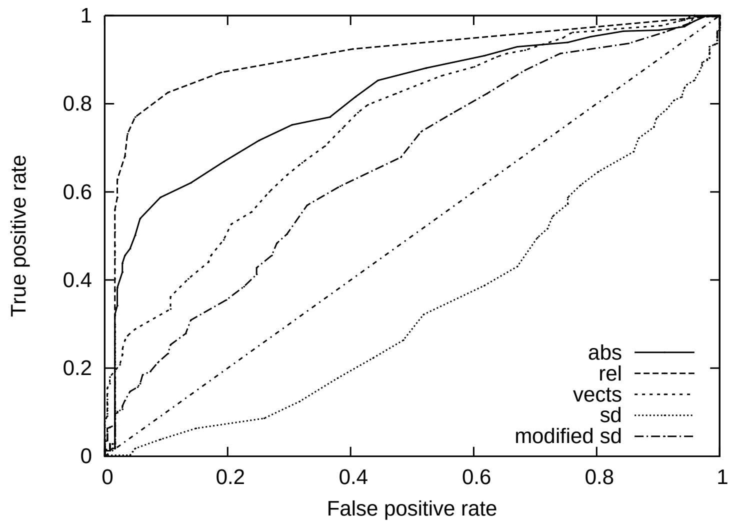

Figure 6.5: ROC curves of the four proposed measures used separately. The corrected *standard deviation* measure is reported with the *modified sd* label.

We will further compare the results of the separate measures with the whole composed approach. Therefore, we created Table 6.1 to show the scores of the separate measures with optimal threshold configurations. The F-measure (F-mes) [109] is used as an evaluation metric, the Precision (Prec) and Recall (Rec) are also reported in this table. Note that the optimal threshold $\hat{T}$ value has been defined for the "best" compromise between precision and recall values as follows:

$$\hat{T} = \arg\min_{T} \mid 1 - \frac{Prec}{Rec} \mid \tag{6.7}$$

Table 6.1: Performance of the measures used separately [%]

Confidence Measure	**Prec**	**Rec**	**F-mes**
absolute confidence value	65.7	60.6	63.0
relative confidence value	69.6	60.8	64.9
vector number	62.2	63.5	62.8
standard deviation	58.9	60.3	59.6

6.4.3 Accuracy of the Whole Composed Approach

In the last experiment, we will evaluate the results of the whole composed confidence measure method. First, we will show the impact of the use of an MLP classifier with the separate measures. Then, we compare and evaluate all possible combinations of the proposed measures in order to show the complementarities among them.

Several MLP configurations are tested. The best MLP topology uses three layers. The number of the input neurons corresponds to the number of measures to combine, 10 neurons are in the hidden layer and two outputs are used to identify the *correctly* and *incorrectly* recognized faces. This MLP topology was set empirically on a small development corpus which contains 120 examples (i.e. 120 confidence values).

The results of this experiment are reported in Table 6.2. These results show that the separate measures used with an MLP have better F-measure values (except *sd* approach) than used in the unsupervised way. A successive addition of the measures improves progressively the F-measure value. When all measures are combined, the resulting F-measure is close to 100%. This figure also shows that all measures bring complementary relevant information and are thus useful to be integrated to the whole composed approach (i. e. the whole combined approach gives the best recognition score).

Table 6.2: Performance of all combinations of the measures by an MLP classifier [%]

Confidence Measure	**Prec**	**Rec**	**F-mes**
1. Separate measures			
abs. confidence value (abs)	92.5	64.8	76.2
rel. confidence value (rel)	96.2	80.4	87.6
vector number (vect)	55.4	84.9	67.0
standard deviation (sd)	54.0	65.3	59.1
2. Combinations of two measures			
abs, rel	97.2	83.5	89.8
abs, sd	70.4	55.8	62.2
abs, vect	95.8	75.8	84.6
rel, sd	95.8	84.3	89.7
rel, vect	97.7	85.6	91.2
sd, vect	67.6	90.6	77.4
3. Combinations of three measures			
abs, rel, sd	96.7	90.0	93.2
abs, rel, vect	97.2	93.7	95.4
abs, sd, vect	93.4	90.5	91.9
rel, sd, vect	94.8	94.8	94.8
4. Combination of all measures (the whole approach)			
abs, rel, sd, vect	100	99.5	**99.8**

6.5 Conclusions

In this chapter, we proposed and evaluated several confidence measure methods for the face recognition. First, we analysed the discrimination abilities of four individual confidence measure. We showed that the *relative confidence value* works very well and that the other methods bring another useful information. Evaluation of the methods confirmed that each of the separate methods has some amount of discriminability. Finally, we evaluated the composed supervised approach which combines the individual measures using an MLP classifier. The composed approach outperforms the individual measures and is suitable for usage in our system.

