Riga Technical University

O. Ņikišins

Effective algorithms for optical image processing and their implementation in microelectronic systems for usage in biometrics

Doctoral Thesis

2013

Riga Technical University

Faculty of Electronics and Telecommunications

Institute of Radioelectronics

Oļegs Ņikišins

Doctoral student of "Electronics" study program

Effective algorithms for optical image processing and their implementation in microelectronic systems for usage in biometrics

Doctoral Thesis

Scientific supervisor

Dr.sc.comp.

Modris Greitāns

Thesis developed in the

Institute of Electronics and Computer Science

Riga 2013

Abstract

This thesis proposes an automatic face recognition system which is based on the extensions of Local Binary Patterns transformation. The system is composed of three stages: face detection, eye localization - based face alignment, face identification. Face detection module determines the presence of the face in the input image and returns an approximate position and dimension of the subject. Detection of the face provides a rough information about the parameters of the object of interest, thus a second step namely face alignment is incorporated. At this stage dimensions of the facial region are determined more accurately based on the locations of facial features, which in our case are eye pupils. The final module is face recognition which operates in the identification mode and determines the identity of an individual out of a pool of people or rejects an identification attempt.

The contributions of the thesis are briefly summarized here: 1) a novel object (face, eye) detection principle , which is based on the combination of Local Binary Pattern histograms with simple classifiers, such as Artificial Neural Network or Support Vector Machine, 2) an accurate face identification algorithm, which is composed of various preprocessing steps, modified Multi-Scale Local Binary Pattern histograms and Weighted Nearest Neighbor Classifier (WNNC), 3) effective mini-batch discriminative feature weighting algorithm supplements the WNNC-based recognition process with statistical data about the classes, which is obtained in the learning process, 4) a fully automatic face recognition system is implemented on TMS320C6416 DSK development board.

The first part of the thesis is dedicated to the problem of frontal face detection. A novel face detector which is based on the combination of Local Binary Patterns with ANN or SVM is introduced. The advantage of this setup is the flexibility of the algorithm, which allows to adjust the trade-off between the dimensionality of the feature space and the complexity of the classifier. As the result, the performance which is comparable to state-of-the-art algorithms is obtained in low-dimensional feature space and with simple classifier.

The problem of eye localization is covered in the second part of the thesis. The above mentioned principle is utilized in the localization of eye regions in the input face image. For further gain in the localization precision the algorithm is supplemented with the second stage, namely detection of eye pupils. The experiments clearly show that the proposed method outperforms many state-of-the-art eye localization approaches.

In the third part the problem of face recognition is addressed. Our face recognition approach

is based on the combination of various preprocessing steps, modified Multi-Scale Local Binary Pattern histograms and Weighted Nearest Neighbor Classifier. The key contribution is obtained in the process of weights learning for the Weighted Nearest Neighbor Classifier. The proposed discriminative feature weighting algorithm is robust, fast, requires only *two* training examples per class and can be applied in any multi-class classification tasks.

The details about DSP-based implementation of automatic face recognition algorithm concludes the main body of the thesis.

Keywords: Face detection, eye localization, face recognition, Local Binary Patterns, Discriminative Feature Weighting, DSP-based automatic face recognition system.

Anotācija

Promocijas darba ietvaros tika izstrādātā automātiska sejas atpazīšanas sistēma, kas balstās uz transformācijas „Lokālie Binārie Tēli" (LBT) paplašinājumiem. Sistēma sastāv no trim posmiem: sejas detektēšanā, sejas centrēšana un sejas atpazīšana. Sejas detektēšana ļauj noteikt sejas esamību ieejas attēlā un atgriež sejas reģiona pozīciju un izmēru. Sejas detektors ļauj noteikt aptuvenu informāciju par objektu, tāpēc tika ieviests otrais posms – sejas centrēšana. Šīs modulis precizē sejas parametrus, balstoties uz atskaites punktiem, par kuriem šajā gadījumā tika izvēlētas acu zīlītes. Pēdējais posms ir sejas atpazīšana, kas darbojas identifikācijas režīmā un ļauj noteikt personas identitāti, salīdzinot seju ar datubāzē esošiem paraugiem, vai noraidīt identifikācijas mēģinājumu.

Darba ietvaros sasniegtie rezultāti īsumā ir sekojoši: 1) tika izstrādāts inovatīvs objektu (sejas, acu) detektors, kas balstās uz LBT histogrammu kombināciju ar vienkāršiem klasifikatoriem, tādiem kā Mākslīgais Neironu Tīkls (MNT) un Atbalsta Vektoru Mašīnas (AVM); 2) precīzs sejas identifikācijas algoritms, kas sastāv no vairākiem priekšapstrādes posmiem, modificētām Dažāda Mēroga LBT histogrammām un Svērta Tuvāko Kaimiņu Klasifikatora (STKK); 3) efektīvs parametru svēršanas algoritms, kas papildina STKK klasifikatoru ar apmācības procesā iegūtu statistisko informāciju par klasēm; 4) pilnībā automātiska sejas atpazīšanas sistēma ir realizēta TMS320C6416 signālprocesorā.

Promocijas darba pirmā nodaļa ir veltīta sejas detektēšanas problēmai. Tika piedāvāts inovatīvs sejas detektors, kas balstās uz LBT kombināciju ar MNT vai AVM klasifikatoriem. Šīs kombinācijas priekšrocība ir elastīgums, kas ļauj rast kompromisu starp parametru telpas dimensionalitāti un klasifikatora sarežģītību. Rezultātā tiek sasniegta algoritma precizitāte, kas ir salīdzināma ar jaunākās paaudzes sejas detektoriem, saglabājot zemu parametru telpas dimensionalitāti un pielietojot vienkāršas klasifikatoru struktūras.

Promocijas darba otrajā nodaļā tiek aprakstīts sejas centrēšanas uzdevums. Iepriekšminētais princips tika paplašināts arī acu detektēšanas uzdevuma pildīšanai, kas ir pirmais sejas centrēšanas algoritma modulis. Otrais modulis ir acu zīlīšu detektors, kas ir domāts augstas precizitātes centrēšanas iegūšanai. Aprakstītie eksperimenti un rezultāti skaidri parāda, ka izstrādātais algoritms precizitātes ziņā pārspēj daudzus modernus acu detektorus.

Trešā nodaļa ir veltīta sejas atpazīšanas problēmai. Piedāvātais sejas atpazīšanas algoritms sastāv no vairākiem priekšapstrādes posmiem, modificētiem Dažāda Mēroga LBT histogrammām un Svērta Tuvāko Kaimiņu Klasifikatora. Kā galveno ieguldījumu atpazīšanas procesa

uzlabošanā būtu jāpiemin STKK klasifikatoram izstrādāto svaru piemeklēšanas algoritmu. Izstrādātais parametru svēršanas algoritms ir stabils, ātrs, tas izmanto tikai divus apmācības piemērus katrai klasei un var tikt izmantots jebkāda veida multi-klašu klasifikācijas uzdevumiem.

Promocijas darbu noslēdz informācija par DSP – balstītu automātiskas sejas atpazīšanas sistēmas realizāciju.

Atslēgvārdi: sejas detektēšana, acu lokalizācija, sejas atpazīšana, Lokālie Binārie Tēli, parametru svēršana, DSP – balstīta automātiska sejas atpazīšanas sistēma

Acknowledgement

The research presented in this thesis has been developed in the Institute of Electronics and Computer Science (Riga, Latvia), under the supervision of Dr.sc.comp. Modris Greitāns. I would like to thank him for the guidance, availability and enthusiasm during the work progress. Additional thanks to the academic staff of Riga Technical University (Faculty of Electronics and Telecommunications) for the obtained theoretical knowledge within the framework of doctoral study program. Special thanks to Profesor Dr. habil. sc. ing. Jānis Jankovskis for guidance and advices about doctoral study process.

This work has been supported by a funding from the following projects:

- Nr. 2DP/2.1.1.1.0/APIA/VIAA/098 - Multimodal biometric technology for safe and easy person authentication - managed by Dr.sc.comp. Modris Greitans,
- Nr. 2009/0219/1DP/1.1.1.2.0/09/APIA/VIAA/020 - R & D Center for Smart Sensors and Networked Embedded Systems - managed by Dr.sc.comp. Leo Selavo (2010-2012),
- State Research Program: Scientific Foundations of Information Technology,
- State Research Program: Development of innovative multi-functional material, signal processing and information technologies for competitive and research intensive products,
- Nr. 2004/0002/VPD1/ESF/PIAA/04/NP/3.2.3.1/0001/0002/0007 - Support for the development of doctoral studies at Riga Technical University.

ESF
EIROPAS SOCIĀLAIS FONDS

ERAF
EIROPAS REĢIONĀLĀS ATTĪSTĪBAS FONDS

Contents

List of Figures

List of Tables

List of Abbreviations

2D Two-dimensional

ANN Artificial Neural Network

CV Cross Validation

DFW Discriminative Features Weighting

DSP Digital Signal Processor

EBW Empirical Block-level Weighting

ECDF Empirical Cumulative Distribution Function

EER Equal Error Rate

EFW Empirical Feature-level Weighting

IBW Iterative Block-level Weighting

IFW Iterative Feature-level Weighting

LBP Local Binary Patterns

LDA Linear Discriminant Analysis

MB-LBP Multi-scale Block Local Binary Patterns

MF Mean Filter

MSLBP Multi - scale Local Binary Patterns

NNC Nearest Neighbor Classifier

PCA Principal Component Analysis

RBF Radial Basis Function

ROI Region of Interest

SV Support Vectors

SVM Support Vector Machines

WNNC Weighted Nearest Neighbor Classifier

List of Mathematical Symbols

$\boldsymbol{I}_L$ - LBP image / labeled image
P - number of sampling points in LBP label
R - radius of LBP label
$\boldsymbol{X}$ - matrix notation
$\boldsymbol{x}$ - vector notation
x - element of a vector
$\boldsymbol{r}$ - vector of radii of MSLBP operator
n_R - number of radii utilized in the MSLBP operator
$\boldsymbol{h}$ - histogram of the image
$f(x)$ - function notation
K - numbers of columns / rows in the LBP regioning grid
N - number of elements in the feature vector / dimensionality of the feature space
M - number of training examples
$\mathbb{R}^{M \times N}$ - space of real numbers of dimensionality $M \times N$
$\equiv$ - assignment symbol
μ - mean value
$\boldsymbol{\Sigma}$ -covariance matrix
svd - singular value decomposition
N_e - number of eigenvectors
Var - value of the variance
O - big O notation
$g(x)$ - sigmoid function
h_Θ - hypothesis parametrized by Θ
$\boldsymbol{x}^\mathsf{T}$- transpose operation
$a_i^{(j)}$ - activation of neuron i in layer j
s_j - number of neurons in layer j *without* a bias unit
L - total number of layers in ANN
$(\boldsymbol{x}^{(i)}, y^{(i)})$ - i^{th} training example, where $\boldsymbol{x}^{(i)}$ is a pattern and $y^{(i)}$ is corresponding label
k - is the number of classes
J - cost function
λ - regularization parameter

η - learning rate
$\delta_i^{(l)}$ - the error of node i in layer l of an ANN
$f'(x)$ - derivative of the function $f(x)$
$\bullet$ - element-wise product operator / Hadamard product
$W_{i,j}^{(l)}$ - the weight of an ANN associated with the connection between unit j in layer l and unit i in layer $l+1$
$\|\boldsymbol{w}\|$ - L2 norm of a vector
$k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ - kernel function in SVM classifier
$\boldsymbol{x}_i \cdot \boldsymbol{x}_j$ - dot product of two vectors
γ - parameter of RBF kernel in non-linear SVM
$\boldsymbol{h}^f$ - LBP histogram of the face pattern
$\boldsymbol{h}^{nf}$ - LBP histogram of non-face pattern
$\boldsymbol{s}$ - vector of sizes of the sliding window
Δ_s - horizontal / vertical step of the sliding window
s_{face} - size of the square face image
N^c - number of centroids
N^{SV} - number of support vectors
d_{eye} - interocular distance
$\boldsymbol{h}^e$ - LBP histogram of the eye pattern
$\boldsymbol{h}^n e$ - LBP histogram of non-eye pattern
s_{eye} - size of the square eye image
a - size of the squared cell in the LBP regioning grid
L_{MSLBP} - size of the MSLBP label

Chapter 1

INTRODUCTION

The term *biometrics* refers to the recognition of humans by their physiological or behavioral characteristics or traits. In physiological biometric the individuals are identified by face, finger prints, palm geometry, DNA, voice and other parameters. Behavioral biometrics are related to the behavior of a person, for example typing rhythm or gait. Compared to other biometrics *face recognition* is considered to be more natural, non-intrusive, user-friendly due to its non destructive essence and can be used without the cooperation of the subject, thus the scope of this research is limited to the task of automatic face recognition. The first automatic face recognition system was introduced by Takeo Kanade in 1973 [54] and has contributed to an increasing attention to this scientific field. Due to increasing computational power of modern computers and successes in pattern recognition, face recognition systems can now operate in real time with high performance under controlled imaging conditions. That results in a wide range of real life applications, such as access control systems, border control, forensics, banking sector, human computer interaction, patient monitoring, image database investigations, video indexing and others.

Face recognition setup can operate in two modes: *verification* (or authentication) and *identification* [5]. The verification system attempts to confirm the claimed identity of a person by comparing the captured features with his/her own templates stored in the system. On the other hand, a face identification system attempts to determine the identity of an individual out of a pool of people. The scope of this research is limited to *identification* task, which in general is more complicated than the verification problem. The identification problem can further be divided in two sub-directions: open set (sometimes open universe) identification and closed set (sometimes closed universe) identification. Open set model occurs when the identity of the person may not be in the database and the opposite assumption is true for the closed set identification principle.

While verification and identification often share the same algorithmic basics in the feature extraction and classification modules, both have different applications. The primary application of verification mode is access control. To date, the access has mostly been controlled by knowledge-based or token-based security principles, such as passwords, PIN codes or ID cards.

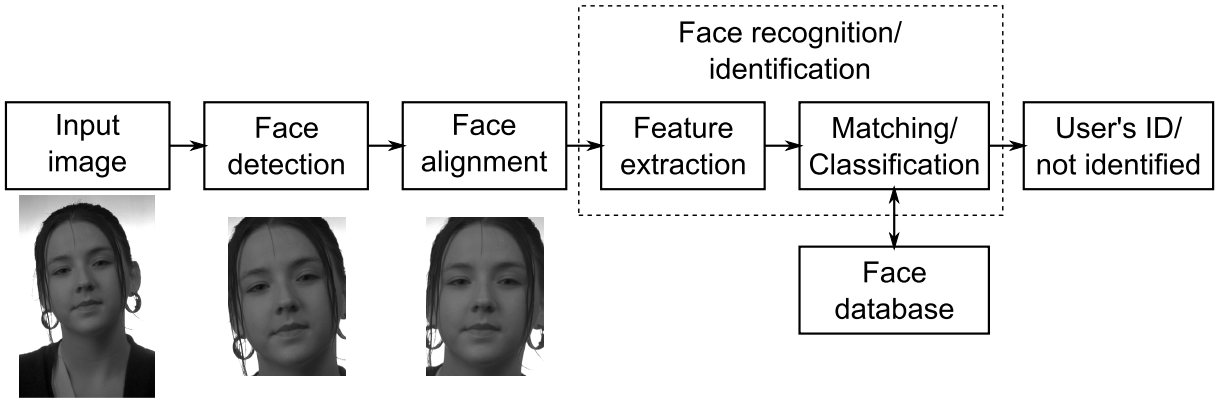


Figure 1.1: The block diagram of automatic face identification system

Face verification has a number of advantages over traditional access systems: the biometric data can not be lost, forgotten, transmitted to other individuals and it is difficult to steal. The main applications of identification mode are video surveillance, investigation of image databases, forensics, monitoring systems and human computer interaction.

1.1 Automatic face identification system

An automatic face identification system is based on three main modules (Figure 1.1): face detection, face alignment and face recognition / identification [85] (Nikisins et al.). Face detection module determines the presence of faces in the input image (or video sequence) and returns their positions and dimensions. The system in Figure 1.1 is often designed to process a single user at a time. In this scenario the term face localization is employed. The scope of this research is limited to the task of face localization, however it can easily be extended for detection problems. Localization of the face provides a rough information about the parameters of the detectable object, thus a second step namely face alignment is usually required. Face alignment allows to define facial parameters more accurately based on the locations of such facial features as eyes, nose, mouth and chin. This information is then utilized in geometrical normalization of the face region. It is shown in literature that face alignment has a large impact on the recognition accuracy [74], [93], [22]. Most of the high performance face recognition techniques assume that face has been localized perfectly, thus face detection and alignment are important research areas in the field of computer vision.

The face identification module consists of two main blocks: feature extraction and matching or classification. The design of feature extraction block is a field of extensive research, where a plethora of both empirical and analytically well-founded techniques have been developed. Ideal features should have high discriminative power to segregate different persons and should be stable to intra-class variability. Intra-class deviations may be caused by unstable illumination, expression changes, off-plane rotations of the face or partial occlusions. Another important

aspects in real life applications are dimensionality of the feature space and the complexity of the feature computation process. The comparison of the face representation to face models that are stored in the database is performed after feature extraction. This stage is called classification or matching. At this step either the identity of the person is determined or an identification attempt is rejected. Ideal classifier should take into consideration the statistical information about the addressed problem and generalize well on previously unseen data. These aspects are often challenging in biometric applications due to small amount of available intra-class data and significant number of classes in the database.

1.2 Challenges

Each block of automatic face recognition system is a field of extensive research and a plethora of various algorithms are designed for each step during last decades [124], [111], [126], [22], [49], [108]. The first two stages can be viewed as a special cases of the object detection task. The detectable object in the first step is face, while the second stage is often based on the detection of the eyes as a reference points for face alignment. Face and eye detection tasks have both common and unique issues to be resolved. Both objects have a large variance in their appearance, due to multiple factors such as skin color and texture, head pose and shape, facial expressions, lighting conditions, partial occlusions (glasses) and other features (hairstyle, beard, make-up). However eye detection task can be considered as more complicated problem due to reduced resolution of the eye image if compared to the resolution of the input facial image. This aspect degrades the amount of available statistical data about the detectable object and yields the degraded performance of the detector. Additionally, the presence of glasses in the face may significantly distort the eye region.

The above issues are also true for the face recognition stage: large variance in face appearance affects face recognition significantly. It has been observed that “the variations between the images of the same face due to illumination and viewing direction are almost always larger than image variations due to change in face identity” [8]. Illumination aspect makes the face recognition task really complicated and requires either to control the lighting conditions or to apply special techniques to compensate the negative impact of the illumination. To compensate the variance in viewing direction the cooperation from the user is needed or the geometrical transformation of the input facial image, which is usually based on 3D face model fitting. In contrast to detection tasks, another issue in face recognition comes from the lack of training images for each person. Small amount of training data does not cover possible intra-class variations and degrades the usability of advanced classifiers such as ANN or SVM. The inability to build reliable models of each individual is called the generalization problem. The recognition performance is also highly dependent from the precision of the localization stage.

Nowadays embedded face recognition systems are gaining increasing attention due to their portability, low power consumption and cost. However embedded solutions places additional

requirements on the algorithmic basics of the system. The computational complexity and dimensionality of the feature space become critical issue for all stages of automatic face recognition algorithm, which is determined by limited computational power and memory of embedded solutions.