Chapter 7

Face Recognition System

The goal of this chapter is to describe the whole proposed face recognition system. Based on the experiments and conclusions in three previous chapters we decided to use the best suitable methods to be integrated into the system. Our decisions depend mainly on the fact that we need to handle a significant amount of *real-world* photographs of variable quality.

7.1 System Architecture

The presented system has (as shown in Figure 7.1) a modular architecture. It is composed of five modules (see the rectangles) connected by dependencies (see the oriented edges). The input image and the recognition results are represented by parallelograms. The storage of the face representation is marked by the *Face Gallery* sign.

The first module $M1$ deals with face extraction. This module converts a color image into its grey-scale representation. Then it performs face detection. The detected face is further extracted from the image in the next step. This module also detects the eyes in the detected face region and transforms and resizes the face.

The second module $M2$ is used to create the face representation. It detects the SIFT key-points and creates a set of SIFT descriptors for a representation of the face image.

The next module $M3$ is used to select the most representative face vectors in order to create a precise face model M. The algorithm implemented within this module is described in Section 4.1.5.

The fourth module $M4$ is responsible for face recognition. A recognized face is compared to the face models stored in the *Face Gallery* and the most similar model is chosen as the recognized one.

The last *confidence measure* module $M5$ is dedicated to identifying whether the recognition result is correct or not. This unique step is particularly important, because when the user knows that the recognition is probably not correct, he can manually correct the recognition result.

Note that the modules $M1$ and $M2$ are used in both, face representation (or

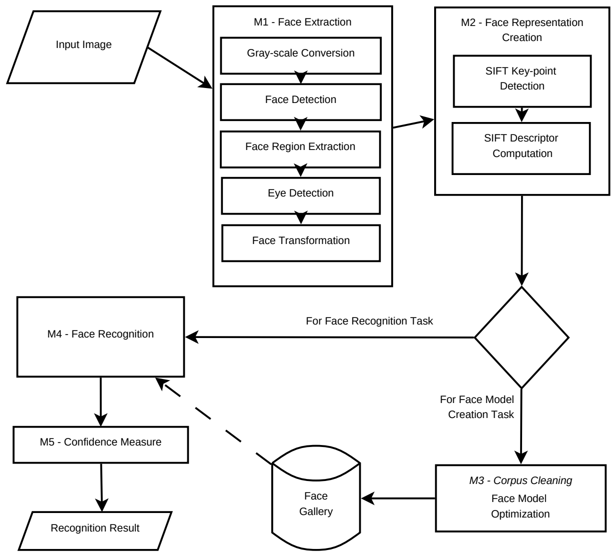

Figure 7.1: System Architecture

modelling) and face recognition tasks. However, module $M3$ is used only for face representation and modules $M4$ and $M5$ are used only for recognition. The last remark is that each module could be used separately and be easily integrated into another face processing system.

M1 - Face Extraction Module

The input of this module is an ordinary photograph containing a human face. First, the image is converted to gray-scale and histogram equalization is performed. The Viola-Jones algorithm described in Section 4.1.1 is used for face detection.

This module further performs eye detection which provides information whether the face detection is correct (see conclusions in Section 4.3).

According to the positions of detected eyes, the face is rotated, resized and placed into the center of the image. The output of this module is thus a gray-scale face image suitable for a further processing.

M2 - Face Representation Module

The goal of this module is to create a representation of one image. The input is a cropped face image prepared by the first module (M1). According to the experiments in Section 5.5.5 we chose the SIFT features to create the representation. Despite the slightly higher recognition rate obtained with the combined SIFT/-SURF descriptor we decided to use only the SIFT features. The reason is that the improvement of recognition rate is only small and that the combined descriptor causes higher computational demands. The output of this module is a set of feature vectors representing the particular face.

M3 - Corpus Cleaning Module

This module is used to create a gallery of face representations modelling the known faces. Its input is a set of face representations obtained from all labelled images using module M2. The number of face images for each person differs significantly. But for the performance of the system it is beneficial if the number of examples is comparable. The goal of this module is thus to chose only the most representative face images for each person. For this task, the corpus cleaning algorithm described in detail in Section 4.1.5 is used. To some extent, also the mislabelled faces present in the photographic database are removed. After this step, the face models are created and stored into the gallery. The output of this module is thus a face gallery with balanced numbers of examples for each person.

M4 - Face Recognition Module

The face recognition module is the crucial part of the whole system. Because of the nature of photographs we use for recognition, it was necessary to find out a method performing well under uncontrolled conditions. In the experiments (see Section 5.5) we tested several methods and proposed some modifications in order to handle the real-world data. According to the extensive experiments performed on the ČTK dataset we chose the SIFT features for face representation. The face representations are compared using the Adapted Kepenekci matching scheme.

The input of this module is a face image containing an unknown person extracted by the face extraction module (M1). The face is compared against all the models stored in the gallery. The output are similarities of the face being recognized to the gallery models.

M5 - Confidence Measure Module

In order to make the system more accurate and reveal the incorrectly recognized faces, we decided to employ also the confidence measure. According to the experiments presented in Section 6.4 we chose the Combined supervised approach for this task. The input of this module are the similarities obtained from the face recognition module (M4). The output is a probability that the gallery face with the highest similarity is the correct face.

7.2 Conclusions

The resulting system is capable to be used for the face recognition under real-world conditions. The face recognition method we used has sufficiently good accuracy. Moreover, the confidence measure module adds a substantial information that helps to verify the recognition results. The results in this form are suitable for practical usage. Currently, the deployment in the ČTK environment is discussed.

Chapter 8

Conclusions and Perspectives

8.1 Conclusions

The automatic face recognition is nowadays one of the most progressing biometric techniques. It allows an identification or verification of human faces stored in digital images or video frames. In this work, we concentrate on the identification scheme. It means we have a photo of an unknown person and we compare it against a gallery of known people. We search for the most similar face representation from the gallery to find who is in the tested image.

This thesis is focused on the face recognition under real-world conditions. It means recognizing people from "ordinary" photographs (uncontrolled scenario), not from images acquired directly for the face recognition in laboratory conditions (well controlled scenario). The main goal is to propose a fully automated face recognition system capable labelling people in these photographs.

The designed system is intended to be used by the Czech News Agency. The ČTK owns huge number of photographs. We used the labelled images (only certain part of the database) and create face representations from them. This is used as a gallery of known people. Further we compare the unlabelled images with the gallery and choose the most similar face representation. Based on the recognition result, the image is annotated. Another usage is labelling of newly acquired images which are inserted into the database.

During the implementation of the system several scientific challenges must be solved and thus novel or adapted approaches were proposed. The research was focused on the following three main topics.

The first challenge was to prepare a face corpus from the labelled part of the ČTK photographic database. This step is necessary to allow recognition of unlabelled and newly acquired images. The task is made difficult by the varying appearance of the faces in this database. Another complications are significantly different numbers of examples for each individual and possible incorrectly labelled images.

We proposed a novel corpus creation algorithm which automatically creates face corpus from a set of real-world photographs. It detects the faces in the photographs and corrects the rotation of the face. Finally, it creates a face representation for each person. The algorithm was designed for the specific needs of our application where multiple images for each person are present. It is further assumed that a certain part of photograph labels is erroneous and therefore the corpus cleaning algorithm is applied to eliminate these errors.

Using the corpus creation algorithm, we created a new facial corpus. This corpus is available for free for scientific purposes. It is possible to download it from `<http://home.zcu.cz/~pkral/sw/>` or upon request to the authors.

Despite the high accuracy of existing methods on controlled databases they often fail when real-world data are used. We thus had to propose and choose a suitable method for the recognition module of our system. Several methods have been developed, evaluated and compared. According to the experimental results we decided to use the SIFT based adapted Kepenekci method for this task. This method achieved very interesting results on the standard face datasets we tested. Moreover, the dominance of the method is even more clear from the results on our ČTK database. In comparison with other tested methods the recognition accuracy is significantly higher.

The results obtained from the face recognition module are still not good enough for the application. Therefore, it is important to verify the recognition results and decide whether they are correct. We proposed a novel confidence measure approach to perform this task. The Two-step Composed supervised confidence measure is based on four measures. Two of them are based on the posterior probability. The other two ones are based on characteristics of the recognized face model. In the second step, these four measures are combined using an MLP classifier. The proposed confidence measure approach allows to successfully decide whether the recognition result is correct in more than 90% of cases.

The main result of this work is a complete face recognition system. The system allows labelling of real-world photographs with very high accuracy and is available for free for research purposes. The deployment of the system in the ČTK is currently negotiated.

8.2 Perspectives

The perspectives for future work are numerous. First, there is still a lot of room for further improvement in the corpus creation process. One possibility is proposition of more sophisticated face transform in the pre-processing step. Further, the confidence measure approach could be adapted and used in this corpus creation task.

For the face recognition itself, a more robust algorithm could be proposed. A promising direction is the SURF based descriptor and its combination with the SIFT descriptor. We made a preliminary experiment confirming that the combination increases robustness and recognition accuracy. Its drawback are higher computational demands. Proposing a better way how to combine the descriptors could bring high accuracy while the computational costs decrease.

Another important issue is the licence of the SIFT and SURF methods. Both methods are patented and the commercial use is expensive. Therefore, we will concentrate also on other methods allowing free commercial use. We performed a preliminary study of methods based on the LBP descriptor [110]. The tested method currently does not achieve as high recognition rate as the SIFT based method but the difference is not drastic and further improvement is still possible.

The perspective for the confidence measure approach is proposing a confidence measure model which is progressively adapted according to the processed data. There is also possibility of finding additional features used for the confidence measure.

Another perspective is to handle automatically the case when the presented face is not in the gallery. If the person is known, the image could be added to the existing representation and make it more accurate. If the person is not yet present in the database, a new face model is created and inserted into the database.