1.3 Scope and contributions

The central idea of this research is to build a fully automatic face recognition system. The main requirement for the system is high recognition precision under semi-controlled lighting conditions, however the computational speed and the simplicity of the algorithms are also important aspects especially when dealing with embedded solutions. Most of the research in this work has been done in the fields of face detection, face alignment which is based on the detection of eye pupils and face recognition. Plethora of various methods are developed to resolve these issues individually (these approaches are briefly discussed in the next sections). However most of the papers on face detection do not consider the final application of the detector while the researchers in the filed of face recognition assume a perfect localization of the face. In this work the problem of automatic face recognition is considered as a unified task. Additionally, the automatic face recognition process is merged by similar algorithmic principles which are based on Local Binary Patterns [88] and their extensions [23]. Many researchers proved the discriminative power and computational simplicity of the LBP in various computer vision fields, which distinguishes them as an attractive features for object description. The unified approach to the construction of the algorithm also reduces the functional complexity of the software, which is important aspect in embedded systems.

The main contributions of this research are briefly summarized here:

- *Frontal face localization* [83] (Nikisins et al.), [85] (Nikisins et al.). A novel face detection principle, which is based on the combination of Local Binary Pattern histograms with simple classifiers, namely Artificial Neural Network or Support Vector Machine, is proposed in this research. The advantage of this setup is the flexibility of the algorithm, which allows to adjust the trade-off between the dimensionality of the feature space and the complexity of the classifier. As the result, the performance which is comparable to state-of-the-art algorithms (Table 3.1 in Chapter 3) is obtained in low-dimensional feature space (several hundreds of features) and with simple classifier (Artificial Neural Network with 10 Neurons in the hidden layer or Support Vector Machine (SVM) with 100-200 Support Vectors). Another advantage is the absence of the down-sampling stage, which is often incorporated in the detection algorithms in order to localize object of various scales. The scope of the experiments with the proposed method is limited to the task of frontal face localization in images taken under semi-controlled lighting conditions. However these limitation can be overcome by incorporating more diversity in the training data.

Instead of localization, the detection task can also be performed if thresholding of the output value of the classifier is added to the system. All classified sub-windows of the input image which exceed the threshold value are considered to be faces.

- *Eye localization* [83] (Nikisins et al.), [85] (Nikisins et al.). The above mentioned principle can also be applied to detection of other objects. Same algorithm is utilized for the localization of eye regions in the input face image. For further gain in the localization precision the algorithm is supplemented with the second stage, namely detection of eye pupils. The performance of the whole system is highly dependent on the localization accuracy, thus the second stage is needed. The experiments clearly show that the proposed method outperforms many state-of-the-art eye localization approaches (Table 4.3 in Chapter 4). Similar to face detection, the scope of the experiments is limited to the task of eye localization in frontal face images taken under semi-controlled lighting conditions. Partial occlusions are presented in the test images in the form of glasses.

- *Effective histogram based sliding window* [83] (Nikisins et al.). Proposed algorithms for object detection utilize the histogram-based sliding window principle. Moreover, the spatially enhanced histograms must be calculated at each scanning position. This is the most time-consuming operation in the introduced detection approaches. In order to resolve this issue an effective algorithm for recalculation of spatially enhanced LBP histograms at each scanning position of the sliding window is introduced in this research. The algorithm is optimized for a single pixel step of the sliding window.

- *Face recognition* [84] (Nikisins et al.), [85] (Nikisins et al.). A novel face recognition approach is introduced in this research. It is based on the combination of various preprocessing steps, modified Multi-Scale Local Binary Pattern histograms [24] and Weighted Nearest Neighbor Classifier. The algorithm shows an equivalent or even improved performance compared to state-of-the-art face recognition techniques.

- *Discriminative feature weighting* [84] (Nikisins et al.). Identification approaches are usually based on various Nearest Neighbor Classifiers. The Discriminative Feature Weighting (DFW) algorithm is developed in this research in order to compensate the statistical incompleteness of Nearest Neighbor Classifier by utilizing the information from all classes. The information obtained in the process of weights learning is incorporated in the recognition process by the use of Weighted Nearest Neighbor Classifier (WNNC). The DFW principles are utilized in two levels: block-level and feature-level weighting [84] (Nikisins et al.). In contrast to other weighting approaches [31] and [110], proposed methodology requires only *two* training examples per class. An algorithm also incorporates special procedure of learning data selection which makes it stable, predictable and provides better recognition results. The reduction of the learning time is another challenging aspect. This issue is very important in the cases of massive training data sets and highly dimensional

feature vectors. Both of these aspects are usually true for biometric applications. This problem is resolved by the introduction of mini-batch principle, which accelerates the proposed training methodology.

- *Demonstrator of embedded face recognition system*. A fully automatic face recognition algorithm is implemented on TMS320C6416 DSK development board that contains a TMS320C6416 fixed-point digital signal processor operating at 600 MHz and an external non-volatile Flash memory of size 512 Kbytes. The algorithmic base of the system is similar to the one described in [85] (Nikisins et al.), but a few simplifications are introduced in order to speedup the system. Proposed LBP and NNC based automatic face recognition algorithm is feasible in embedded systems and requires less than 2.3×10^9 CPU cycles to process a single 0.3 Mpixel image.

The main results of the thesis are published in the following scientific papers:

1. O. Nikisins and M. Greitans. Local binary patterns and neural network based technique for robust face detection and localization. *Proceedings of the Special Interest Group on Biometrics and Electronic Signatures (BIOSIG 2012)*, pages 147--158, September 2012

2. O. Nikisins and M. Greitans. A mini-batch discriminative feature weighting algorithm for lbp - based face recognition. *Peoceedings of IEEE International Conference on Imaging Systems and Techniques (IST 2012)*, pages 170--175, July 2012

3. O. Nikisins and M. Greitans. Reduced complexity automatic face recognition algorithm based on local binary patterns. *Peoceedings of 19th International Conference on Systems, Signals and Image Processing (IWSSIP 2012)*, pages 447--450, April 2012

4. Olegs Nikisins, Modris Greitans, Rihards Fuksis, Mihails Pudzs, and Zanda Serzane. Increasing the reliability of biometric verification by using 3d face information and palm vein patterns. In Arslan Brömme and Christoph Busch, editors, *BIOSIG*, volume 164 of *LNI*, pages 133--138. GI, 2010

5. O. Nikisins, R. Fuksis, M. Greitans, and M. Pudzs. Infrared imaging system for analysis of blood vessel structure. *Elektronika ir Elektrotechnika*, (1):45--48, 2010

1.4 Organization of the Thesis

The main body of the thesis is organized in 5 chapters as follows:

Chapter 2 introduces the theoretical preliminaries of the thesis. The basics of Local Binary Patterns and their extension namely Multi-scale Local Binary Patterns are discussed in the first part of the chapter. These operators underlie the feature extraction modules in all blocks of automatic face recognition system. The dimensionality of the feature space affects the computational

speed of the algorithms and memory requirements to store the data, thus the techniques for dimensionality reduction are also covered in Chapter 2. A brief overview of popular classifiers concludes the chapter.

Chapter 3 is dedicated to the problem of frontal face detection in digital still images. The main face detection approaches, including previous LBP-based detection techniques, are reviewed and a novel cluster of LBP-based face detection techniques is introduced. Significant attention is given to a unified principle of performance evaluation. The description of experimental setup and overall analysis of the performance of the detector is given in the conclusion.

Chapter 4 covers a second stage of automatic face recognition process, namely eye localization. The prior work in the field of eye localization, including previous LBP-based eye localization techniques, are briefly discussed at the beginning. A novel eye localization approach is introduced next. It consists of two main blocks: localization of eye regions and detection of eye pupils in the eye images. The first block is an extension of previously described face detection technique to another task. The second atypical stage complements the algorithm in order to raise the localization precision. The evaluation methodology, experimental setup and performance analysis concludes the chapter.

Chapter 5 addresses the problem of face recognition. Significant papers in the field of face recognition, including LBP-based techniques, are observed first. Proposed LBP and MSLBP-based face recognition approaches are described in details next. Significant attention is given to a novel mini-batch discriminative feature weighting algorithm both in feature and block levels. The mathematical and visual interpretation of the algorithm is presented. Experimental evaluation of the proposed face recognition methodologies is provided at the end of the chapter.

Chapter 6 describes the embedded implementation of automatic face recognition system in the TMS320C6416 DSP. The algorithmic part of the system is proposed first. The local face database is collected in order to test the system and analyze the performance of the algorithms. The structure of the system, implementational details and timing analysis are covered in the conclusion of the chapter.

Chapter 2

THEORETICAL PRELIMINARIES

2.1 Feature Extraction

One of the most fundamental tasks in computer vision and pattern recognition is feature extraction [34]. The input signal of any computer vision system is usually a digital image or a video stream, which in turn is the set of pixels. Pixel value can potentially be considered as the simplest feature for object description, however such feature selection is very inefficient due to low robustness of the descriptor to various transformations and possibly very high dimensionality of the feature space. Therefore the design of feature extraction block is a field of extensive research, where a plethora of both hand-crafted and analytically well-founded techniques have been developed. Ideal features should have high discriminative power to segregate different objects and should be stable to intra-class variability. Intra-class deviations may be caused by unstable illumination, appearance variations, off-plane rotations of the object, partial occlusions and other aspects. Another important aspects in real life applications are the dimensionality of the feature space and the complexity of the feature computation process.

2.1.1 Local Binary Patterns

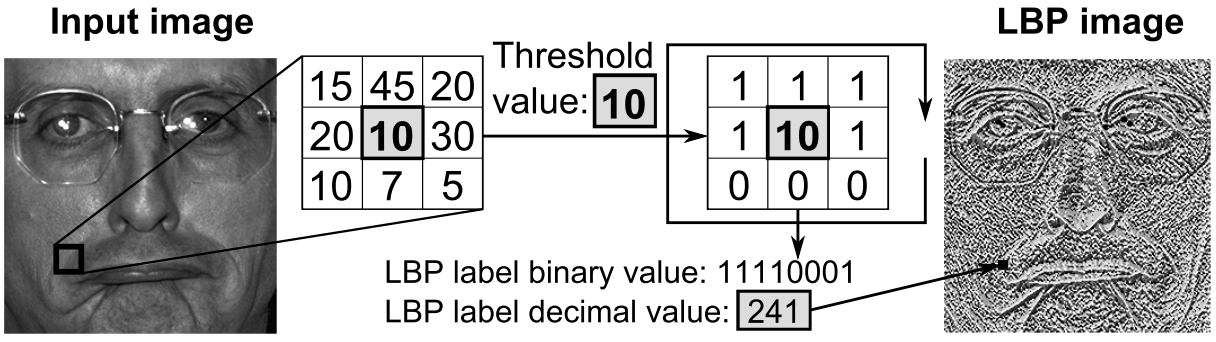


Figure 2.1: An example of labelling process of 3×3 neighborhood, $(P = 8, R = 1)$ (semantically similar to [9])

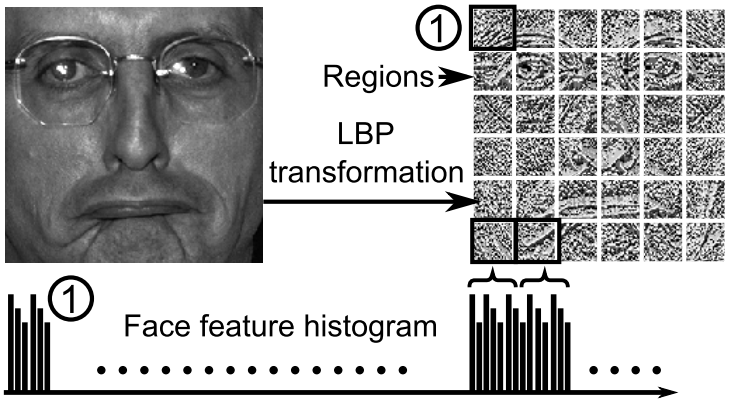


Figure 2.2: The process of spatial LBP histogram calculation, the regioning grid is 6×6

LBP operator for the first time was introduced in [88] as a texture descriptor. In the input image each pixel is labelled by thresholding its 3×3 - neighborhood with the center value and representing the result as a binary number. An example of the labeling procedure for 3×3 region of an input image is illustrated in Figure 2.1. The histogram of labels is used as the descriptor of the image. Later extensions of LBP operator [89] use neighborhoods of different sizes.

The notation (P, R) is usually used for neighborhoods description [89] , where P is the number of sampling points on a circle of radius R. A histogram $\boldsymbol{h}$ of the labelled image $\boldsymbol{I}_L(x, y)$ can be calculated:

$$h_i = \sum_{x,y} f(\boldsymbol{I}_L(x, y) = i - 1), i = 1, \ldots, n, \tag{2.1}$$

where $n = 2^P$ is the number of different labels and

$$f(A) = \begin{cases} 1, & A \text{ is true} \\ 0, & A \text{ is false.} \end{cases} \tag{2.2}$$

Histogram in the Equation 2.1 effectively represents the distribution of the gray-scale values in the digital input image [104], but the spatial information about the object is lost. In order to save spatial information about the object the division of the LBP transformed image into small regions $R_1, R_2, \ldots, R_m$ is required, where m is the number of regions. The regioning process has a lot of possible variations: regions could be of arbitrary shapes and dimensions, in different locations and with mutual overlapping. In order to simplify the regioning procedure the devision of the LBP image into $K \times K$ regions is selected, where K is the number of columns and rows in the regioning grid, in Figure 2.2 $K = 6$.

A spatially enhanced histogram is calculated by the substitution of region histograms into a single feature histogram:

$$h_{i+n\cdot(j-1)} = \sum_{x,y} f(\boldsymbol{I}_L(x, y) = i - 1) \cdot f((x, y) \in R_j), i = 1, \ldots, n, j = 1, \ldots, m. \tag{2.3}$$

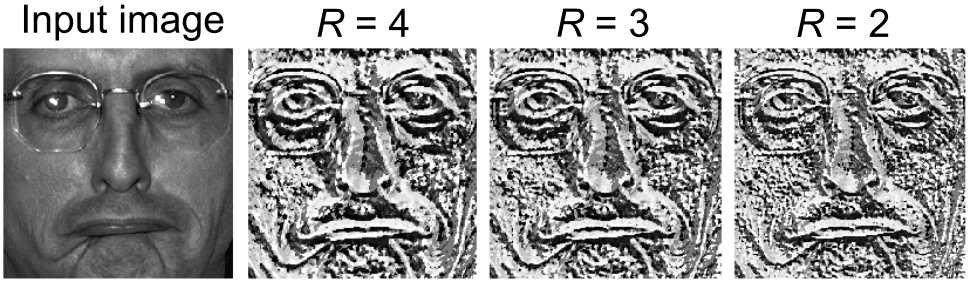


Figure 2.3: An example of LBP transformation results for different R values

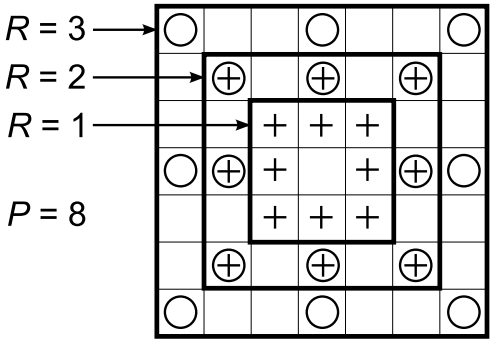


Figure 2.4: Principle of MSLBP labeling process of the 7×7 neighborhood, $(P = 8, \boldsymbol{r} = (1, 2, 3))$

The process of spatial LBP histogram calculation is schematically displayed in Figure 2.2. The vector $\boldsymbol{h}$ is now effectively represents both local and global features of the object.

In real life systems the scales of captured objects are different. The normalization of the spatially enhanced histogram $\boldsymbol{h}$ is needed before the classification step in order to get a coherent description:

$$h_i \equiv h_i / \sum_{j=1}^{N} h_j, i = 1, \ldots, N, \tag{2.4}$$

where N is the number of the elements in the vector $\boldsymbol{h}$. In general N is equal to $N = 2^P \cdot K^2$.

2.1.2 Multi-scale Local Binary Patterns

A lot of extensions of the LBP paradigm were developed since it was introduced. One of them is Multi-scale Local Binary Patterns (MSLBP), which was presented in [23] and is based on a very simple principle of varying the radius R of the LBP label and combining the resulting histograms. The examples of the output LBP images for different R values are displayed in Figure 2.3. Small values of R extract local features of the object / face, while the increased values of R distinguish the global features of the face.

The neighborhood of the MSLBP operator is described with the following parameters $(P, \boldsymbol{r} = (R_1, R_2, \ldots, R_{n_R}))$, where n_R is the number of radii utilized in the process of MSLBP calculation, Figure 2.4. Each pixel in MSLBP image is described with n_R values.

A spatially enhanced LBP histograms for different values of $\boldsymbol{r} = (R_1, R_2, \ldots, R_{n_R})$ could

be determined according to the Equation (2.3): $\boldsymbol{h}^{(1)}, \boldsymbol{h}^{(2)}, \ldots, \boldsymbol{h}^{(n_R)}$. All the histograms in the set $\boldsymbol{h}^{(1)}, \boldsymbol{h}^{(2)}, \ldots, \boldsymbol{h}^{(n_R)}$ describe the same object, but the normalization procedure is still needed before further processing, because the sums of elements in the histograms depend on the values of R:

$$h_i^{(k)} \equiv h_i^{(k)} / \sum_{j=1}^{N} h_j^{(k)}, i = 1, \ldots, N, k = 1, \ldots, n_R. \tag{2.5}$$

Various approaches for the calculation of the resulting MSLBP histogram are possible. One of them is based on the sequential substitution of the histograms $\boldsymbol{h}^{(1)}, \boldsymbol{h}^{(2)}, \ldots, \boldsymbol{h}^{(n_R)}$ into a single feature vector [23], but this leads to a significant increase of the feature space dimensionality. The number of the elements in the resulting MSLBP vector in this case is $N = 2^P \cdot K^2 \cdot n_R$ and the information in the feature vector is highly redundant.

Another approach of the MSLBP histogram calculation is based on the summation of the vectors:

$$\boldsymbol{h} = \sum_{i=1}^{n_R} \boldsymbol{h}^{(i)}. \tag{2.6}$$

2.2 Dimensionality Reduction

2.2.1 Dimensionality of LBP feature space

The parameters of the LBP operator and regioning grid directly affect on the dimensionality of the feature space. In general the dimensionality N of the spatially enhanced LBP (or MSLBP) histogram is defined as follows:

$$N = 2^P \cdot K^2, \tag{2.7}$$

where P - the number of sampling points in the LBP operator;

K - the number of the columns and rows in the LBP regioning grid.

In the task of face recognition the parameters P and K usually meet the following conditions [9], [84]: $P \geq 8$ and $K \geq 6$. The corresponding dimensionality of the feature space according to the Equation (2.7): $N \geq 9216$.

One of the approaches for the reduction of the feature space dimensionality is based on the observation, that not all features in the LBP histograms are equally important in the recognition process. This extension is called *uniform patterns* [9]. A Local Binary Pattern is called uniform if it has no more than two bitwise transitions from 0 to 1 or vice versa when the binary string is considered circular [9]. Examples of the uniform patterns are: 00000000, 01110000, 10000011. For the value of $P = 8$ the number of uniform patterns is 59, which reduces the dimensionality of the feature space: $N = 59 \cdot K^2$. This approach is computationally effective, because it is based on the sampling of the corresponding entries in the LBP histogram, however it suffers from the lack of mathematical justification and is based on the empirical observations. Another PCA-

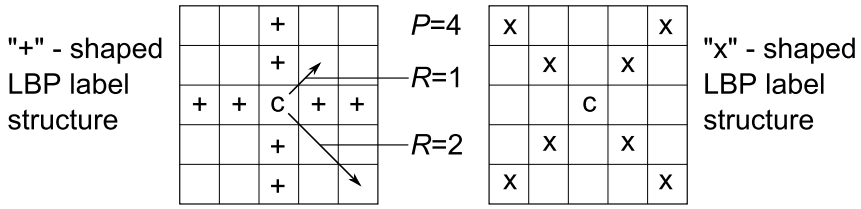


Figure 2.5: "x"-shaped and "+"-shaped LBP labels with variable radius R

based dimensionality reduction approach, which is successfully used in the face recognition task is described in subsection 2.2.2.