List of Acronyms

AAM Active Appearance Model

AFR Automatic Face Recognition

ALH Adaptive Local Hyperplane

CM Confidence Measure

CTK Czech News Agency

DCT Discrete Cosine Transform

DTLBP Dynamic Threshold Local Binary Pattern

EBGM Elastic Bunch Graph Matching

EP Evolutionary Pursuit

FERET The Facial Recognition Technology

FLD Fisher Linear Discriminant

FSIFT Fixed Key-point SIFT

GA Genetic Algorithm

GMM Gaussian Mixture Model

GWT Gabor Wavelet Transform

HKNN K-local Hyperplane distance Nearest Neighbour

HMM Hidden Markov Model

ICA Independent Component Analysis

KFLD Kernel Fisher Linear Discriminant

KPCA Kernel Principal Component Analysis

LBP Local Binary Pattern

LDA Linear Discriminant Analysis

LDP Local Derivative Pattern

LFW Labeled faces in the Wild

LGBPH Local Gabor Binary Pattern Histogram

LTP Local Ternary Pattern

MLP Multilayer Perceptron

NN Neural Network

ORL Olivetti Research Laboratory

PCA Principal Component Analysis

RBF Radial Basis Function

ROC Receiver Operating Characteristic

RST Radial Symmetry Transform

SIFT Scale Invariant Feature Transform

SOM Self Organizing Map

SURF Speeded-up Robust Features

SVM Support Vector Machine

SVM-DA SVM-based Discriminant Analysis

TT Trace Transform

U-SURF Upright Speeded-up Robust Features

2D-DCT 2D-Discrete Cosine Transform

2DLDA 2D Linear Discriminant Analysis

2DPCA 2D Principal Component Analysis

Bibliography

Journal Publications

[1] Lenc, L. and Král, P., "Automatic Face Recognition Approaches," *Journal of Theoretical and Applied Information Technology*, ISSN: 1992-8645, Volume 59, No 3, January 2014.

[2] Lenc, L. and Král, P., "Automatic Face Recognition System Based on the SIFT Features," *Computers and Electrical Engineering*, ISSN: 0045-7906 - under review.

Conference Publications Indexed in Scopus

[3] Lenc, L. and Král, P., "Automatic face recognition: Methods improvement and evaluation," *ICAART 2011 - Proceedings of the 3rd International Conference on Agents and Artificial Intelligence*, Volume 1, Pages 604-608, January 2011.

[4] Lenc, L. and Král, P., "Confidence measure for automatic face recognition," *KDIR 2011 - Proceedings of the International Conference on Knowledge Discovery and Information Retrieval*, Pages 365-368, October 2011.

[5] Lenc, L. and Král, P., "Novel matching methods for automatic face recognition using SIFT," *IFIP Advances in Information and Communication Technology*, Volume 381 AICT, Issue PART 1, Pages 254-263, September 2012.

[6] Lenc, L. and Král, P., "Face recognition under real-world conditions," *ICAART 2013 - Proceedings of the 5th International Conference on Agents and Artificial Intelligence*, Volume 2, Pages 250-256, February 2013.

[7] Lenc, L. and Král, P., "Automatic face corpus creation," *ICAART 2013 - Proceedings of the 5th International Conference on Agents and Artificial Intelligence*, Volume 2, Pages 582-586, February 2013.

[8] Lenc, L. and Král, P., "A Combined SIFT/SURF Descriptor for Automatic Face Recognition," *ICMV 2013 - Proceedings of the 6th International Conference on Machine Vision*, Volume 9067, Pages 90672C-90672C-6, November 2013.

[9] Král, P. and Lenc L., "A Composed Confidence Measure for Automatic Face Recognition in Uncontrolled Environment," *ICAART 2014 - Proceedings of the 6th International Conference on Agents and Artificial Intelligence*, Pages 250-256, March 2014.

Other Conference Publications

[10] Lenc, L. and Král, P., "Gabor Wavelets for Automatic Face Recognition," *38th International Conference on Current Trends in Theory and Practice of Computer Science*, January 2012.

[11] Lenc, L. and Král, P., "Automatically Detected Feature Positions for LBP Based Face Recognition," *AIAI 2014 - 2014 The 10th Artificial Intelligence Applications and Innovations*, September 2014 - accepted.

[12] Lenc, L. and Král, P., "Two-step Supervised Confidence Measure for Automatic Face Recognition," *MLSP 2014 - IEEE International Workshop on Machine Learning for Signal Processing*, September 2014 - accepted.

References

[1] MIS Biometrics, "Mis biometrics home," 2008.

[2] S. Lawrence, S. Giles, A. Tsoi, and A. Back, "Face recognition: A convolutional neural network approach," *IEEE Trans. on Neural Networks*, 1997.

[3] A. V. Nefian and M. H. Hayes, "Hidden markov models for face recognition," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1998.

[4] David S. Bolme, *Elastic Bunch Graph Matching*, Ph.D. thesis, Colorado State University, 2003.

[5] David G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 2, 2004.

[6] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Conference on Computer Vision and Pattern Recognition*, 2001.

[7] Rahul D Chaudhari, Ashok A Pawar, and Rakesh S Deore, "The historical development of biometric authentication techniques: A recent overview," *International Journal of Engineering*, vol. 2, no. 10, 2013.

[8] Alphonse Bertillon, "Signaletic instructions: Including the theory and practice of anthropometrical identification, trans," *RW McClaughry (Chicago: American Bertillon Prison Bureau, 1896)*, 1912.

[9] Edward Richard Henry, *Classification and uses of finger prints*, George Routledge and Sons, 1900.

[10] John G Daugman, "Biometric personal identification system based on iris analysis," Mar. 1 1994, US Patent 5,291,560.

[11] W. W. Bledsoe, "Man-machine facial recognition," Tech. Rep., Panoramic Research Inc., Palo Alto, CA, 1966.

[12] L. Sirovich and M. Kirby, "Low-dimensional procedure for the characterization of human faces," *Journal of the Optical Society of America*, vol. 4, 1987.

[13] Alex P. Pentland, B. Moghaddam, and T. Starner, "View-based and modular eigenspaces for face recognition," in *IEEE Computer Society Conference on In Computer Vision and Pattern Recognition*. Computer Vision and Pattern Recognition, 1994.

[14] Wenyi Zhao, Rama Chellappa, P Jonathon Phillips, and Azriel Rosenfeld, "Face recognition: A literature survey," *Acm Computing Surveys (CSUR)*, vol. 35, no. 4, pp. 399–458, 2003.

[15] Linlin Shen and Li Bai, "A review on gabor wavelets for face recognition," *Pattern analysis and applications*, vol. 9, no. 2-3, pp. 273–292, 2006.

[16] David G. Lowe, "Object recognition from local scale-invariant features," in *International Conference on Computer Vision*, 1999.

[17] P. J. Phillips, H. Wechsler, J. Huang, and P. Rauss, "The FERET database and evaluation procedure for face recognition algorithms," *Image and Vision Computing*, vol. 16, no. 5, pp. 295–306, 1998.

[18] A.J. Goldstein, L.D. Harmon, and A.B. Lesk, "Identification of human faces," vol. 59, no. 5, pp. 748–760, May 1971.

[19] T. Kanade, *Computer recognition of human faces*, Birkhauser Verlag, 1977.

[20] Matthew A. Turk and Alex P. Pentland, "Face recognition using eigenfaces," in *IEEE Computer Society Conference on In Computer Vision and Pattern Recognition*. Computer Vision and Pattern Recognition, 1991.

[21] A. Hyvärinen and E. Oja, "Independent component analysis: algorithms and applications," *Neural Netw.*, vol. 13, no. 4-5, pp. 411–430, May 2000.

[22] Bruce A. Draper, Kyungim Baek, Marian Stewart Bartlett, and J. Ross Beveridge, "Recognizing faces with PCA and ICA," in *COMPUTER VISION AND IMAGE UNDERSTANDING, SPECIAL ISSUE ON FACE RECOGNITION*, 2003, pp. 115–137.

[23] Marian Stewart Bartlett, Javier R. Movellan, and Terrence J. Sejnowski, "Face recognition by independent component analysis," *IEEE Transactions on Neural Networks*, pp. 1450–1464, 2002.

[24] Peter N. Belhumeur, João P. Hespanha, and David J. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1997.

[25] Ming hsuan Yang, "Face recognition using kernel methods," 2001.

[26] Stan Z Li and Anil K Jain, *Handbook of face recognition*, springer, 2011.

[27] A.S. Georghiades, P.N. Belhumeur, and D.J. Kriegman, "From few to many: Illumination cone models for face recognition under variable lighting and pose," *IEEE Trans. Pattern Anal. Mach. Intelligence*, vol. 23, no. 6, pp. 643–660, 2001.

[28] Dattatray V Jadhav and Raghunath S Holambe, "Rotation, illumination invariant polynomial kernel fisher discriminant analysis using radon and discrete cosine transforms based features for face recognition," *Pattern Recognition Letters*, vol. 31, no. 9, pp. 1002–1009, 2010.

[29] Stanley Roderick Deans, *The Radon transform and some of its applications*, DoverPublications. com, 2007.

[30] Tao Yang and Vojislav Kecman, "Face recognition with adaptive local hyperplane algorithm," *Pattern Analysis and Applications*, vol. 13, no. 1, pp. 79–83, 2010.

[31] Pascal Vincent and Yoshua Bengio, "K-local hyperplane and convex distance nearest neighbor algorithms," in *Advances in Neural Information Processing Systems*, 2001, pp. 985–992.