The scope of the LBP operator in the image processing is significantly extended since it was introduced as a texture descriptor in [88]. Another successful applications of the LBP operator are the tasks of face detection and localization [95], [85], [83]. The proposed face detection and localization techniques are based on the sliding window approach [104]. At each position of the sliding window the feature vector, which represents the image within the window, is calculated and is then classified by the previously trained classifier. The computational performance of the classification is directly dependent on the length of the feature vector and the reduction of feature space dimensionality is one of the main issues in this field.

A simple and effective technique for the reduction of the LBP histogram dimensionality with application in face detection and localization tasks is proposed in [83]. It is based on the reduced number of the sampling points in the LBP operator: $P = 4$. The resulting dimensionality of the feature space is $N = 16 \cdot K^2$. Two structures of the LBP labels with parameters $(P = 4, R)$ are introduced in [83]: "x"-shaped and "+"-shaped, see Figure 2.5 for details. The designations "x" and "+" in the grids of the Figure 2.5 denote the locations of the LBP sampling points.

2.2.2 PCA for data compression

The Principal Component Analysis (PCA) based data compression technique is successfully used in the field of face recognition and in the case of highly redundant features this compression methodology is very effective [60].

Suppose $\boldsymbol{X}$ is a set with learning data, where rows $\boldsymbol{x}$ in the matrix $\boldsymbol{X}$ stands for the training examples. In the case of LBP data compression $\boldsymbol{x}$ represents the LBP histogram. The dimensionality of the matrix is $\boldsymbol{X} \in \mathbb{R}^{M \times N}$, where M is the number of training examples.

The first step of the algorithm is the preprocessing of the input data called **mean normalization**. The mean value μ_j of the features $\boldsymbol{X}_{:,j}$ is subtracted from it:

$$\boldsymbol{X}_{:,j} \equiv \boldsymbol{X}_{:,j} - \mu_j, \text{where} \tag{2.8}$$

$$\mu_j = \frac{1}{M} \sum_{i=1}^{M} \boldsymbol{X}_{i,j}. \tag{2.9}$$

The covariance matrix of the **normalized** $\boldsymbol{X}$ is:

$$\boldsymbol{\Sigma} = \frac{1}{M} \cdot \boldsymbol{X}^{\mathsf{T}} \boldsymbol{X}. \tag{2.10}$$

The singular value decomposition [14] is utilized to compute the eigenvectors and eigenvalues of the covariance matrix $\boldsymbol{\Sigma}$:

$$[\boldsymbol{U}, \boldsymbol{S}, \boldsymbol{V}] = \mathbf{svd}(\boldsymbol{\Sigma}), \tag{2.11}$$

where the columns of matrix $\boldsymbol{U}$ are the eigenvectors of $\boldsymbol{\Sigma}$ and the diagonal elements of $\boldsymbol{S}$ are the corresponding eigenvalues.

In order to reduce the dimensionality of the input data the N_e number of vectors of $\boldsymbol{U}$ are selected:

$$\boldsymbol{U}_{reduce} = \boldsymbol{U}_{:,(1:N_e)}. \tag{2.12}$$

The projected data is now calculated as follows:

$$\boldsymbol{Z} = \boldsymbol{X} \cdot \boldsymbol{U}_{reduce}, \tag{2.13}$$

where the rows of $\boldsymbol{Z}$ are the new feature vectors with reduced dimensionality: $\boldsymbol{Z} \in \mathbb{R}^{M \times N_e}$.

The parameter N_e is determined in order to keep the required value of the variance Var in the data. For this purpose the following equation is used:

$$Var = \frac{\sum_{i=1}^{N_e} s_{i,i}}{\sum_{i=1}^{N} s_{i,i}}, \tag{2.14}$$

where N_e is the parameter to be determined with given value of Var, and $s_{i,i}$ are the diagonal elements of the matrix $\boldsymbol{S}$.

2.3 Pattern Classification

Supervised learning is one of the most fundamental tasks in machine learning. In supervised learning we have training and test examples. The training examples are usually given in the form of an ordered pair $(\boldsymbol{x}, y)$ where $\boldsymbol{x}$ is a pattern and y is a label [36]. A test example is a pattern $\boldsymbol{x}$ with unknown label. The goal of the classification is to predict labels for test examples. The term *supervised* comes from the fact that an explicit labels are provided for training examples by a teacher. The classifiers, which are mentioned below belong to the concept of supervised learning.

2.3.1 Nearest Neighbor Classifier

Nearest Neighbor Classifier (NNC) is on of the most simple and popular algorithms for the prediction of the class of a test example. This approach is also widely used in the task of face recognition and other biometric applications. In biometrics the number of classes (persons) in the database is usually significant, but the intra-class information (number of templates per person in the database) is insufficient. This fact degrades the usability of complicated classifiers, such as Artificial Neural Networks or Support Vector Machines, in the task of face recognition [84].

The learning step of the NNC is trivial: the learning pairs $(\boldsymbol{x}, y)$ are simply stored in the database. The prediction stage for the test example is based on its distance to every training example. Once the distances are determined, keep the k closest training examples, where $k \geq 1$ is a fixed integer. Then, look for the label that is the most frequently seen among these examples. This label is the predicted label for the test example.

There are only two parameters to be evaluated in the NNC: the value of k nearest training examples and the *distance function*. The value of k is usually limited by the number of training examples per class in the database and in the case of face recognition is often $k = 1$. Several most popular distance functions d for the comparison of two histograms $\boldsymbol{h}^{(1)}$ and $\boldsymbol{h}^{(2)}$ are:

Euclidean distance:

$$d(\boldsymbol{h}^{(1)}, \boldsymbol{h}^{(2)}) = \sqrt{\sum_{i=1}^{N} (h_i^{(1)} - h_i^{(2)})^2}, \tag{2.15}$$

Histogram intersection:

$$d(\boldsymbol{h}^{(1)}, \boldsymbol{h}^{(2)}) = \sum_{i=1}^{N} \min(h_i^{(1)}, h_i^{(2)}), \tag{2.16}$$

Chi square statistic:

$$d(\boldsymbol{h}^{(1)}, \boldsymbol{h}^{(2)}) = \sum_{i=1}^{N} \frac{(h_i^{(1)} - h_i^{(2)})^2}{(h_i^{(1)} + h_i^{(2)})}. \tag{2.17}$$

Weighted Nearest Neighbor Classifier (WNNC) is another popular classification algorithm [39]. WNNC in the task of LBP-based face recognition is originally introduced in [9] and later successfully supplemented in [84] where authors propose the algorithms for weights learning. WNNC approach improves the classification performance by enhancing the components more relevant to the recognition.

The idea of the WNNC could be implemented in two levels: feature-level and block-level weighting [84]. In the feature-level weighting the corresponding weights are determined for each feature in the feature vector, while in the block-level weighting the weights within the block of the feature vector are the same for all features, see Figure 2.6 for details.

The distance functions in the Equations (2.15) - (2.17) could be extended in the weighted form. The weighted histogram intersection is determined as follows:

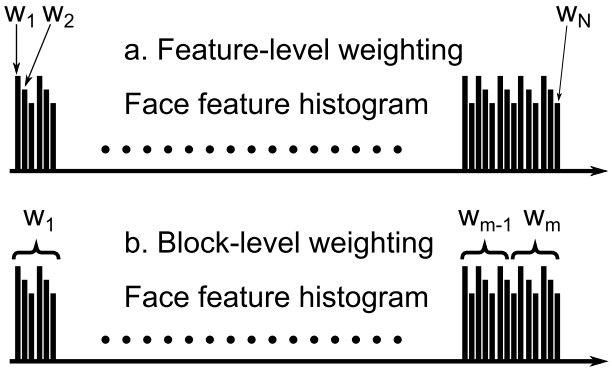


Figure 2.6: Approaches for weighting in the feature (a) and block (b) levels

Histogram intersection with feature-level weighting:

$$d^{fw}(\boldsymbol{h}^{(1)}, \boldsymbol{h}^{(2)}) = \sum_{i=1}^{N} \min(w_i h_i^{(1)}, w_i h_i^{(2)}), \tag{2.18}$$

where w_i is the weight of the feature h_i.

Histogram intersection with block-level weighting:

$$d^{bw}(\boldsymbol{h}^{(1)}, \boldsymbol{h}^{(2)}) = \sum_{j} \sum_{i} \min(w_j h_{i,j}^{(1)}, w_j h_{i,j}^{(2)}), \tag{2.19}$$

where i - feature number within the block j. In the case of spatially enhanced LBP histogram $j = 1, \ldots, K^2$ and $i = 1, \ldots, N/K^2$.

An obvious disadvantage of the NNC method is the amount of computations needed to make predictions. Suppose that N *training* examples in the space $\mathbb{R}^M$ are given. Then the recognition stage of one training example requires $O(NM)$ time, compared to just $O(M)$ time to apply a linear classifier such as a perceptron [36]. If the dimensionality of the training data is small $M \leq 20$, then the classification can be done much faster, but for high dimensional data no useful approaches are known to speed up the process [55].

2.3.2 Artificial Neural Network

An Artificial Neural Network (ANN) is another popular classification paradigm in the machine learning that is inspired by the operational aspects of biological neural networks. The complexity of real neurons is highly abstracted, but an ANN concept is successfully applied in many engineering and scientific fields [128]. An ANN consists of interconnected artificial neurons and the input information is classified by incorporating the *weights* of the respective signals. In most cases an ANN is an adaptive system, that utilize the learning information in order to adapt its structure. Neural networks are non-linear classifiers which are usually used to find the relationships between the input information and the desired output. The first model of the neural network was introduced in [78] (1943). Hundreds of different models, which are considered as

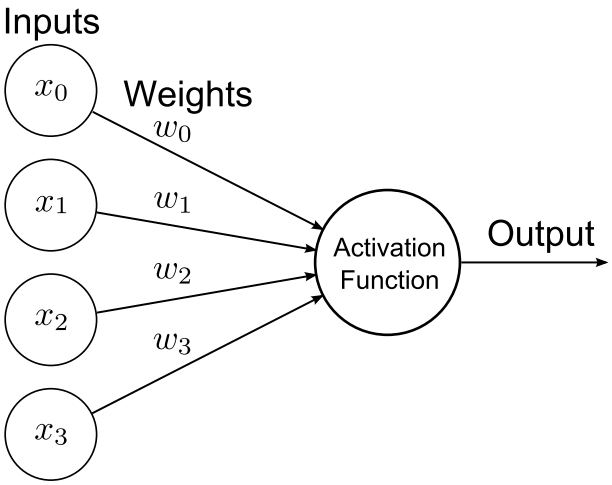


Figure 2.7: Model of an artificial neuron

ANNs have been developed since that time. The model, which is based on the back-propagation algorithms [98] is presented in the next sections, since it is one of the most popular ANN models, and many other extensions are based on it.

Model of an Artificial Neuron

The authors in [78] proposed a binary threshold unit as a model for an artificial neuron, see Figure 2.7 for details. The activation function (Figure 2.7) in [78] is a threshold. This mathematical model computes a weighted sum of its inputs x_i, $i = 0, \ldots, N$ and generates an output of 1 if the sum is above the threshold, or 0 otherwise. The input element x_0 is called a *bias unit* and is always equal to 1: $x_0 = 1$.

The model of the neuron has been later generalized in many aspects. An obvious one is to implement different activation functions into the model. Some of these functions are piecewise linear, Gaussian and sigmoid. The sigmoid function is the most frequently used in an ANN [50]. Sigmoid is a smoothed version of classical [78] activation function. A derivative of the sigmoid is defined for all input values and it can be used with gradient descent based learning methods. The standard sigmoid function is the *logistic* function, which is defined as follows:

$$g(x) = \frac{1}{1 + \exp(-\alpha x)}, \tag{2.20}$$

where parameter α determines the slope of a function.

The *hypothesis* of a single perceptron with logistic activation function, which maps the input vector $\boldsymbol{x}$ to output label y can be written:

$$h_w(\boldsymbol{x}) = g(\boldsymbol{w}^\mathsf{T}\boldsymbol{x}) = \frac{1}{1 + \exp(-\boldsymbol{w}^\mathsf{T}\boldsymbol{x})}, \tag{2.21}$$

where $\boldsymbol{w} = (w_0, \ldots, w_N)$ - is a vector of weights, see Figure 2.7.

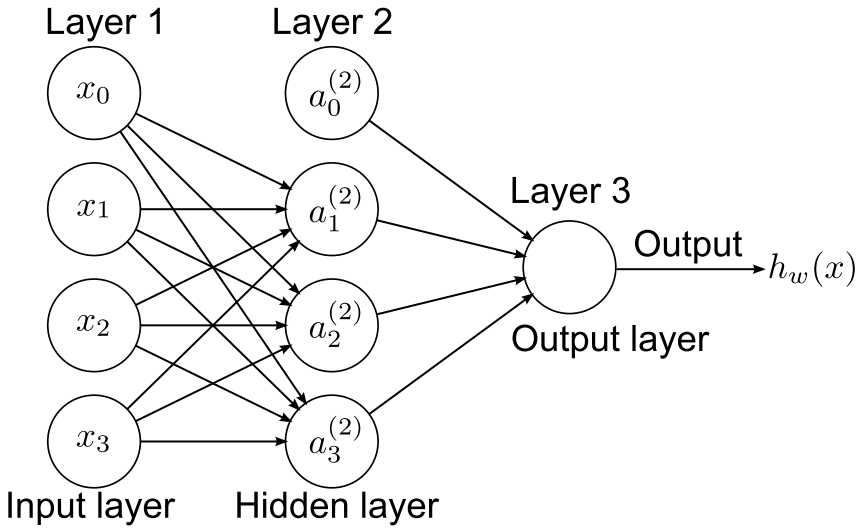


Figure 2.8: Architecture of feed-forward artificial neural network

Architecture of an Artificial Neural Network

ANNs can be represented as weighted directed graphs [50], see Figure 2.8 for details. Artificial neurons are nodes and weighted directed edges connect the outputs and inputs of neurons. The network in Figure 2.8 is a *feed-forward* network, which have no loops in connections. This structure is the most common in classification task and is discussed here in details.

In the common feed-forward networks the neurons are organized in layers, which have connections between them. The layers could be divided into three groups: *input layer*, *hidden layers* and *output layer*, Figure 2.8. The information is submitted in the network via the elements of the input layer $x_0, \ldots, x_N$. The number of hidden layers could be different, but in most cases no more than three layers are used. The dimensionality of the output layer depends on the amount of classes in the classification problem. For two classes the output layer has only one neuron. The elements x_0 and $a_0^{(2)}$ in Figure 2.8 are *bias units*.

In general, feed-forward networks are *static*, they produce only one set of output values for a given input [50]. Feed-forward networks have no *memory*, which means that their response to an input is independent of the previous state of the network.

To explain the specific computations represented by an ANN the following parameters are introduced:

- $a_i^{(j)}$ - *activation* (the value, which is computed by the neuron) of unit i in layer j,
- $\boldsymbol{W}^{(j)}$ - matrix of weights controlling function mapping from layer j to layer $j+1$.

The computations performed by an ANN in the Figure 2.8 can now be represented in *non-*

vectorized form as follows:

$$\begin{aligned} a_1^{(2)} &= g(W_{10}^{(1)}x_0 + W_{11}^{(1)}x_1 + W_{12}^{(1)}x_2 + W_{13}^{(1)}x_3), \\ a_2^{(2)} &= g(W_{20}^{(1)}x_0 + W_{21}^{(1)}x_1 + W_{22}^{(1)}x_2 + W_{23}^{(1)}x_3), \\ a_3^{(2)} &= g(W_{30}^{(1)}x_0 + W_{31}^{(1)}x_1 + W_{32}^{(1)}x_2 + W_{33}^{(1)}x_3). \end{aligned} \tag{2.22}$$

The output value of an ANN is:

$$h_w(\boldsymbol{x}) = a_1^{(3)} = g(W_{10}^{(2)}a_0^{(2)} + W_{11}^{(2)}a_1^{(2)} + W_{12}^{(2)}a_2^{(2)} + W_{13}^{(2)}a_3^{(2)}). \tag{2.23}$$

The dimensionality of the parameter matrix $\boldsymbol{W}^{(j)}$ depends on the structure of an ANN: $\boldsymbol{W}^{(j)} \in \mathbb{R}^{s_{j+1}\times(s_j+1)}$, where s_j is the number of neurons in layer j and s_{j+1} is the number of neurons in layer $j+1$.

In order to get a vectorized implementation of Equations (2.22) - (2.23) the extra parameters $\boldsymbol{z}^{(j)}$ are introduced, where the upper index represents parameters associated with layer j. Parameters in vector $\boldsymbol{z}^{(j)} = (z_1^{(j)}, \ldots, z_{s_j}^{(j)})$ are equal to weighted linear combinations, which are the arguments of sigmoid functions in Equations (2.22) - (2.23), for example:

$$\begin{aligned} z_1^{(2)} &= W_{10}^{(1)}x_0 + W_{11}^{(1)}x_1 + W_{12}^{(1)}x_2 + W_{13}^{(1)}x_3, \\ z_1^{(3)} &= W_{10}^{(2)}a_0^{(2)} + W_{11}^{(2)}a_1^{(2)} + W_{12}^{(2)}a_2^{(2)} + W_{13}^{(2)}a_3^{(2)}. \end{aligned} \tag{2.24}$$

To generalize the mathematical calculations, let's denote the input vector $\boldsymbol{x}$ as the activations in the input layer: $\boldsymbol{a^{(1)}} = \boldsymbol{x}$. The activation functions of the hidden layer (Figure 2.8) can now be determined in the *vectorized* form:

$$\begin{aligned} \boldsymbol{z}^{(2)} &= \boldsymbol{W}^{(1)}\boldsymbol{a}^{(1)}, \\ \boldsymbol{a}^{(2)} &= g(\boldsymbol{z}^{(2)}). \end{aligned} \tag{2.25}$$

Equations (2.25) determine the activations in the hidden layer $\boldsymbol{a}^{(2)} = (a_1^{(2)}, \ldots, a_{s_2}^{(2)})$. The bias unit of the hidden layer is substituted next in the vector $\boldsymbol{a}^{(2)}$:

$$\boldsymbol{a}^{(2)} \equiv (a_0^{(2)}, \boldsymbol{a}^{(2)}). \tag{2.26}$$

The dimensionality of the updated vector $\boldsymbol{a}^{(2)}$ is $\mathbb{R}^{s_2+1}$. Finally, the output value of the hypothesis of an ANN can be calculated:

$$\begin{aligned} \boldsymbol{z}^{(3)} &= \boldsymbol{W}^{(2)}\boldsymbol{a}^{(2)}, \\ h_w(\boldsymbol{x}) &= \boldsymbol{a}^{(3)} = g(\boldsymbol{z}^{(3)}). \end{aligned} \tag{2.27}$$

The described methodology could be easily extended to any number of layers in an ANN.

Process of computing the activations from input to hidden layers and then to output layer is called the *forward propagation*.

Artificial Neural Network Learning - the Back-propagation Algorithm

In the context of an ANN the ability to learn can be viewed as the problem of updating network architecture and weights of the connections so that a network can efficiently perform a specific task [50]. The network learns the weights from available training patterns and the performance of the system is improved over time by iteratively updating the weights. The ability to learn input-output relationships from a given collection of representative examples is one the major advantages of an ANN over traditional expert systems (rules are specified by human experts).