[32] Suresh Balakrishnama and Aravind Ganapathiraju, "Linear discriminant analysis-a brief tutorial," *Institute for Signal and information Processing*, 1998.

[33] Chengjun Liu and Harry Wechsler, "Face recognition using evolutionary pursuit," in *University of Freiburg*, 1998, pp. 2–6.

[34] Sanun Srisuký, Kongnat Ratanarangsanký, Werasak Kurutach, and Sahatsawat Waraklang, "Face recognition using a new texture representation of face images," 2003.

[35] A. Martinez and R. Benavente, "The ar face database," Tech. Rep., Univerzitat Autonòma de Barcelona, 1998.

[36] Imran Naseem, Roberto Togneri, and Mohammed Bennamoun, "Linear regression for face recognition," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 32, no. 11, pp. 2106–2112, 2010.

[37] Helge Toutenburg et al., *Linear models and generalizations: least squares and alternatives*, Springer, 2008.

[38] Gareth J. Edwards, Christopher J. Taylor, and Timothy F. Cootes, "Interpreting face images using active appearance models.," in *FG*. 1998, pp. 300–305, IEEE Computer Society.

[39] T. F. Cootes, K. Walker, and C.J. Taylor, "View-based active appearance models," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2000, pp. 484–498, Springer.

[40] David DeMers, Garrison Cottrell, et al., "Non-linear dimensionality reduction," *Advances in neural information processing systems*, pp. 580–580, 1993.

[41] Meng Joo Er, Shiqian Wu, Juwei Lu, and Hock Lye Toh, "Face recognition with radial basis function (rbf) neural networks," *Neural Networks, IEEE Transactions on*, vol. 13, no. 3, pp. 697–710, 2002.

[42] Shinichi Tamura, Hideo Kawai, and Hiroshi Mitsumoto, "Male/female identification from 8×6 very low resolution face images by neural network," *Pattern Recognition*, vol. 29, no. 2, pp. 331–335, 1996.

[43] F. Samaria and S. Young, "Hmm-based architecture for face identification," *Image and Vision Computing*, 1994.

[44] Peter M. Corcoran and Claudia Iancu, *New Approaches to Characterization and Recognition of Faces*, 2011.

[45] Bernd Heisele, Purdy Ho, and Tomaso Poggio, "Face recognition with support vector machines: Global versus component-based approach," in *ICCV 2001. Proceedings. Eighth IEEE International Conference on Computer Vision*. IEEE, 2001, vol. 2, pp. 688–694.

[46] Sang-Ki Kim, Youn Jung Park, Kar-Ann Toh, and Sangyoun Lee, "Svm-based feature extraction for face recognition," *Pattern Recognition*, vol. 43, no. 8, pp. 2871–2881, 2010.

[47] Terence Sim, Simon Baker, and Maan Bsat, "The cmu pose, illumination, and expression database," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 25, no. 12, pp. 1615–1618, 2003.

[48] Yin Zhang and Zhi-Hua Zhou, "Cost-sensitive face recognition," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 32, no. 10, pp. 1758–1769, 2010.

[49] Timo Ojala, Matti Pietikäinen, and David Harwood, "A comparative study of texture measures with classification based on featured distributions," *Pattern recognition*, vol. 29, no. 1, pp. 51–59, 1996.

[50] Timo Ahonen, Abdenour Hadid, and Matti Pietikäinen, "Face recognition with local binary patterns," in *Computer Vision - ECCV 2004*, Tomás Pajdla and Jiří Matas, Eds., vol. 3021 of *Lecture Notes in Computer Science*, pp. 469–481. Springer Berlin Heidelberg, 2004.

[51] Ahonen Timo, Abdenour Hadid, and Matti Pietikäinen, "Face description with local binary patterns: Application to face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, pp. 2037–2041, 2006.

[52] Weisheng Li, Peng Fu, and Lifang Zhou, "Face recognition method based on dynamic threshold local binary pattern," in *Proceedings of the 4th International Conference on Internet Multimedia Computing and Service*. ACM, 2012, pp. 20–24.

[53] Xiaoyang Tan and Bill Triggs, "Enhanced local texture feature sets for face recognition under difficult lighting conditions," *Image Processing, IEEE Transactions on*, vol. 19, no. 6, pp. 1635–1650, 2010.

[54] Hong Yang and Yiding Wang, "A lbp-based face recognition method with hamming distance constraint," in *Image and Graphics, 2007. ICIG 2007. Fourth International Conference on*. IEEE, 2007, pp. 645–649.

[55] Xiaoyang Tan and Bill Triggs, "Fusing gabor and lbp feature sets for kernel-based face recognition," in *Analysis and Modeling of Faces and Gestures*, pp. 235–249. Springer, 2007.

[56] Zhihua Xie, "Single sample face recognition based on dct and local gabor binary pattern histogram," in *Intelligent Computing Theories*, pp. 435–442. Springer, 2013.