The development of the *back-propagation* learning algorithms [98] has made the *multilayer perceptron* (Figure 2.8) the most popular architecture among researches and users of neural networks. The back-propagation algorithm is a *gradient-descent* method [103], which minimizes the *cost function* of the network.

To explain the cost function of the multilayer perceptron with logistic activation functions the following notations are introduced:

- $\{(\boldsymbol{x}^{(1)}, y^{(1)}), (\boldsymbol{x}^{(2)}, y^{(2)}), \ldots, (\boldsymbol{x}^{(M)}, y^{(M)})\}$ - set of learning examples,
- L - total number of layers in the network,
- k - the number of classes.

Two classification tasks can be performed by the multilayer perceptron:

- *binary classification* - the output layer has one unit and the label y is a *number* with possible values $\{0, 1\}$,
- *multi-class classification (k classes)* - the output layer has k units and the label $\boldsymbol{y}$ is a *vector* of the dimensionality $\mathbb{R}^k$, for example $\boldsymbol{y}(k = 4) = \{0, 1, 0, 0\}$.

The cost function $J(\boldsymbol{W})$ of an ANN can be written in the following form:

$$J(\boldsymbol{W}) = -\frac{1}{M}\left(\sum_{i=1}^{M}\sum_{j=1}^{k} y_j^{(i)} \log(h_w(\boldsymbol{x}^{(i)}))_j + (1 - y_j^{(i)}) \log(1 - (h_w(\boldsymbol{x}^{(i)}))_j)\right) + \frac{\lambda}{2M}\sum_{l=1}^{L-1}\sum_{i=1}^{s_l}\sum_{j=1}^{s_l+1}(W_{ji}^{(l)})^2, \tag{2.28}$$

where $(h_w(\boldsymbol{x}))_i$ denotes the i^{th} output of an ANN in the case of multi-class classification problem $h_w(\boldsymbol{x}) \in \mathbb{R}^k$;

λ - the regularization parameter, which is used to prevent overfitting [11]. The regularization term penalizes the weights in order to reduce the complexity of the hypothesis.

The purpose of the back-propagation algorithm is to minimize the cost function in Equation (2.28) by adjusting the weights: $\min_{\boldsymbol{W}}(J(\boldsymbol{W}))$. The gradient-descent method is used for that. The following update rule is executed until the algorithm converges:

$$W_{i,j}^{(l)} \equiv W_{i,j}^{(l)} - \eta \frac{\partial J(\boldsymbol{W})}{\partial W_{i,j}^{(l)}}, \tag{2.29}$$

where parameter η is called *learning rate* and is used to control the speed of the learning process.

The Equation (2.29) can be interpreted as follows: the adjustment of each weight $W_{i,j}^{(l)}$ will be negative of a constant η multiplied by the dependence of the previous weight on the error of an ANN, which is a partial derivative of $J(\boldsymbol{W})$ in respect to $W_{i,j}^{(l)}$.

The key step of the back-propagation algorithm is to determine the partial derivatives of $J(\boldsymbol{W})$ in respect to $W_{i,j}^{(l)}$. The function $J(\boldsymbol{W})$ is a *non-convex* and the gradient descent is susceptible to local optima, however, in practice it performs well.

The first important aspect in the training process is random initialization of each parameter $W_{i,j}^{(l)}$ to a small random value near zero. If this step is not performed and all parameters are initially set to the identical values, then all hidden layers will end up learning the same function of the input (formally, $a_1^{(i)} = a_2^{(i)} = a_3^{(i)} = \ldots$, for any input $\boldsymbol{x}$). The random initialization is needed for the purpose of *symmetry breaking*.

The intuitive explanation of the back-propagation algorithm is as follows [3]. For a given training example $(\boldsymbol{x}, y)$ the *forward propagation* is first performed in order to compute all the activations in the network, including the output value of the hypothesis $h_w(\boldsymbol{x})$. Then, the error term $\delta_i^{(l)}$ should be determined for each node i in layer l, that measures the influence of the particular node onto any errors in the output. For an output node the difference between the networks activation and the true target value can be measured directly and used to define $\delta_i^{(L)}$, where L is the output layer. The computation of the $\delta_i^{(l)}$ terms for the hidden layers is more complicated and is discussed below.

For a single training example $(\boldsymbol{x}, \boldsymbol{y})$ in an ANN with L layers the error terms $\boldsymbol{\delta}^{(L)}$ for the units in the output layer are calculated as follows:

$$\boldsymbol{\delta}^{(L)} = \boldsymbol{a}^{(L)} - \boldsymbol{y}. \tag{2.30}$$

The errors for the units in the hidden layer $l = (2, \ldots, L-1)$:

$$\begin{aligned} \boldsymbol{\delta}^{(l)} &= (\boldsymbol{W}^{(l)})^{\mathsf{T}} \boldsymbol{\delta}^{(l+1)} \bullet g'(\boldsymbol{z}^{(l)}), \\ g'(\boldsymbol{z}^{(l)}) &= g(\boldsymbol{z}^{(l)}) \bullet (1 - g(\boldsymbol{z}^{(l)})), \end{aligned} \tag{2.31}$$

where the symbol $\bullet$ denotes the element-wise product operator, also called the Hadamard product.

The back-propagation for M training examples $\{(\boldsymbol{x}^{(1)}, \boldsymbol{y}^{(1)}), (\boldsymbol{x}^{(2)}, \boldsymbol{y}^{(2)}), \ldots, (\boldsymbol{x}^{(M)}, \boldsymbol{y}^{(M)})\}$

is based on the Equations (2.30) and (2.32) with some extensions. First, the terms $\Delta_{ij}^{(l)}$ are set to zero for all values of l, i, j: $\Delta_{ij}^{(l)} = 0$. These terms are used as the accumulators in the computation of partial derivatives $\frac{\partial J(\boldsymbol{W})}{\partial W_{i,j}^{(l)}}$. Next, the for-loop is executed in order to update $\Delta_{ij}^{(l)}$ terms:

For $i = 1$ to M {

1. Set $\boldsymbol{a}^{(1)} = \boldsymbol{x}^{(i)}$
2. Perform forward-propagation to compute $\boldsymbol{a}^{(l)}$ for $l = 2, 3, \ldots, L$
3. Using $\boldsymbol{y}^{(i)}$, compute $\boldsymbol{\delta}^{(L)} = \boldsymbol{a}^{(L)} - \boldsymbol{y}^{(i)}$
4. Compute $\boldsymbol{\delta}^{(L-1)}, \boldsymbol{\delta}^{(L-2)}, \ldots, \boldsymbol{\delta}^{(2)}$ (Equation 2.32)
5. $\boldsymbol{\Delta}^{(l)} \equiv \boldsymbol{\Delta}^{(l)} + \boldsymbol{\delta}^{(l+1)}(\boldsymbol{a}^{(l)})^{\mathsf{T}}$ }

The partial derivatives can be calculated based on the $\Delta_{ij}^{(l)}$ terms:

$$\begin{aligned} \frac{\partial J(\boldsymbol{W})}{\partial W_{i,j}^{(l)}} &= \frac{1}{M}\Delta_{ij}^{(l)} + \lambda W_{ij}^{(l)}, \text{ for } j \neq 0, \\ \frac{\partial J(\boldsymbol{W})}{\partial W_{i,j}^{(l)}} &= \frac{1}{M}\Delta_{ij}^{(l)}, \text{ for } j = 0. \end{aligned} \tag{2.32}$$

To train the neural network the steps of gradient descent algorithm are now repeatedly performed according to the Equation (2.29) in order to reduce the cost function $J(\boldsymbol{W})$.

2.3.3 Support Vector Machines

Support Vector Machines (SVMs) is a state-of-the-art method of supervised learning, which is successfully used in various applications. SVM classification approach is a relatively recent development in the field of pattern recognition introduced in [116] with origins from [115]. The original support vector classifier was introduced as a *linear* separator of *two* classes, however this limitations were later overcome by developing non-linear SVM classifier and multi-class classification approaches.

Modern nonlinear SVMs arises from two main ideas [58]. The first idea is the *mapping* of the feature vectors in a non-linear way to a high (possibly infinite) dimensional space, where the linear classification is utilized. For an original feature vector of the dimensionality $\boldsymbol{x} \in \mathbb{R}^N$ the transformed feature vector is given by $\Phi(\boldsymbol{x})$. The label y remains the same, so the training example $(\boldsymbol{x}_i, y_i)$ becomes $(\Phi(\boldsymbol{x}_i), y_i)$. Then the hyperplane which separates the training examples $\{(\Phi(\boldsymbol{x}_i), y_i), \ldots, (\Phi(\boldsymbol{x}_M), y_M)\}$ is determined in the transformed space. The transformed feature vector $\Phi(\boldsymbol{x}_i)$ should lie on one side of the hyperplane if the label $y_i = -1$ and $\Phi(\boldsymbol{x}_i)$ lies on the other side of the hyperplane if $y_i = 1$. This approach results in a nonlinear classifier in the original space.

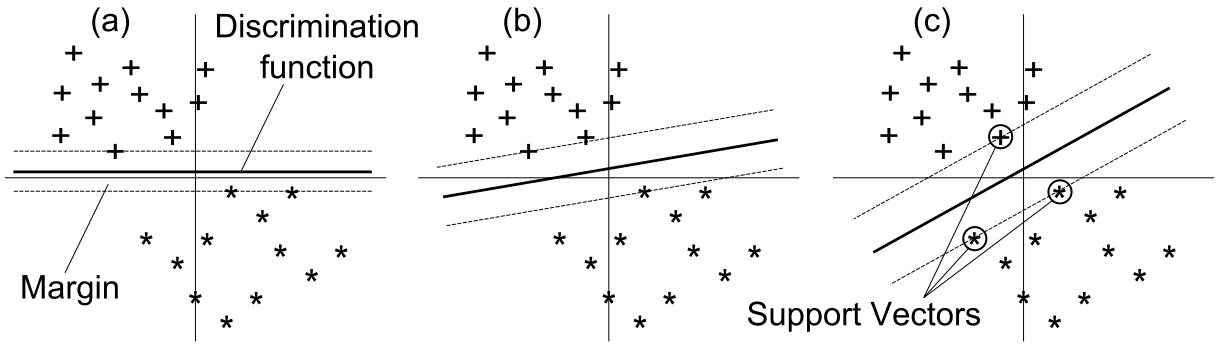


Figure 2.9: The idea of large margin classification in SVM, (a) and (b) are non-optimal solutions and (c) - optimal discrimination function

The second key idea is to find the hyperplane, which separates the data with *large margin*. In general the number of separating hyperplanes is infinite, but the one that separates the data as much as possible is selected. Many hyperplanes perform equally well on the training data (even perfect separation is possible), but the performance on a new data can very significantly. An example of the linearly separable classes with two-dimensional feature vectors is displayed in Figure 2.9. An infinite number of discrimination functions perform perfectly on the training data, for example Figure 2.9 (a) and (b), but they might have different generalization performance. An obvious challenge is to find the separating hyperplane which performs better than others in terms of their error rate on previously unseen examples. It turns out that the hyperplane with large margin provides good generalization performance, Figure 2.9 (c). The fact that the large margin classifier tends to have good generalization performance has been justified both in theory and practice [58].

Support Vector Machines for linearly separable classes

Consider the case of linearly separable data (Figure 2.9) with a set of N-dimensional feature vectors $\boldsymbol{x}$ and associated identifiers of the class $y \in \{-1, 1\}$. The linear discrimination function with parameters $\boldsymbol{w}$ is defined in the following form:

$$\boldsymbol{w}^{\mathsf{T}}\boldsymbol{x} + b = 0. \tag{2.33}$$

Two parallel hyperplanes are defined in order to maximize the margin:

$$\begin{aligned} \boldsymbol{w}^{\mathsf{T}}\boldsymbol{x} + b &= 1, \\ \boldsymbol{w}^{\mathsf{T}}\boldsymbol{x} + b &= -1. \end{aligned} \tag{2.34}$$

Planes in the Equation (2.34) are passing through the support vectors and the training patterns are not allowed between them. To guarantee that no training patterns are present between them,

the following inequality must hold for all training patterns $\boldsymbol{x}_i$:

$$y_i(\boldsymbol{w}^\mathsf{T}\boldsymbol{x}_i + b) \geq 1. \tag{2.35}$$

The distance between the hyperplanes in the Equation (2.34) is $2/\|\boldsymbol{w}\|$. The minimization of the $\|\boldsymbol{w}\|$ subject to the constraints given in the Equation (2.35) causes the maximization of the margin. This is a quadratic optimization problem with linear inequality constraints. Only the points closest to the boundary participate in defining the discrimination hyperplane. These points are called the *support vectors*. Once the optimization process is performed the discrimination hyperplane is parallel to and positioned in the middle between these two parallel hyperplanes, Figure 2.9.

The Lagrangian theory is utilized to simplify the optimization process by the reformulation of the minimization problem and avoiding the inequality constraints. However the description of the optimization is beyond the scope of this work and is described in details in the literature [21], [58].

Once the training of the SVM is performed and the parameters $\boldsymbol{w}$ and b are determined the two-class classification of a pattern $\boldsymbol{x}$ can be accomplished according to the following discrimination function:

$$f(\boldsymbol{x}) = \boldsymbol{w}^\mathsf{T}\boldsymbol{x} + b, \tag{2.36}$$

where the class $y_{\boldsymbol{x}}$ of the vector $\boldsymbol{x}$ is determined as follows:

$$\begin{aligned} y_{\boldsymbol{x}} &= +1, \text{ if } f(\boldsymbol{x}) \geq 0, \\ y_{\boldsymbol{x}} &= -1, \text{ if } f(\boldsymbol{x}) < 0. \end{aligned} \tag{2.37}$$

In terms of training vectors the discrimination function is restated in the form:

$$f(\boldsymbol{x}) = \sum_{i=1}^{M} \alpha_i y_i (\boldsymbol{w}^\mathsf{T}\boldsymbol{x}) + b, \tag{2.38}$$

where α_i are Lagrange multipliers.

For some training examples i the values of the Lagrange multipliers α_i are equal to zero: $\alpha_i = 0$. The corresponding training examples $(\boldsymbol{x}_i, y_i)$ does not affect the maximum margin hyperplane. The rest of training vector do contribute the final discrimination function. These examples have positive α_i values and are *support vectors*.

Support Vector Machines for non-linear and non-separable classification

In the most cases of real-world applications the data is not linearly separable, see Figure 2.10 for an example. To allow the training of the SVM in non-linearly separable cases the *soft margin* training principle is introduced in [29]. Soft margin training allows some training examples to

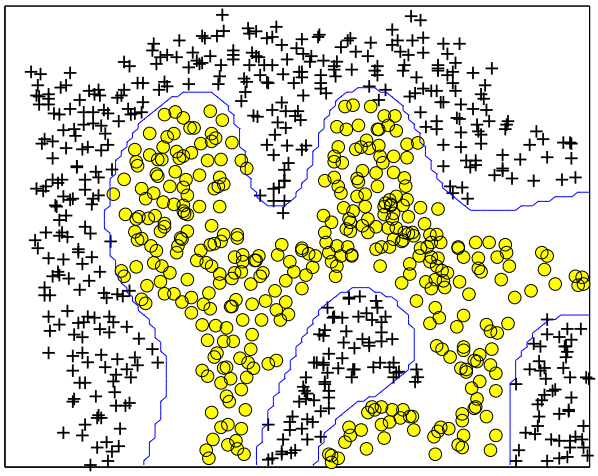


Figure 2.10: An example of non-linearly separable data in two dimensional feature space, the Gaussian RBF kernel is utilized to get the decision boundary

remain on the wrong side of the decision boundary, Figure 2.10. This principle resulted in a very wide use of the SVM both in science and industry.

The Equation (2.35) is modified to allow some training patterns to remain miss-classified, by introducing the *slack variable* $\xi_i > 0$ for each example:

$$y_i(\boldsymbol{w}^\mathsf{T}\boldsymbol{x}_i + b) \geq 1 - \xi_i. \tag{2.39}$$

The minimization of the $\|\boldsymbol{w}\|$ subject to the constraint in the Equation (2.39) can be solved using Lagrange multipliers. The optimization problem subject to the constraint (2.39) is defined as follows:

$$\min_{\boldsymbol{w},b,\xi} \frac{1}{2}\boldsymbol{w}^\mathsf{T}\boldsymbol{w} + C\sum_{i=1}^{M}\xi_i, \tag{2.40}$$

where $C > 0$ is the penalty parameter of the error term.

The *kernel trick* [10] facilitated the extension of the SVM to non-linear classification problems [18]. To determine the similarity of two feature vectors $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ in a linear space the *kernel function* $k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is determined as the dot product $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{x}_i \cdot \boldsymbol{x}_j$. The linear classifier described above uses such a kernel function. The dot product of the linear SVM can be replaced with non-linear kernel functions:

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \Phi(\boldsymbol{x}_i) \cdot \Phi(\boldsymbol{x}_j). \tag{2.41}$$

This operation performs the non-linear transformation of the feature space in which the linear discrimination hyperplane can be determined. This results a non-linear hyper-surface in the original feature space, Figure 2.10. In general, the SVM classification algorithm in the non-linear case is similar to that used for linear discrimination, but the dot products are replaced by a kernel function.

A number of simple kernels are usually used in the SVMs: d^{th} - order polynomial kernel, Equation (2.42); RBF kernel, Equation (2.43); Gaussian RBF kernel, Equation (2.44):

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\boldsymbol{x}_i \cdot \boldsymbol{x}_j)^d, \tag{2.42}$$

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp(-\gamma \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2), \text{ for } \gamma > 0, \tag{2.43}$$

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|}{2\sigma^2}\right). \tag{2.44}$$

The discriminative function of non-linear SVM is

$$f(\boldsymbol{x}) = \sum_{i=1}^{M} \alpha_i y_i k(\boldsymbol{x}_i, \boldsymbol{x}) + b. \tag{2.45}$$

As discussed in the Equation (2.37) the sign of the function $f(\boldsymbol{x})$ in the Equation (2.45) determines the class of the feature vector $\boldsymbol{x}$.

The SVM optimization problem can be solved analytically for a small number of data, however in most real-world cases the quadratic optimization problem must be solved numerically. The amount of literature about the topic of general convex optimization and SVM optimization in particular is very large, and many efficient software tools have been developed for both. A Library for Support Vector Machines (LIBSVM) is used in the simulations of this work [25].

Chapter 3

FACE DETECTION

The precise alignment of the facial region in the input image is one of the most important aspects in many applications, such as automatic face recognition, video surveillance, human computer interaction, digital photography, retail, systems for the automatic adjustment of the power consumption. The resulting performance of the above systems depends on the precision of the face detection stage. Face detection is the first module in the automatic face recognition system, which is schematically introduced in Figure 1.1. In many scenarios of the face recognition applications the user is interested in the cooperation with the system, thus the face detection module is designed to deal with frontal faces. The task of frontal face detection is the subject of this chapter.

The face detection task can be viewed as a special case of the *detection of object class*. The purpose is to find the locations and the sizes of the objects in the image, which belong to the specific class. It is possible to try to apply a recognition algorithms to every part of the image in order to understand the objects in it, but it is obviously very slow and error-prone approach. The more effective way is to construct special purpose detectors which are used for rapid detections of regions of interest (ROI), where the particular objects are most likely to occur.

Face detection is a field of intensive research with plethora of various algorithms designed during the last decades, both for video - based and image - based face detection. In video sequence the time domain is present, which is used to simplify the detection task in different manners. In digital images only the spatial and quantitative information about the scene is introduced. The scope of this research is limited to the task of face detection in digital images.

3.1 Related work

Many face detection algorithms are proposed in scientific papers in the last decades. A comprehensive survey of many algorithms is provided in [124] with more recent reviews in [111] and [126]. According to the taxonomy of [124] these methods can be divided into three main categories: *feature-based*, *template-based* and *appearance-based* face detectors.

Feature-based detectors attempt to find the locations of the local face features and then utilize the knowledge about their relative positions. Such features as eyes, nose, mouth are usually used in the feature-based face detectors. Feature-based methods are generally applied in the face *localization* (one face per image) tasks with a good quality of the input image. These methods are robust to variations in the illumination conditions, occlusions and off-plane rotations of the face, but are usually computationally expensive. A feature-based detector with SVM classier and applications both for face detection and recognition is introduced in [45] with later extensions in [46]. An example of boosting-based algorithm is proposed in [100].

Template-based approaches are robust in a wide range of pose and expression variability. One of the most widely-used techniques in the field of template based detection is active appearance models [27]. These methods typically require good initialization near the face and therefore are not suitable for fast face detection.

Appearance-based approaches perform scanning of the input image with a small overlapping windows with the purpose of searching the most likely face candidates. Most of the modern appearance based approaches rely on statistical classifiers, which are optimized using the sets of labeled face and non-face training examples. The block-diagram of the appearance-based face detection system is displayed in Figure 3.1.

The concept of *sliding window* (sometimes, scanning window) is the key idea of the appearance based methods. A sliding window scans the input image at different locations with a constant (Figure 3.1, (a)), or a variable (Figure 3.1, (b)) size of the window. Selection of the scanning concept depends on the global structure of the face detection system. Downsampling is performed if the size of extracted image is desired to be of the constant size, Figure 3.1, (a). Structure (a) is usually utilized if the pattern in the sliding window is classified directly after preprocessing, however an optional feature extraction module is sometimes incorporated. If the dimensionality of the feature space obtained in the feature extraction module is independent form the size of extracted window, then structure (b) (Figure 3.1) is preferable. Preprocessing of the subwindows is sometimes performed before the classification stage. Preprocessing might include the subtraction of linear gradient function, histogram equalization and other techniques, which are usually addressed to resolve the problem of variable lighting conditions. The classifier is the final block of the face detection system, which rates the input pattern as either *face* or *nonface*. The appearance-based methods mainly differ in the choice of preprocessing steps, selected features and classifier. Some of the most significant approaches are discussed below.

The authors in [97] present one of the earliest appearance-based face detectors and introduce innovations which are widely used nowadays. The approach in [97] is based on the collection of labeled face and non-face images. The database is then artificially augmented by mirroring, rotation, scaling and translating the images by a small offsets in order to make the classifier less sensitive to such effects. Authors then apply an ANN directly to the gray-level training examples of the size 20×20. The resulting ANN outputs the likelihood of a face at the center of every overlapping window. Since several overlapping patches may appear near a face, an

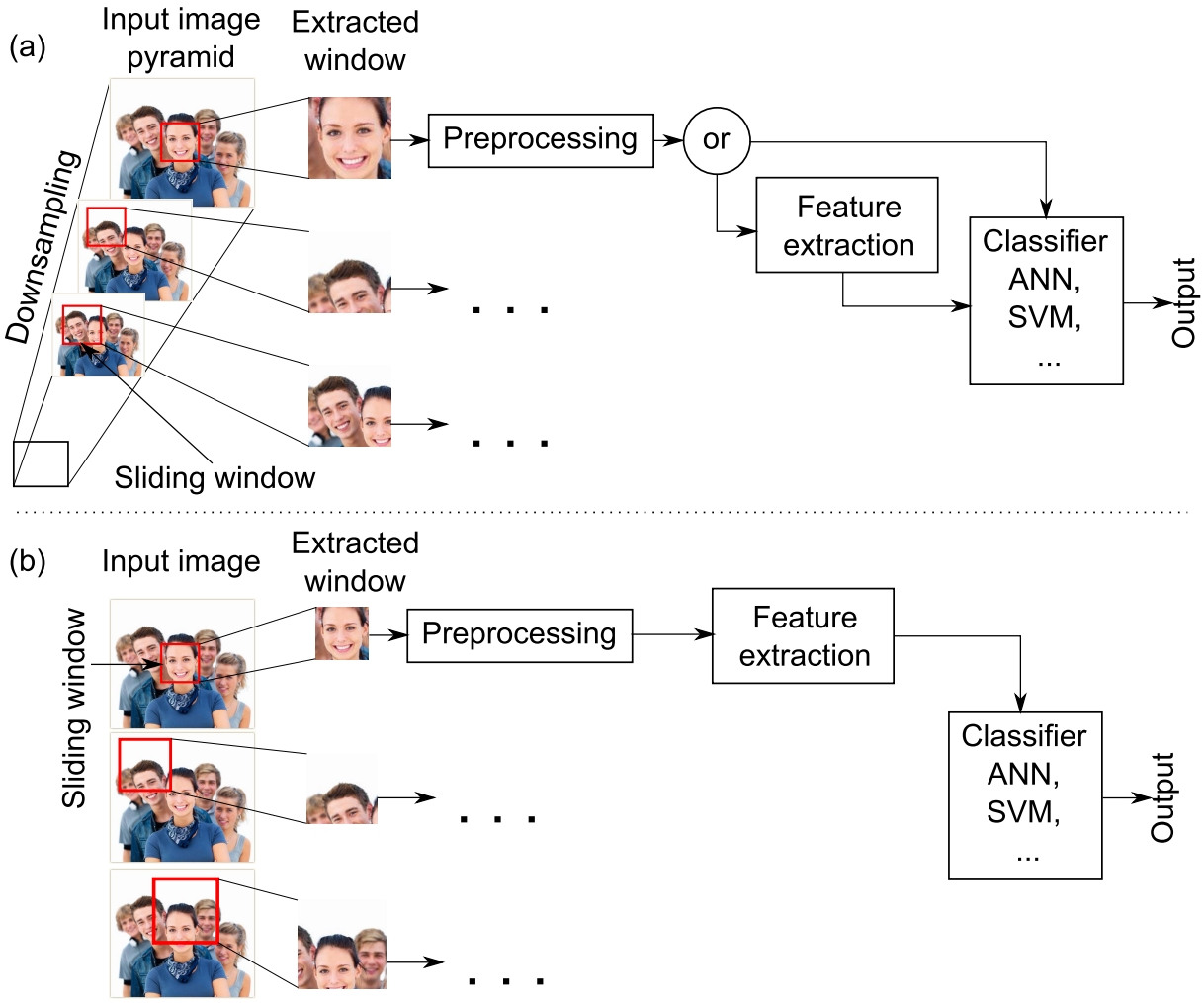


Figure 3.1: The block-diagram of appearance-based face detection system with two sliding window concepts: (a) - constant size of the sliding window, and (b) - variable size of the sliding window (semantically similar to [97])

additional merging network is used to merge multiple detections.

Instead of an ANN authors in [90] use a support vector machine to classify the input patterns as either face or non-face. The SVM principles discussed in the earlier sections are utilized in [90].

A distribution based system which incorporates two main steps is developed in [107]. First, the distribution of face patterns is partitioned into six clusters and then the PCA subspace is fitted to each of the resulting clusters. The same is done for the non-face examples. A distance is then computed between an input pattern and its projection onto the PCA subspace for each of the 12 clusters. In a second step, a neural network is trained to classify faces and non-faces based on this distances.

The best known and currently the most widely used face detection algorithm was proposed in [117]. The authors introduced the *boosting* technique to the computer vision problem, which involves training of a series of simple classifiers with increasing discriminative power and then

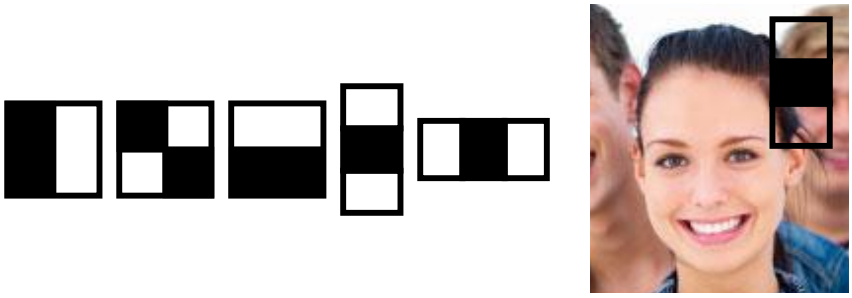


Figure 3.2: Haar-like features used in boosting-based face detection algorithm (semantically similar to [117])

combining their outputs. The hypothesis of the boosting classifier is a sum of *weak learner*:

$$h_\alpha(\boldsymbol{x}) = \text{sign}\left(\sum_{j=0}^{n} \alpha_j h_j(\boldsymbol{x})\right) \tag{3.1}$$

where each weak learner $h_j(\boldsymbol{x})$ is a very simple function of the input $\boldsymbol{x}$, which contributes to the classification performance. In most cases the weak learners are threshold functions.

The authors in [117] propose the features that are the differences of rectangular regions in the input image, see Figure 3.2 for details. Features in the Figure 3.2 are derived from Haar wavelets and are computed by subtraction of pixels inside the white rectangles from the pixels in the black ones. The advantage of these features is their discriminating power and fast computation time. These properties makes the algorithm applicable in real-time face detection applications, however the training time can be quite slow (in the order of weeks) because of the large number of possible features that need to be processed at each stage of the learning.

3.2 Face detection using Local Binary Patterns

Local Binary Pattern operator was originally introduced as a texture descriptor, but the discriminative power and computational simplicity of the LBP enhanced the scope of the operator in many computer vision fields. One of them is face detection. Some of the most significant papers in this field are discussed below.

The idea of LBP-based face detection is not novel. One of the first LBP-based face detection algorithms was introduced in 2004 [40]. Authors in [40] introduced the principle, which is based on the combination of the LBP and SVM classifier. Face is represented by the LBP histograms of the overlapping face regions with following parameters of the LBP operator: $P = 4, R = 1$. Additionally the description of the face is enhanced by including the global LBP histogram, which is computed over the whole face image. The length of the final face feature vector is 203. Experimental results in [40] show that the combination of LBP and SVM performs well if compared to the state of the art methods.

The methodology for object detection using spatial histogram features is introduced in [130].

The performance of the algorithm is demonstrated on the task of car detection, however the principle can be extended for other applications. Authors in [130] describe the representation of the object, which combines texture and spatial structures. Specially, the object is modeled by their spatial LBP histograms over local patches. A Fisher criterion is employed in order to evaluate the discriminability of spatial LBP histograms. A hierarchical classification using cascade histogram matching and support vector machine is employed for object detection.

Face detection algorithm, which is based on Multi-Block LBP representations is introduced in [131]. The basic idea of Multi-Block LBP is that the difference rule in Haar-like features (Figure 3.2) is replaced with encoding of rectangular regions by the modified local binary pattern operator (Multi-Block LBP). Proposed face detection approach is more robust then the one based on traditional Haar-like features. Another advantage of MB-LBP is that the number of exhaustive set of MB-LBP features is much smaller than Haar-like features, which makes the learning faster. A boosting-based learning method is selected to achieve the goal of feature selection and face detection with weak classifiers.

Another face detection approach is developed in [123]. The key concept of the algorithm is a novel feature space, namely Locally Assembled Binary (LAB) features. LAB features are inspired by the success of Haar feature and LBP for face detection, but these ideas are hardly modified in [123] and only general concepts are similar. The classification function is determined with RealBoost learning algorithm.

Above mentioned LBP-based face detection algorithms have *one common property*: the sliding window with constant size is utilized in the detection stage of the algorithms. Down-sampling is needed in this case, which results in the loss of statistical data about the object on a very first stage of the algorithm.

3.2.1 Nearest Neighbor Classifier - based face detection

The idea to use the combination of Local Binary Patterns and Nearest Neighbor Classifier for the task of face detection is first introduced in [85] (Nikisins et al.). Proposed method belongs to appearance-based face detection approaches, see Figure 3.1 (b). The advantage of the algorithm is the use of sliding window with variable size, which is needed to detect faces of various sizes. The down-sampling stage is absent in this case, and therefore statistical data about the object is not lost on a very first stage of the algorithm.

The general structure of the LBP and NNC based face detection algorithm is schematically displayed in the Figure 3.3. The first step of the algorithm is to calculate the LBP transformation of the input image, Figure 3.3 (2). The LBP transformed image is scanned with the sliding window of variable size. At each position of the sliding window the representation of the object is calculated. The spatially enhanced histogram is selected as the representative feature vector, see section 2.1.1 for details. Authors in [85] (Nikisins et al.) compute the histogram of the LBP image in the sliding window as a representation of the pattern which can be viewed as

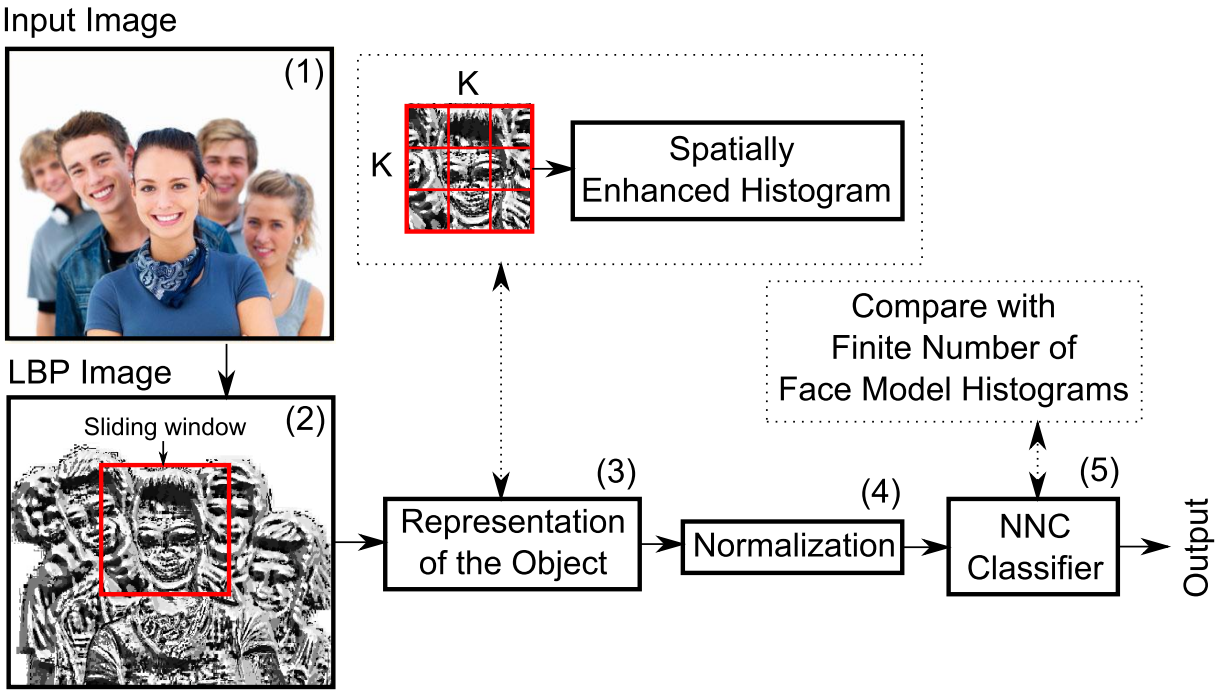


Figure 3.3: The block-scheme of NNC and LBP based face detection algorithm

a particular case of the spatially enhanced histogram with $K = 1$. However, in this case the spatial information about the object is completely lost and therefore the overall performance of the detector degrades. The spatially enhanced histogram is implemented in the algorithm in order to introduce the spatial information about the face in the feature vector. The negative aspect is that the length of the feature vector is now longer: $N = K \cdot 2^P$, instead of $N = 2^P$ bins in the ordinary histogram, however this issue can be overcome by varying the parameter P of the LBP operator. The normalization of the feature vector is needed at each scanning position due to variable size of the sliding window in order to get a coherent description of the face, Equation (2.5).

At each scanning position (x, y) the Euclidean distances between the spatially enhanced histogram and the finite number of face model histograms are calculated. The distance matrix $\boldsymbol{D}^{s,i}$ is obtained for each size of the sliding window s and each face model histogram $\boldsymbol{h}^{f,i}$:

$$\boldsymbol{D}^{s,i}_{x,y} = \sqrt{\sum_{j=1}^{N} (\boldsymbol{h}_j - \boldsymbol{h}^{f,i}_j)^2}, \tag{3.2}$$

where $\boldsymbol{D}^{s,i}_{x,y}$ is the value of Euclidean distance at (x, y) position of the sliding window and $\boldsymbol{h}_j$ stands for the corresponding spatially enhanced histogram.

An example of the distance matrix is displayed in the Figure 3.4, where scanning is performed with the sliding window of the size equal to expected size of the face. Scanning and classification are usually a time consuming operations, therefore scanning positions (x, y) are usually selected with a small step which does not degrade the detection performance. The step

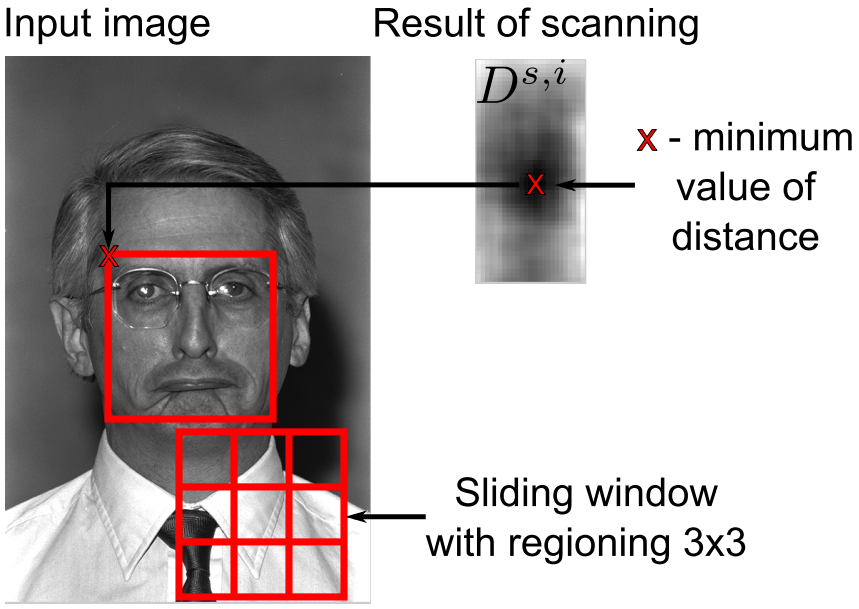


Figure 3.4: An example of the distance matrix $\boldsymbol{D}^{s,i}$ with following scanning parameters: regioning grid $K = 3$, step of the sliding window is equal to 5 pixels

for the positions of the sliding window Δ_s in Figure 3.4 is equal to 5 pixels.