[57] Baochang Zhang, Yongsheng Gao, Sanqiang Zhao, and Jianzhuang Liu, "Local derivative pattern versus local binary pattern: face recognition with high-order local pattern descriptor," *Image Processing, IEEE Transactions on*, vol. 19, no. 2, pp. 533–544, 2010.

[58] V. Blanz, S. Romdhani, and T. Vetter, "Face identification across different poses and illuminations with a 3d morphable model," in *International Conference on Automatic Face and Gesture Recognition*, 2002.

[59] W. Zhao and R. Chellappa, "3d model enhanced face recognition," in *International Conference on Image Processing*, 2000.

[60] L. Wiskott, J.-M. Fellous, N. Krüger, and C. von der Malsburg, "Face recognition by elastic bunch graph matching," *Intelligent Biometric Techniques in Fingerprint and Face Recognition*, 1999.

[61] L. Shen, *Recognizing Faces - An Approach Based on Gabor Wavelets*, Ph.D. thesis, University of Nottingham, 2005.

[62] B. Kepenekci, *Face Recognition Using Gabor Wavelet Transform*, Ph.D. thesis, The Middle East Technical University, 2001.

[63] Jae Byung Park and Avinash C Kak, "A truncated least squares approach to the detection of specular highlights in color images," in *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*. IEEE, 2003, vol. 1, pp. 1397–1403.

[64] P.-N. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining*, Addison-Wesley, 2005.

[65] J. Krizaj, V. Struc, and N. Pavesic, "Adaptation of sift features for robust face recognition," 2010.

[66] M. Aly, "Face recognition using sift features," 2006.

[67] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool, "Speeded-up robust features (surf)," *Comput. Vis. Image Underst.*, vol. 110, no. 3, pp. 346–359, June 2008.

[68] Geng Du, Fei Su, and Anni Cai, "Face recognition using surf features," pp. 749628–749628–7, 2009.

[69] Philippe Dreuw, Pascal Steingrube, Harald Hanselmann, and Hermann Ney, "Surf-face: Face recognition under viewpoint consistency constraints," in *Proceedings of the British Machine Vision Conference*. 2009, pp. 7.1–7.11, BMVA Press, doi:10.5244/C.23.7.

[70] Song Cao, "A fast surf way for human face recognition with cell similarity," in *Industrial Electronics and Applications (ICIEA), 2011 6th IEEE Conference on*, 2011, pp. 166–169.

[71] Manuele Bicego, Andrea Lagorio, Enrico Grosso, and Massimo Tistarelli, "On the use of sift features for face authentication," in *In: Conf. on Computer Vision and Pattern Recognition Workshop (CVPRW). (2006*. 2006, p. 35, IEEE Computer Society.

[72] Aleix M. Martinez, "Recognizing imprecisely localized, partially occluded and expression variant faces from a single sample per class," 2002.

[73] A. Wagner, J. Wright, A. Ganesh, Zihan Zhou, H. Mobahi, and Yi Ma, "Toward a practical face recognition system: Robust alignment and illumination by sparse representation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 2, pp. 372–386, 2012.

[74] Ke Huang and Selin Aviyente, "Sparse representation for signal classification," *Advances in neural information processing systems*, vol. 19, pp. 609, 2007.

[75] Paola Campadelli and Raffaella Lanzarotti, "A face recognition system based on local feature characterization," in *Advanced Studies in Biometrics*, pp. 147–152. Springer, 2005.

[76] C. Zhang and Z. Zhang, "A survey of recent advances in face detection," Tech. Rep., Microsoft Research, 2010.

[77] M.-H. Yang, D. j. Kriegman, and N. Ahuja, "Detecting faces in images: A survey," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2002.

[78] Henry A. Rowley, S. Baluja, and T. Kanade, "Neural network-based face detection," in *IEEE Pattern Analysis and Machine Intelligence*, 1998.

[79] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, 1997.

[80] L. Bai, L. Shen, and Y. Wang, "A novel eye location algorithm based on radial symmetry transform," in *International Conference on Pattern Recognition*, 2006.

[81] Z. Qian and D. Xu, "Automatic eye detection using intensity filtering and k-means clustering," *Pattern Recognition Letters*, 2010.

[82] M. Nixon, "Eye spacing measurement for facial recognition," in *Applications of Digital Image Processing*, 1985.

[83] A. Yuille, D. Cohen, and P. Hallinan, "Feature extraction from faces using deformable templates," in *Computer Society Conference on Computer Vision and Pattern Recognition*, 1989.

[84] S. A. Sirohey and A. Rosenfeld, "Eye detection in a face image using linear and nonlinear filters," *Pattern Recognition*, 2001.

[85] T. D´Orazio, M. Leo, G. Cicirelli, and A. Distante, "An algorithm for real time eye detection in face images," in *International Conference on Pattern Recognition*, 2004.

[86] Grégory Senay, Georges Linares, and Benjamin Lecouteux, "A segment-level confidence measure for spoken document retrieval," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*. IEEE, 2011, pp. 5548–5551.