The coordinates $(x^{s,i}_{face}, y^{s,i}_{face})$ of the top left corner of the face bounding box in the input image at a particular scale s for the face model $\boldsymbol{h}^{f,i}$ are determined by the coordinates $(x^{s,i}_{\min(\boldsymbol{D})}, y^{s,i}_{\min(\boldsymbol{D})})$ of the minimum in the similarity matrix $\boldsymbol{D}^{s,i}$ (see Figure 3.4 for details):

$$\begin{aligned} \{x^{s,i}_{\min(\boldsymbol{D})}, y^{s,i}_{\min(\boldsymbol{D})}\} &= \text{find}(\boldsymbol{D}^{s,i} = \min(\boldsymbol{D}^{s,i})), \\ x^{s,i}_{face} &= \Delta_s(x^{s,i}_{\min(\boldsymbol{D})} - 1) + R + 1, \\ y^{s,i}_{face} &= \Delta_s(y^{s,i}_{\min(\boldsymbol{D})} - 1) + R + 1. \end{aligned} \tag{3.3}$$

Sizes of the faces in the input image vary considerably even in semi-controlled environment, therefore the detections with various sizes of the sliding window are needed: $\boldsymbol{s} = (s_1, s_2, \ldots, s_n)$. The number of face models is also usually greater than one $k \geq 1$ which results in the total number of detections equal to $n \cdot k$. The combination of multiple detections is called **merging**. The simplest merging approach selects the detection with smallest Euclidean distance. In this case the minimum value of the Euclidean distance $d^{s,i}_{\min}$ in each matrix $\boldsymbol{D}^{s,i}$ is determined:

$$\{x^{s,i}_{\min(\boldsymbol{D})}, y^{s,i}_{\min(\boldsymbol{D})}, d^{s,i}_{\min}\} = \text{find}(\boldsymbol{D}^{s,i} = \min(\boldsymbol{D}^{s,i})). \tag{3.4}$$

The most probable location (x_{face}, y_{face}) and size s_{face} of the face are now determined by the minimum in reference values $\{d^{s,i}_{\min}\}, s = 1, \ldots, n; i = 1, \ldots, k$.

To determine the face model histograms $\boldsymbol{h}^{f,i}$ the **color FERET** face database [1] is adopted. Face regions are extracted from all frontal images of the database, which are stored in the **fa** and **fb** subsets. The number of persons in the dataset is 993 with at least two frontal face im-

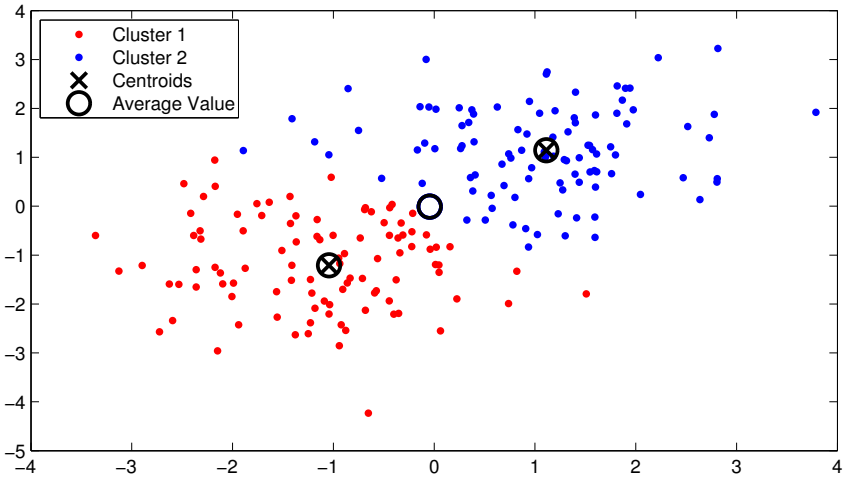


Figure 3.5: An example of **k-means** clustering for 2D data points with two clusters

ages available for each person, which results in total of 2722 images. The number of training examples is then artificially augmented by rotating and scaling of the origins. The normalized spatially enhanced histograms of the LBP transformed face images are next calculated for all samples. The method of unsupervised learning, namely, **k-means** clustering is selected to partition M face histograms into k clusters with corresponding centroids. The centroids are assumed to be normalized face model histograms. An example of **k-means** operation on 2D data points with two clusters is displayed in the Figure 3.5. One of possible approaches for the selection of face model is to calculate the average of all training examples, however this method is oversimplified and does not reflect the spatial distribution of the training data. The spatial information about the face class is enriched in the case of clustering and centroids are a better choice for facial models [51], see Figure 3.5 for details.

3.2.2 Artificial Neural Network - based face detection

The idea to use the combination of Local Binary Patterns and Artificial Neural Network for the face detection problem is proposed in [83] (Nikisins et al.). The general structure of the algorithm is very similar to the LBP and NNC based face detection approach. The only difference is in the classification module, which is now an Artificial Neural Network, see Figure 3.6. The advantage of this setup is the flexibility of the classifier, which allows to adjust the trade-off between the dimensionality of the feature space and the complexity of the classifier. Blocks (1) - (4) in the Figure 3.6 are discussed in details in the subsection 3.2.1 and only the classification module is described here.

At each scanning position (x, y) of the sliding window the probability of being a face pattern is calculated by the pre-trained ANN based on the input representation of the object. The probability matrix is obtained for each size s of the sliding window:

$$\boldsymbol{P}^{s}_{x,y} = h_w(\boldsymbol{h}_{x,y}), \tag{3.5}$$

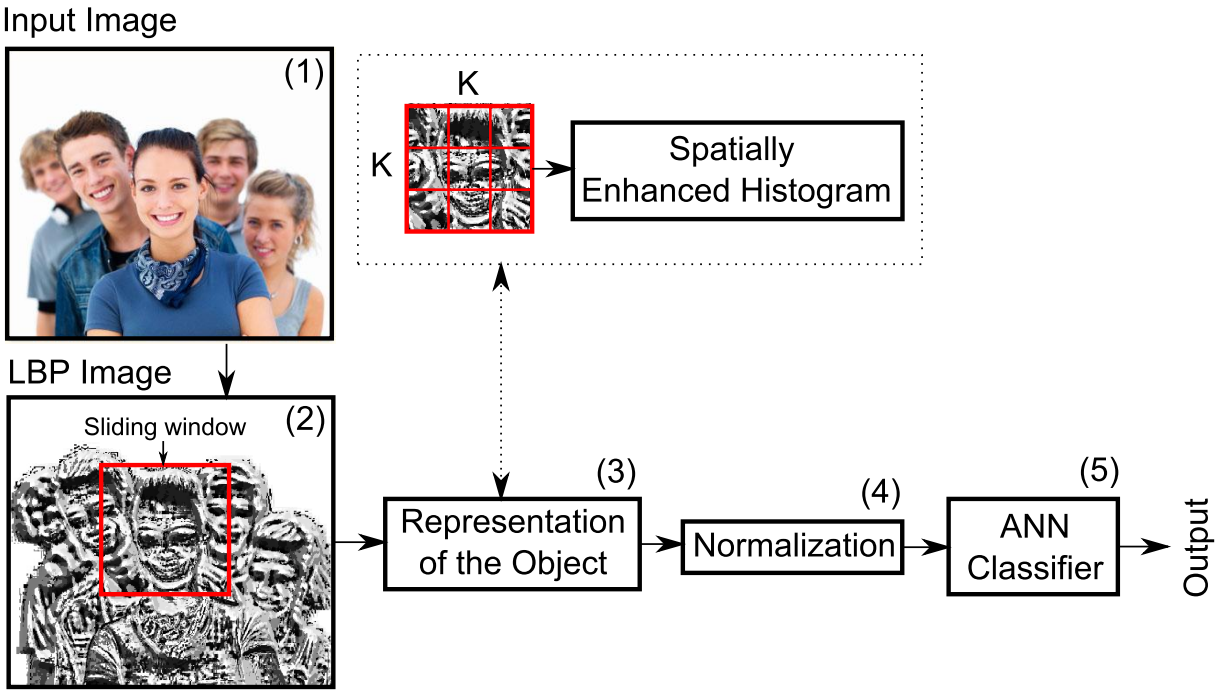


Figure 3.6: The block-scheme of LBP and ANN based face detection algorithm

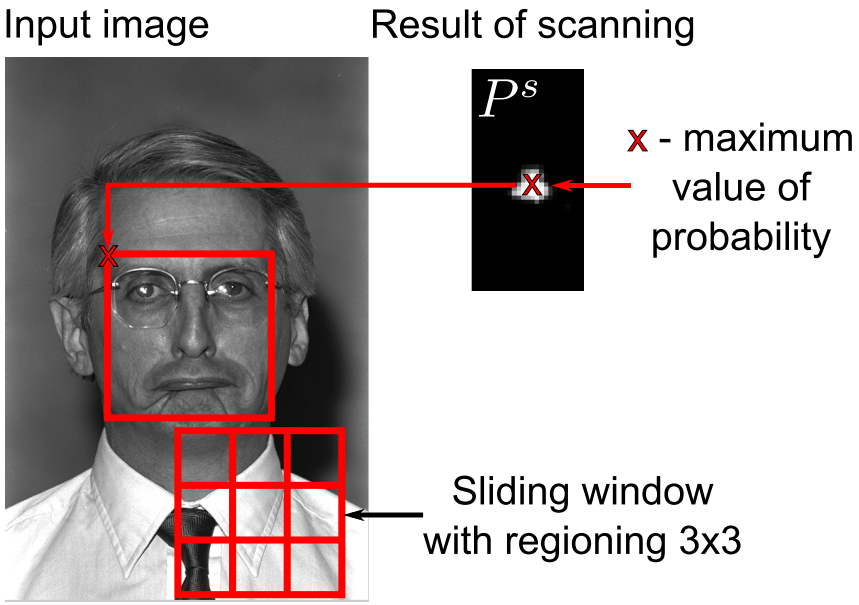


Figure 3.7: An example of the probability matrix $\boldsymbol{P}^s$ with following scanning parameters: regioning grid $K = 3$, step of the sliding window is equal to 5 pixels, s is equal to the expected size of the face

where $\boldsymbol{P}^s_{x,y}$ is the value of the probability at the position (x, y) for the scale s, which is equal to the hypothesis value h_w of an ANN for the corresponding spatially enhanced LBP histogram $\boldsymbol{h}_{x,y}$. An output value $h_w(\boldsymbol{h}_{x,y})$ of an ANN is calculated according to the methodology described in the subsection 2.3.2.

An example of the probability matrix is displayed in the Figure 3.7, where scanning is performed with the sliding window of the size equal to expected size of the face. The step for the

positions of the sliding window Δ_s in Figure 3.7 is equal to 5 pixels.

The coordinates (x^s_{face}, y^s_{face}) of the top left corner of the face bounding box in the input image at a particular scale s are determined by the coordinates $(x^s_{\max(\boldsymbol{P})}, y^s_{\max(\boldsymbol{P})})$ of the maximum in the probability matrix $\boldsymbol{P}^s$ (see Figure 3.7):

$$\begin{aligned} \{x^s_{\max(\boldsymbol{P})}, y^s_{\max(\boldsymbol{P})}\} &= \text{find}(\boldsymbol{P}^s = \max(\boldsymbol{P}^s)), \\ x^s_{face} &= \Delta_s(x^s_{\max(\boldsymbol{P})} - 1) + R + 1, \\ y^s_{face} &= \Delta_s(y^s_{\max(\boldsymbol{P})} - 1) + R + 1, \end{aligned} \tag{3.6}$$

where the summand $(R + 1)$ is added in order to compensate the size of LBP image which is smaller than the original input image.

Similar to the NNC-based face detection approach the detections with various sizes of the sliding window are needed: $\boldsymbol{s} = (s_1, s_2, \ldots, s_n)$, but in this case the total number of detections is equal to n - one detection per scale. The combination of multiple detections is called merging. The simplest merging approach selects the detection with highest probability. In this case the maximum probability value $p^s_{\max}$ in each matrix $\boldsymbol{P}^s$ is determined:

$$\{x^s_{\max(\boldsymbol{P})}, y^s_{\max(\boldsymbol{P})}, p^s_{\max}\} = \text{find}(\boldsymbol{P}^s = \max(\boldsymbol{P}^s)). \tag{3.7}$$

The most probable location (x_{face}, y_{face}) and size of the face s_{face} are now determined by the maximum in reference values $\{p^s_{\max}\}, s = 1, \ldots, n$.

The training of ANN classifier (Figure 3.6, block (5)) is performed on face and non-face spatially enhanced LBP histograms, which are extracted from the color FERET database. The process of ANN training and tuning is discussed in results section.

3.2.3 Support Vector Machine - based face detection

The effectiveness of the combination of LBP and Support Vector Machine for the face detection task is first discussed in [40]. The algorithm in [40] is an appearance-based approach and incorporates the principle of sliding window with constant size, see Figure 3.1 (a) for details. The size of the sliding window is set to 19×19 pixels, which limits the scope of the algorithm to the task of face detection in low resolution images and degrades the overall performance of the system due to reduced statistical information about the object.

The proposed algorithm is very similar to the LBP and ANN based face detection approach, see Figure 3.6. The only difference is in the classification module, which is replaced by Support Vector Machine, Figure 3.6 block (5). The advantage of this setup is the robustness of the classifier. SVM is well founded in statistical learning theory and has been successfully applied in various computer vision problems. Blocks (1) - (4) in the Figure 3.6 are discussed in details in the subsection 3.2.1 and only the classification module is described here.

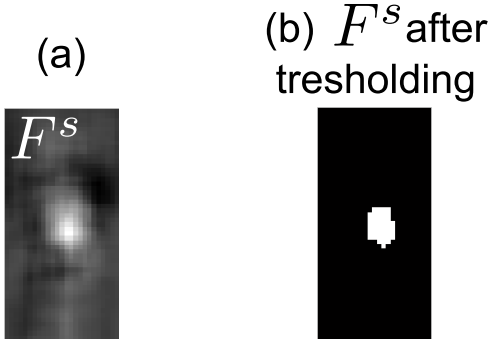


Figure 3.8: An example of the discriminative matrix $\boldsymbol{F}^s$ before (a) and after (b) thresholding with following scanning parameters: $K = 3$, $\Delta_s = 5$, s is equal to the expected size of the face

At each scanning position (x, y) of the sliding window the value of discriminative function is calculated by the pre-trained non-linear SVM classifier based on the input representation of the object. The matrix with values of discriminative function is obtained for each size s of the sliding window:

$$\boldsymbol{F}^s_{x,y} = f(\boldsymbol{h}_{x,y}), \tag{3.8}$$

where $\boldsymbol{F}^s_{x,y}$ is the value of discriminative function at the position (x, y) for the scale s, which is calculated according to the methodology described in the subsection 2.3.3, Equation (2.45).

An example of the discriminative matrix is displayed in the Figure 3.8 (a), where scanning is performed with the sliding window of the size equal to the expected size of the face. The step for the positions of the sliding window Δ_s in Figure 3.8 is equal to 5 pixels. The thresholding of the values in matrix $\boldsymbol{F}^s$ is usually performed according to the Equation (2.37) in order to determine the class of the object, see an example of $\boldsymbol{F}^s$ after thresholding in Figure 3.8 (b). However in this case the resulting position of the face is unclear, therefore the original discriminative matrix $\boldsymbol{F}^s$ is selected for further processing.

Similar to an ANN-based face detection approach the coordinates (x^s_{face}, y^s_{face}) of the top left corner of the face bounding box in the input image at a particular scale s are determined by the coordinates $(x^s_{\max(\boldsymbol{F})}, y^s_{\max(\boldsymbol{F})})$ of the maximum in the discriminative matrix $\boldsymbol{F}^s$ (see Figure 3.8 (a)):

$$\begin{aligned} \{x^s_{\max(\boldsymbol{F})}, y^s_{\max(\boldsymbol{F})}\} &= \text{find}(\boldsymbol{F}^s = \max(\boldsymbol{F}^s)), \\ x^s_{face} &= \Delta_s(x^s_{\max(\boldsymbol{F})} - 1) + R + 1, \\ y^s_{face} &= \Delta_s(y^s_{\max(\boldsymbol{F})} - 1) + R + 1. \end{aligned} \tag{3.9}$$

The detections with various sizes of the sliding window are needed: $\boldsymbol{s} = (s_1, s_2, \ldots, s_n)$. The combination of multiple detections is based on the selection of the result with highest value of discriminative function. In this case the maximum value of discriminative function $f^s_{\max}$ is determined in each matrix $\boldsymbol{F}^s$:

$$\{x^s_{\max(\boldsymbol{F})}, y^s_{\max(\boldsymbol{F})}, f^s_{\max}\} = \text{find}(\boldsymbol{F}^s = \max(\boldsymbol{F}^s)). \tag{3.10}$$

The most probable location (x_{face}, y_{face}) and size of the face s_{face} are now determined by the maximum in reference values $\{f^s_{\max}\}, s = 1, \ldots, n$.

The training of SVM classifier is performed on face and non-face spatially enhanced LBP histograms, which are extracted from the color FERET database. The process of SVM training and tuning is discussed in the subsection 3.6.4.

3.3 Efficient histogram based sliding window

An issue for the proposed face detection methodologies is the development of the effective histogram-based sliding window for the calculation of **spatially enhanced histograms** at each scanning position. Similar approaches are discussed in [121], however our methodology [83] (Nikisins et al.) is optimized for the computation of **spatially** enhanced histograms, see Figure 2.2.

The first step of the proposed sliding window methodology is to calculate two banks of histograms: $\boldsymbol{H}^y$ and $\boldsymbol{H}^x$, see Figure 3.9 for details. Each row of $\boldsymbol{H}^y$ is the histogram of the corresponding row in the LBP transformed region $\boldsymbol{R}^y$ (Figure 3.9). Similarly rows in $\boldsymbol{H}^x$ are the histograms of corresponding columns of $\boldsymbol{R}^x$ region. The dimensionality of the matrices: $\boldsymbol{H}^y \in \mathbb{R}^{m \times 2^P}$, $\boldsymbol{H}^x \in \mathbb{R}^{n \times 2^P}$, $\boldsymbol{R}^y \in \mathbb{R}^{m \times a}$ and $\boldsymbol{R}^x \in \mathbb{R}^{a \times n}$, where m and n are respectively the number of rows and columns in the LBP image, a - size (in pixels) of the squared cell in the LBP regioning grid.

Banks $\boldsymbol{H}^y$ and $\boldsymbol{H}^x$ are next utilized in the process of histogram calculation of the region $\boldsymbol{R}$ (Figure 3.9, white square) for all possible positions of that region. The histogram at the starting position, row number $i = 1$ and column number $j = 1$, is equal:

$$\boldsymbol{h}_{start} = \boldsymbol{h}_{1,1} = \sum_{l=1}^{a} \boldsymbol{h}^y_l,$$

where $\boldsymbol{h}^y$ and $\boldsymbol{h}^x$ are respectively the rows of the $\boldsymbol{H}^y$ and $\boldsymbol{H}^x$: $\boldsymbol{H}^y = \{\boldsymbol{h}^y_1; \boldsymbol{h}^y_2; \ldots; \boldsymbol{h}^y_m\}$ and $\boldsymbol{H}^x = \{\boldsymbol{h}^x_1; \boldsymbol{h}^x_2; \ldots; \boldsymbol{h}^x_n\}$.

The cell $\boldsymbol{R}$ of the regioning grid of the sliding window is first shifted *vertically*. For the recalculation of the histogram $\boldsymbol{h}_{i+1,j}$ of the region $\boldsymbol{R}$ at the position $(i+1, j)$ the following operations are performed:

$$\boldsymbol{h}_{i+1,j} = \boldsymbol{h}_{i,j} - \boldsymbol{h}^y_i + \boldsymbol{h}^y_{i+a},$$

at the same time the bank $\boldsymbol{H}^y$ is updated:

$$\boldsymbol{h}^y_{i,(I^{LBP}_{i,j}+1)} \leftarrow \boldsymbol{h}^y_{i,(I^{LBP}_{i,j}+1)} - 1 \text{ and}$$

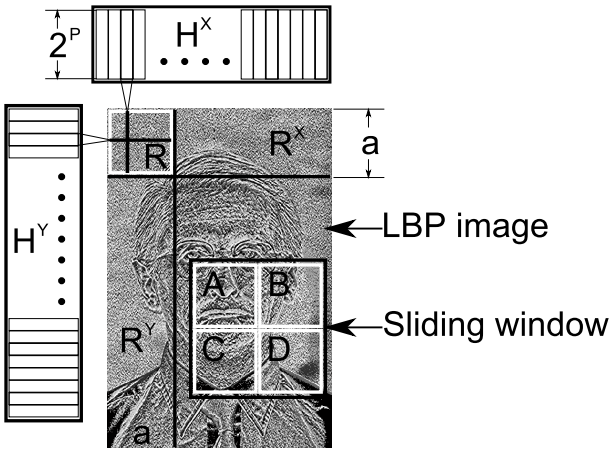


Figure 3.9: The process of calculation of the spatially enhanced LBP histogram

$$\boldsymbol{h}^{y}_{i,(I^{LBP}_{i,j+a}+1)} \leftarrow \boldsymbol{h}^{y}_{i,(I^{LBP}_{i,j+a}+1)} + 1,$$

where the notation $I^{LBP}_{i,j}$ stands for the value of LBP image at the position (i, j). The $\boldsymbol{h}_{start}$ value is updated when *horizontal* shift is performed:

$$\boldsymbol{h}_{1,j+1} = \boldsymbol{h}_{start} \leftarrow \boldsymbol{h}_{start} - \boldsymbol{h}^{x}_{j} + \boldsymbol{h}^{x}_{j+a}$$

The result of the proposed scanning methodology is a three dimensional array with histograms $\boldsymbol{h}_{i,j}$: $\boldsymbol{H} \in \mathbb{R}^{(m-a+1)\times(n-a+1)\times 2^P}$. These histograms are used to calculate the spatially enhanced histogram at any position of the sliding widow. For the position of the sliding window in the Figure 3.9 the spatially enhanced histogram is $\boldsymbol{h}^{s}_{i,j} = \{\boldsymbol{h}_A, \boldsymbol{h}_B, \boldsymbol{h}_C, \boldsymbol{h}_D\}$, where the vectors $\boldsymbol{h}_A, \boldsymbol{h}_B, \boldsymbol{h}_C, \boldsymbol{h}_D$ are corresponding entries of the $\boldsymbol{H}$. The histogram $\boldsymbol{h}^{s}_{i,j}$ is analyzed by the classifier in order to classify it as a face or non-face pattern.

3.4 Face detection: performance evaluation

The performance of the face detection algorithm is evaluated on a database with only one face located in each test image. This scenario is easier than the task of face detection in images with multiple persons, but is the reality in biometric access and security systems which are often designed to operate with a single person. The correct face detection is determined with two main parameters:

- the displacement of face region from the expected (ground-truth) face position,
- the deviation between the detected and actual sizes of the face.

Both of the above parameters may be encoded into a single restrictive criteria. Authors in [52] introduced a relative error measure based on the distance between the detected and the expected eye center positions. Let C_L and C_R be the true positions of the left and right eyes

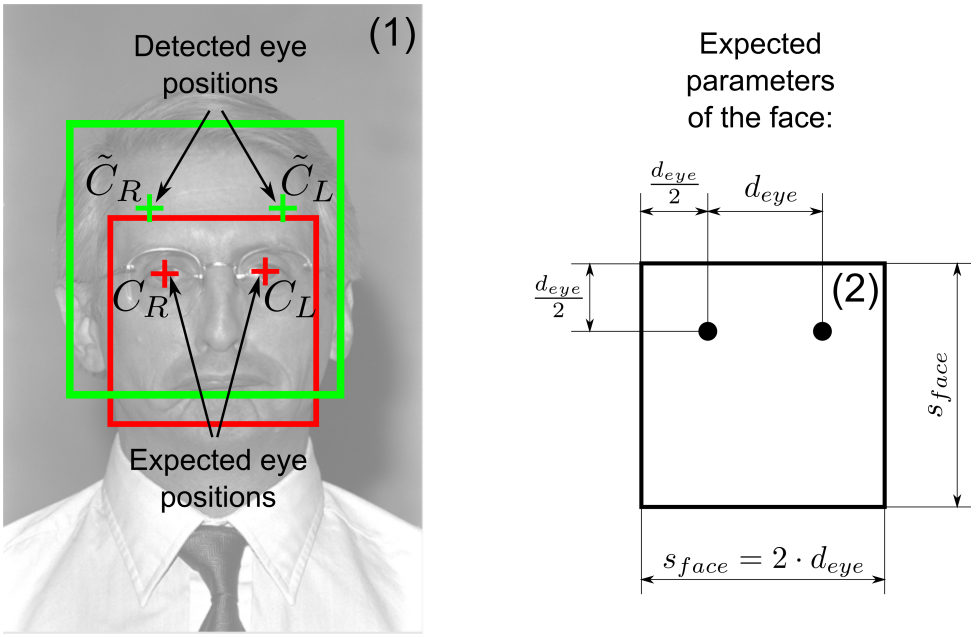


Figure 3.10: The displacement of the detected eye positions from the expected coordinates (1); the expected parameters of the face (2)

correspondingly and let $\tilde{C}_L$ and $\tilde{C}_R$ be the left and right eye positions estimated by the face detection algorithm, see Figure 3.10 (1) for details. The criteria for the evaluation of the detector performance can be written as follows:

$$\eta_{face} = \frac{\max(d(C_L, \tilde{C}_L), d(C_R, \tilde{C}_R))}{d(C_L, C_R)}, \tag{3.11}$$

where the notation $d(a, b)$ stands for the value of Euclidean distance between points a and b.

In literature a successful detection is often accounted if $\eta_{face} \leq 0.25$, which corresponds to a quarter of an interocular distance [95]. The expected parameters of the face are displayed in the Figure 3.10 (2), therefore the locations of the points $\tilde{C}_L$ and $\tilde{C}_R$ can easily be determined if the coordinates of the detected face bounding box are known, Equation (3.3).

The distribution of the proposed errors η_{face} for all faces in the test set is converted into empirical cumulative form. Such representation is called Empirical Cumulative Distribution Function (ECDF). This approach provides a unified face detection measure which is accepted by many researchers.

3.5 Experimental setup

Proposed face detection approaches are based on various Machine Learning techniques and highly rely on the training data to find a discriminative function between two classes: faces and non-faces. Robustness of the detector to appearance variability is achieved by incorporating different aspects of the scene into the training data. The amount of issues in the face detection

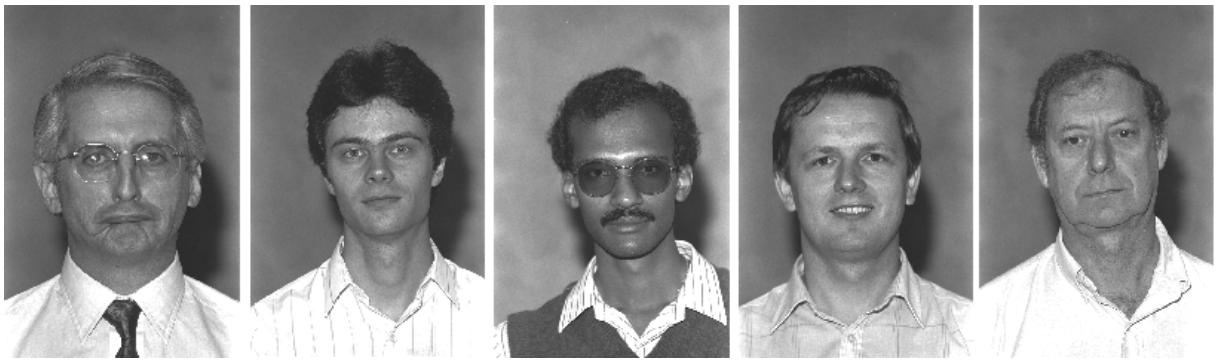


Figure 3.11: Frontal face images for the first five persons in the color FERET [1] database

task is significant and and can be explained with three main sources:

- the physiological and emotional factors of the face such as different facial expressions, skin color, gender, aging, beard and others,
- partial occlusions in the face image: glasses, hair and other objects,
- imaging problems: different scales of the faces in the image, rotations of the head both from the vertical position of the face and off-plane, lighting conditions, different parameters of the camera.

Most of the above mentioned aspects with some limitations are introduced in the utilized training data. The color FERET database is chosen as the basis of the training data. This database covers most of the issues that may appear with some exclusions: the database is collected in the semi-controlled environment with fixed lighting conditions, the hardware setup is not changed and partial occlusions are only present in the form of glasses and hairstyle. The stated task is limited to frontal frontal face detection, therefore only the frontal subsets are selected from the database (**fa** and **fb** subsets) and only insignificant off-plane rotations which are natural for humans are present in the training set. An example of frontal face images for the first five persons in the color FERET database is displayed in the Figure 3.11. The number of frontal faces in the database is equal to 2722, which is not sufficient for training of the ANN or SVM classifier [101], [20]. This set is artificially augmented in order to get enough training data. The process of training data forming is schematically displayed in the Figure 3.12 and consists of the following steps:

- All frontal face images are horizontally mirrored. This stage is needed for the clear separation of the training and test data, because all results of the algorithms performance will be reported for *non-modified* FERET database in order to make the results accessible for the research community. The size of the test set is equal to the number of frontal faces in the color FERET database $M_{test} = 2722$.
- The rotation of the mirrored face image by the the angles $\alpha = (-10, -5, 0, 5, 10)$. This step introduces the robustness of the algorithm to the rotations of the head from the vertical

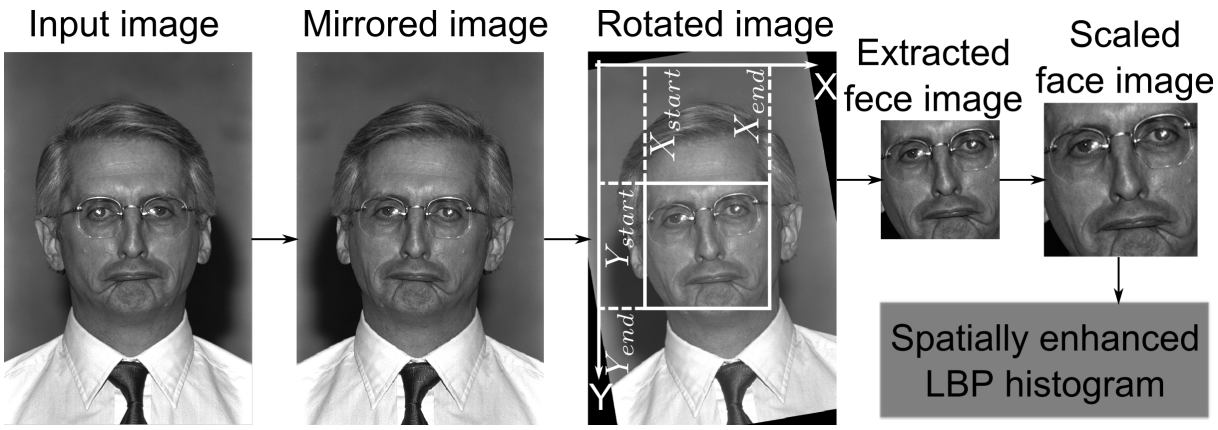


Figure 3.12: The process of forming an artificial training data for the face class

position. The distribution of the angles between the eye-line and horizontal in the frontal images of the FERET database is plotted in Figure 3.13. The range is approximately from -15 to 15 degrees. After the rotation stage the distribution is enhanced by 10 degrees in both directions.

- The face region is extracted from the rotated image. The bounding box of the face is described with four coordinates, see Figure 3.12 for details, which are calculated as follows:

$$\begin{aligned} X_{start} &= \max\left(\left\{1, \text{round}\left(\frac{X(C_R)+X(C_L)}{2} - d_{eye}\right)\right\}\right), \\ X_{end} &= \min\left(\{X_{max}, X_{start} + 2 \cdot d_{eye}\}\right), \\ Y_{start} &= \max\left(\left\{1, \text{round}\left(\frac{Y(C_R)+Y(C_L)}{2} - \frac{d_{eye}}{2}\right)\right\}\right), \\ Y_{end} &= \min\left(\{Y_{max}, Y_{start} + 2 \cdot d_{eye}\}\right), \end{aligned} \tag{3.12}$$

  where the notations $X(C_R)$ and $X(C_L)$ stand for the X coordinates of the left and right eye pupils and $Y(C_R)$, $Y(C_L)$ are the corresponding Y coordinates, see Figure 3.10 for details; X_{max} and Y_{max} are the width and hight of the input image in pixels.

- The extracted face region is scaled by the factors $Scale = (0.8, 1, 1.2)$. The distribution of face sizes in the frontal images of the FERET database is plotted in Figure 3.13. The original range is approximately from 110 to 480 pixels, which is enhanced after the scaling to the values of $\sim$90 to $\sim$580 pixels.

- The spatially enhanced LBP histogram is calculated according to the methodology described in Section 2.1.1.

The initial set of frontal face images and corresponding histograms is now augmented to a set of 40830 images. This set is sufficient for the training stage of the proposed NNC-based face detection algorithm, however it should be supplemented by the non-face training examples if

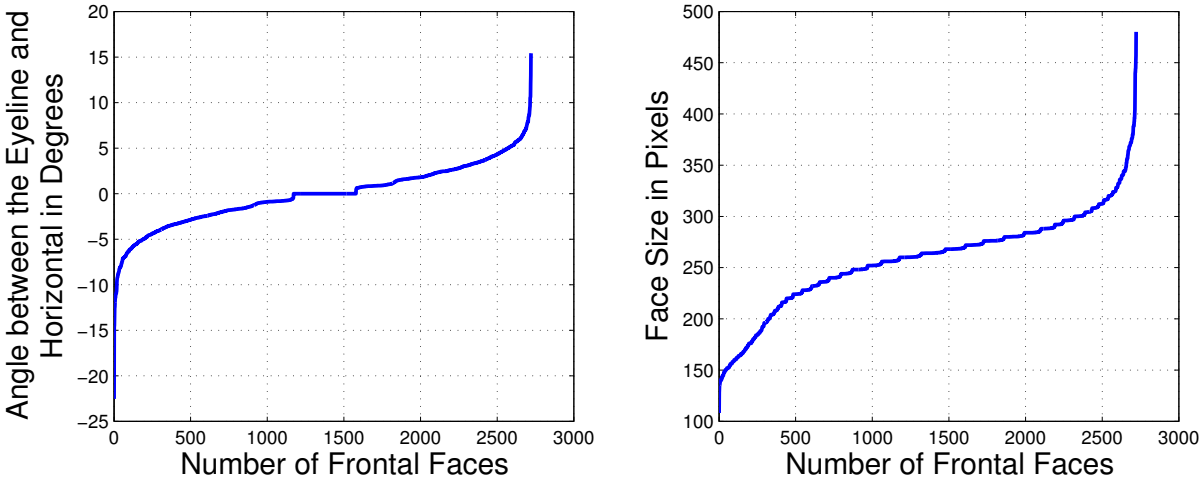


Figure 3.13: Left plot: The distribution of the angles between the eye-line and horizontal in the frontal images of the FERET database; Right plot: the distribution of face sizes in the database

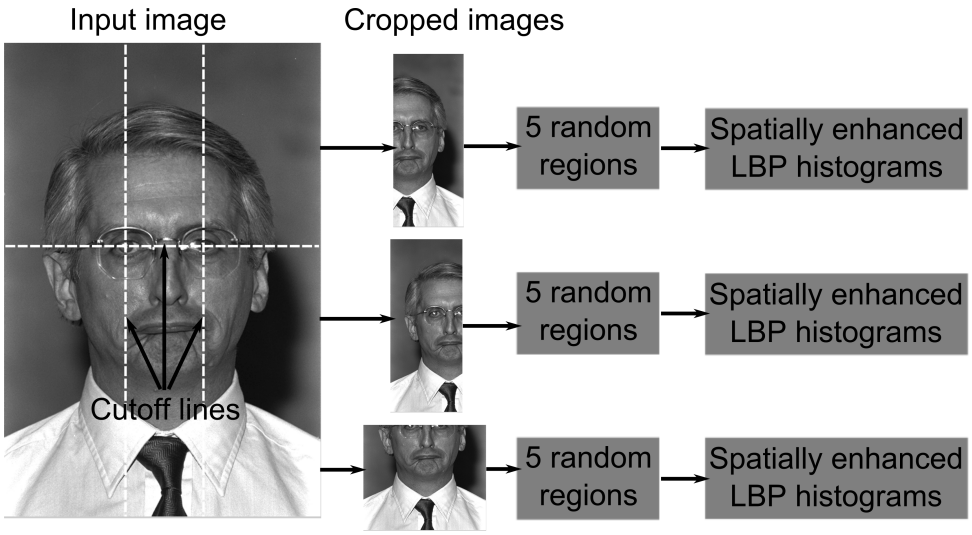


Figure 3.14: The process of forming the non-face training data

the training of an ANN or SVM is performed. The methodology of calculating the non-face patterns is described below.

The process of forming the non-face training data is schematically displayed in the Figure 3.14. The LBP-based face detection techniques are prone to detection offsets due to integral and spatially depleted nature of the face descriptive vector which in fact is a histogram. Therefore the images which include the part of the face should be included in the training set, which makes an algorithms more robust to detection offsets. This goal is achieved by cropping the input image with the face in it along the cutoff lines, see Figure 3.14 for details. Five random regions are extracted from the cropped images and the spatially enhanced LBP histogram is calculated for each random region. These operations are performed for all frontal face images in the color FERET database and 40830 non-face patterns are obtained as a result.

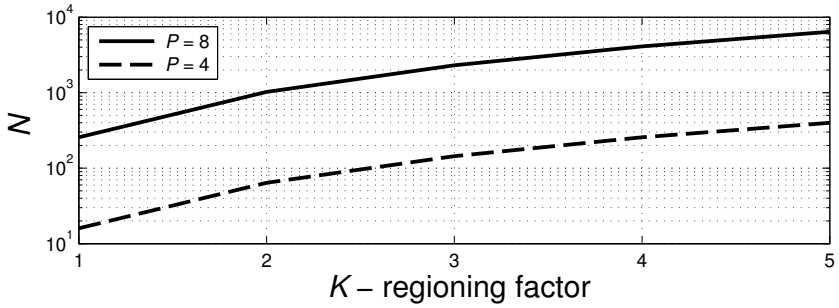


Figure 3.15: The dimensionality of the feature space N in logarithmic scale for different values of P and K

The resulting number of face and non-face LBP histograms is equal to 81660. ANN and SVM machine learning techniques require few separate data sets for the attenuation of classifier parameters, which are called Training, Cross Validation (CV) and Test sets. The initial 81660 histograms are split into Training and Cross Validation sets with corresponding proportions 70% and 30%. The Test set contains *non modified* frontal face images from the color FERET database. The resulting sizes of the sets are: $M_{train} = 57162$, $M_{CV} = 24498$ and $M_{test} = 2722$.

3.6 Simulation results

Evaluation of the proposed face detection algorithms is performed on a color FERET database. The modified images from the database are utilized in the process of the classifier training and optimization of the parameters of the algorithms. The LBP transformation is at the core of the proposed methods, therefore the evaluation of LBP operator parameters is discussed in the following subsection.