[87] Grégory Senay and Georges Linares, "Confidence measure for speech indexing based on latent dirichlet allocation.," in *INTERSPEECH*, 2012.

[88] H. Jiang, "Confidence measures for speech recognition: a survey," *Speech Communication*, vol. 45, no. 4, pp. 455–470, 2005.

[89] Bertrand Servin, Simon de Givry, and Thomas Faraut, "Statistical confidence measures for genome maps: application to the validation of genome assemblies," *Bioinformatics*, vol. 26, no. 24, pp. 3035–3042, 2010.

[90] X. Hu and P. Mordohai, "A quantitative evaluation of confidence measures for stereo vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2121–2133, 2012.

[91] Sanparith Marukatat, Thierry Artières, Patrick Gallinari, and Bernadette Dorizzi, "Rejection measures for handwriting sentence recognition," in *Frontiers in Handwriting Recognition, 2002. Proceedings. Eighth International Workshop on*. IEEE, 2002, pp. 24–29.

[92] Kostas Proedrou, Ilia Nouretdinov, Volodya Vovk, and Alexander Gammerman, "Transductive confidence machines for pattern recognition," in *ECML'02*, 2002, pp. 381–390.

[93] Fayin Li and Harry Wechsler, "Open world face recognition with credibility and confidence measures," in *Audio-and Video-Based Biometric Person Authentication*. Springer, 2003, pp. 462–469.

[94] Stefan Eickeler, Mirco Jabs, and Gerhard Rigoll, "Comparison of confidence measures for face recognition.," in *FG*. 2000, pp. 257–263, IEEE Computer Society.

[95] Kieron Messer, Josef Kittler, Mohammad Sadeghi, Miroslav Hamouz, Alexey Kostin, Fabien Cardinaux, Sebastien Marcel, Samy Bengio, Conrad Sanderson, Norman Poh, et al., "Face authentication test on the banca database," in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*. IEEE, 2004, vol. 4, pp. 523–532.

[96] T. Sim, S Baker, and M. Bsat, "The cmu pose, illumination, and expression database," *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2003.

[97] K.C. Lee, J. Ho, and D. Kriegman, "Acquiring linear subspaces for face recognition under variable lighting," *IEEE Trans. Pattern Anal. Mach. Intelligence*, vol. 27, no. 5, pp. 684–698, 2005.

[98] Oliver Jesorsky, Klaus J Kirchberg, and Robert W Frischholz, "Robust face detection using the hausdorff distance," in *Audio-and video-based biometric person authentication*. Springer, 2001, pp. 90–95.

[99] Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," Tech. Rep. 07-49, University of Massachusetts, Amherst, October 2007.

[100] Lior Wolf, Tal Hassner, and Itay Maoz, "Face recognition in unconstrained videos with matched background similarity," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 529–534.

[101] L. Lenc and P. Král, "Face recognition under real-world conditions," in *International Conference on Agents and Artificial Intelligence*, Barcelona, Spain, February 14-18 2013.

[102] Jiang Chen and Jian-Yun Nie, "Automatic construction of parallel English-Chinese corpus for cross-language information retrieval," in *The sixth conference on Applied natural language processing*. 2000, Association for Computational Linguistics.

[103] J. Tomás, J. Fabregat, M. del Val, F. Casacuberta, D. Picó, A. Sanchís, and E. Vidal, "Automatic development of spanish-catalan corpora for machine translation," in *Second International Workshop on Spanish Language Processing and Language Technologies*, Jaén, Spain, 2001.

[104] P. Král, C. Cerisara, and J. Klečková, "Automatic Dialog Acts Recognition based on Sentence Structure," in *ICASSP'06*, Toulouse, France, May 2006, pp. 61–64.

[105] D.G. Lowe, "Software for sift," 2004.

[106] Jiwen Lu, Yap-Peng Tan, and Gang Wang, "Discriminative multimanifold analysis for face recognition from a single training sample per person," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 35, no. 1, pp. 39–51, 2013.

[107] E. Lleida and R. C. Rose, "Likelihood Ratio Decoding and Confidence Measures for Continuous Speech Recognition," in *ICSLP'96*, Philadelphia, USA, 1996, vol. 1, pp. 478–481.

[108] Christopher D Brown and Herbert T Davis, "Receiver operating characteristics curves and related decision measures: A tutorial," *Chemometrics and Intelligent Laboratory Systems*, vol. 80, no. 1, pp. 24–38, 2006.

[109] DMW Powers, "Evaluation: From precision, recall and f-measure to roc., informedness, markedness & correlation," *Journal of Machine Learning Technologies*, vol. 2, no. 1, pp. 37–63, 2011.

[110] Ladislav Lenc and Pavel Král, "Automatically detected feature positions for lbp based face recognition.," in *International Conference on Artificial Intelligence Applications and Innovations*, 2014.