3.6.1 Evaluation of parameters of Local Binary Patterns

The number of parameters to be optimized in the proposed face detection algorithms is large, therefore it is important to distinguish the parameters that could be evaluated before the adjustment of the whole system. Attenuation of the parameters of an algorithm apart of the final system is a complicated task, which requires the development of special evaluation techniques. These techniques usually do not guarantee the best possible choice of the parameters, but are a good compromise for the reduction of parameters amount that should be adjusted in the final algorithm. The encoding of patterns in the face and non-face classes is based on the spatially enhanced LBP histograms. The histograms are determined by four main parameters:

- P - number of sampling points in the LBP label;
- R - radius of the LBP label in pixels;

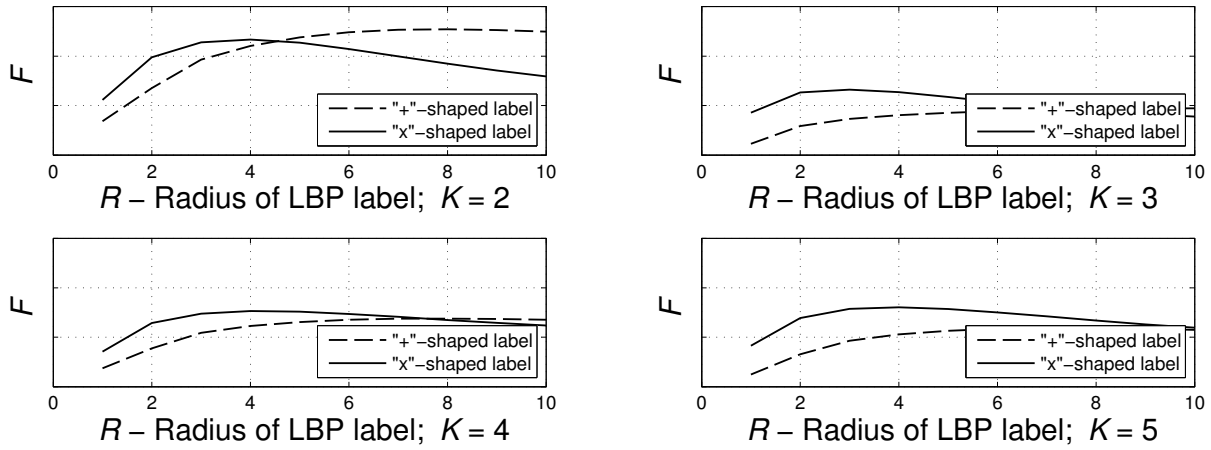


Figure 3.16: Dependencies F (3.13) for the evaluation of LBP operator radius R and structure

- Structure of the LBP label. Possible values: "+" - shaped or "x" - shaped, see Figure 2.5 for details;
- K - number of columns and rows in the regioning grid, Figure 2.2.

The LBP structure and parameters P and R are evaluated in this subsection. The regioning factor K is optimized later in conjunction with the classifier due to aspects which are discussed in the next sessions. The parameters P and K have a direct impact on the dimensionality of the feature space: $N = 2^P \cdot K^2$. The dimensionality of the feature space N in logarithmic scale for different values of P and K is plotted in Figure 3.15. In order to introduce the spatial information about the object in the LBP histogram the value of K should meet the following constraints : $K \geq 2$, which results in a highly dimensional feature space with $P = 8$: $N \geq 1024$. The value of $P = 8$ is a good choice for the face recognition stage, but is not suitable for the task of face detection due to unacceptably high dimensionality of the feature space. Therefore, the number of sampling points is set to $P = 4$.

The next aspects to be evaluated are the structure and the radius of the LBP operator. The introduced methodology for the selection of these parameters is based on the following expression:

$$F = \frac{(M-1)/M \sum_{i=1}^{M} \sum_{j=1}^{M} d(\boldsymbol{h}_i^f, \boldsymbol{h}_j^{nf})}{\sum_{i=1}^{M} \sum_{j=1}^{M} d(\boldsymbol{h}_i^f, \boldsymbol{h}_j^f) + \sum_{i=1}^{M} \sum_{j=1}^{M} d(\boldsymbol{h}_i^{nf}, \boldsymbol{h}_j^{nf})}, \tag{3.13}$$

where the notation $d(\boldsymbol{h}_i^f, \boldsymbol{h}_j^{nf})$ stands for the value of Euclidean distance between face histogram $\boldsymbol{h}_i^f$ and non-face histogram $\boldsymbol{h}_j^{nf}$.

An intuitive idea of the Equation (3.13) is to select the R value and LBP structure, which *maximize* the ratio F between *average* inter class and intra class Euclidean distances. This methodology is originally introduced in [83] (Nikisins et al.), but the authors in [83] (Nikisins et al.) utilized this approach without the regioning of the object: $K = 1$. This aspect is added in current version of the evaluation. In general the function F and histograms $\boldsymbol{h}^f$ and $\boldsymbol{h}^{nf}$ in the Equation (3.13) are dependent on the R, K and structure of the LBP: $F(R, K, \text{structure})$,

$\boldsymbol{h}^f(R, K, \text{structure})$, $\boldsymbol{h}^{nf}(R, K, \text{structure})$. The face histograms $\boldsymbol{h}^f$ are calculated for all frontal images ($M = 2722$) in the color FERET database and non-face histograms $\boldsymbol{h}^{nf}$ are extracted from the same images according to the methodology from Section 3.5. The dependencies $F(R, K, \text{structure})$ are displayed in the Figure 3.16. The absolute values of F are not needed for the evaluation, only the locations of the maximums are of the importance.

The curves F for the "x"-shaped structure have an explicit maximum and are in most cases ($K = (3, 4, 5)$) higher than the ones of the "+"-shaped structure, see Figure 3.16. In most cases the maximum in curves of the "x"-shaped structure is achieved for $R = 4$ ($K = (2, 4, 5)$) and the value of F at $R = 4$ is also close to maximum for $K = 3$, which is a good choice for LBP radius according to the proposed methodology.

The selected parameters of the LBP operator are summarized here:

- $P = 4$ - number of sampling points in the LBP label;
- $R = 4$ - radius of the LBP label in pixels;
- Structure of the LBP label: "x" - shaped, see Figure 2.5 for details.

3.6.2 Results for Nearest Neighbor Classifier - based face detection

The amount of parameters in the proposed face detection algorithm is significant, therefore the parameters of LBP operator are set to the values determined in the Section 3.6.1 in order to reduce the space of variables. The main aspects of the proposed LBP and NNC based face detection algorithm to be evaluated in this section are:

- the regioning factor K of the regioning grid (Figure 2.2),
- the number of face models $\boldsymbol{h}^f$, which is equal to the number of centroids N^c extracted from the training data of the face class.

Both of the above factors have a direct impact on the precision and computational time of the algorithms. The regioning grid introduces the spatial information about the object into the feature vector, but increases the dimensionality of the feature space. Increasing the number of centroids enriches the information about the face class, but requires the comparison with N^c face models at each position of the sliding window which increases the computational time. Additionally, high amount of clusters makes the system prone to miss detections due to increasing proximity of the centroids to the non-face class.

Authors in [83] (Nikisins et al.) set minimum requirements for the dimensionality of the LBP feature space for reasonable performance of the LBP and ANN based face detection system. The proposed regioning factor is $K = 3$. Boundaries of regioning factor are extended here to the values $K = (2, 3, 4)$. The number of face models is selected in the range: $N^c = (2, 3, 4, 5)$. Further augment of K and N^c makes the system computationally expensive and might degrade the precision of the detector.

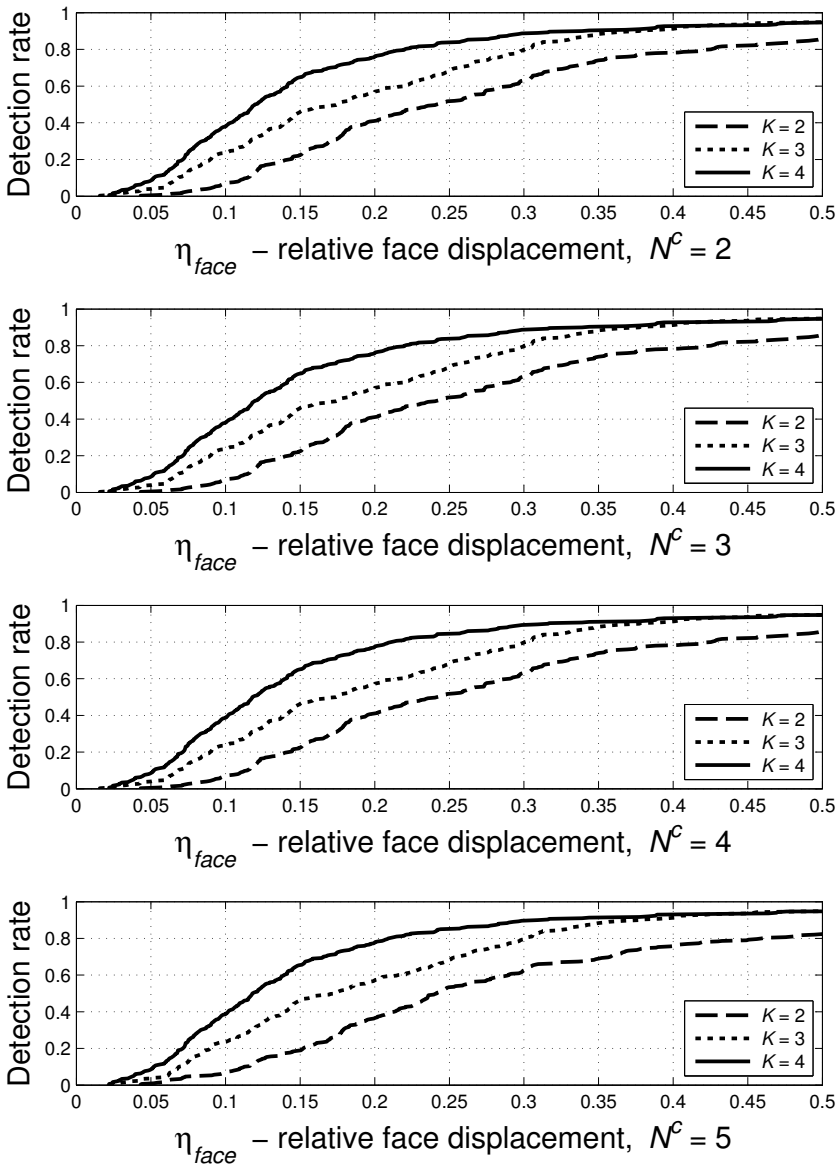


Figure 3.17: Cumulative distributions of η_{face} for different values of K and N^c of LBP and NNC based face detection algorithm; tested on 300 randomly selected frontal facial images of the color FERET

The size of the sliding window is iteratively increased from 150×150 to 400×400 pixels with 50 pixels step, because the main amount of faces in the FERET database is of that size, Figure 3.13. The step of the sliding window is $\Delta_s = 10$ pixels which is a small fraction $(1/15)$ of the minimum expected size of the face and should not degrade the precision of the detector significantly.

The total number of input image scans is equal to 72 (3 K values $\times$ 4 N^c values $\times$ 6 different scales), which means that the FERET database should be processed 72 times. The amount of calculations is significant, therefore the size of the database is reduced almost ten times and only 300 frontal facial images are randomly selected for the evaluation process. Once the best K and N^c parameters are determined the whole database is utilized for performance evaluation with

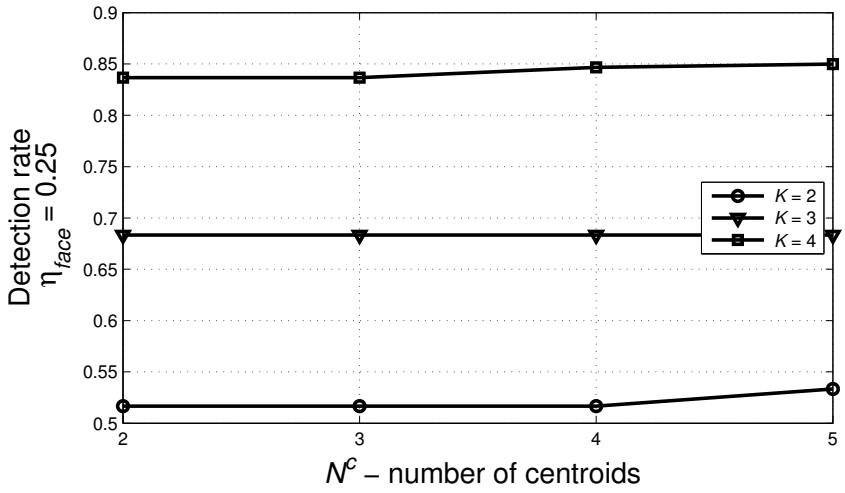


Figure 3.18: Detection rates for $\eta_{face} = 0.25$ and different values of K and N^c of LBP and NNC based face detection algorithm; tested on 300 randomly selected frontal facial images of the color FERET

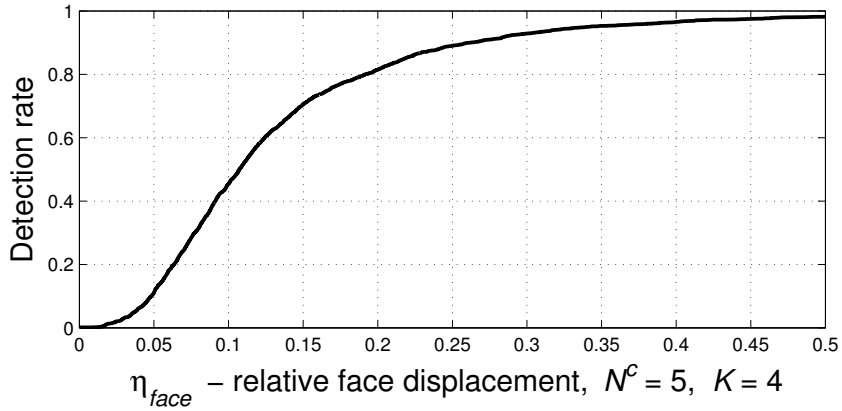


Figure 3.19: Cumulative distribution of η_{face} with $K = 4$ and $N^c = 5$ of LBP and NNC based face detection algorithm; tested on **all** frontal facial images of the color FERET

these parameters.

The corresponding cumulative distributions of η_{face} for different values of K and N^c are displayed in Figure 3.17. The borderline value of relative relative face displacement is $\eta_{face} = 0.25$, which corresponds to a quarter of interocular distance and is a criteria accepted by many researchers [95]. The detection rates at $\eta_{face} = 0.25$ for the values $K = (2, 3, 4)$ and $N^c = (2, 3, 4, 5)$ are plotted in Figure 3.18. The best detection result is achieved for the values $K = 4$ and $N^c = 5$: $P(\eta_{face} = 0.25) = 85.0\%$. The main growth of the performance is obtained by increasing the regioning factor K and only a slight improvement is observed when the number of centroids is incremented. For the value of $K = 3$ the rise of N^c value does not improve the detection rate at all.

Once the best values $K = 4$ and $N^c = 5$ are determined the algorithm is tested on all frontal face images of the color FERET database. The corresponding cumulative distribution function of η_{face} is displayed in the Figure 3.19. The value of η_{face} is calculated according to

the Equation (3.11), however this criteria might incorporate an additional biased error if ground-truth coordinates of the eyes are utilized in the calculations directly. The value $\eta_{face} = 0$ if the face is detected perfectly by the algorithm. The detected eye pupils $\tilde{C}_L$ and $\tilde{C}_R$ are **always** located on the horizontal line, see Figure 3.10 for details, while the ground-truth coordinates of the eyes are in general located on the slanted line. Therefore $\eta_{face} \neq 0$ even if the detection is correct, because the face detector is not able to determine the deviation of the eye-line from horizontal. The new ground-truth coordinates of the eyes, which are always located on the horizontal line are calculated in order to compensate the described issue:

$$\begin{aligned} X(C_L) &= \text{round}(X_{start} + \frac{3}{2}d_{eye}), \\ Y(C_L) &= \text{round}(Y_{start} + \frac{d_{eye}}{2}), \\ X(C_R) &= \text{round}(X_{start} + \frac{d_{eye}}{2}), \\ Y(C_R) &= Y(C_L), \end{aligned} \tag{3.14}$$

where the starting coordinates of the face bounding box X_{start} and Y_{start} are calculated according to the Equations (3.12) based on *non-updated* ground-truth coordinates of the eyes. The updated ground-truth coordinates are utilized for ECDF calculation in the Figure 3.19.

The detection rates for the values $\boldsymbol{\eta}_{face} = (0.25, 0.5)$ in the Figure 3.19 are:

$$\begin{aligned} P(\eta_{face} = 0.25) &= 89.0\%, \\ P(\eta_{face} = 0.5) &= 98.2\% \end{aligned}$$

The first five results of the face detection for the values $\boldsymbol{\eta}_{face} = (0.1 \pm 0.01, 0.25 \pm 0.01, 0.5 \pm 0.01)$ are displayed in Figure 3.20. The visual inspection of the results in the Figure 3.20 reveals that there are three possible sources of the detection mistakes: the error is caused by the offset of the face bounding box (for example, images (1) and (4) of $\eta_{face} = 0.25$); the size of the object is misclassified (images (1) and (5) of $\eta_{face} = 0.5$); the combination of two previous reasons. The principle of ceiling analysis is used in order to determine the error, which is contributed by incorrect size of the face. In this case the scanning of the input image is performed with the window of the size equal to the expected size of the face. In other words, the size of the face is known. This approach excludes the negative impact of the size selection procedure and allows to understand the limits of the proposed face detection approach.

The detection rate at $\eta_{face} = 0.25$ with known size of the detectable face s_{face} is $P(\eta_{face} = 0.25, s_{face} - \text{known}) = 95.9\%$, see Figure 3.21 for details. The difference with the value of detection rate, which is obtained in scenario with previously unknown size of the face is significant: $\Delta P(\eta_{face} = 0.25) = P(s_{face} - \text{known}) - P(s_{face} - \text{unknown}) = 6.9\%$. The $\Delta P(\eta_{face} = 0.25)$ amount of the precision is lost due to the need to detect faces at various sizes of the sliding win-

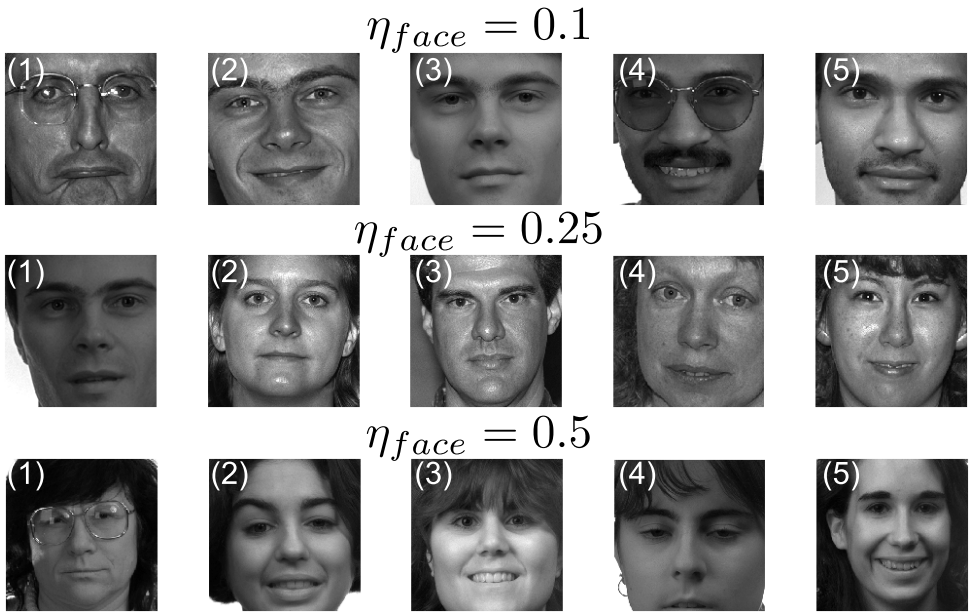


Figure 3.20: The first five detection results for different values of $\eta_{face} \pm \Delta\eta_{face}$, where the range $\Delta\eta_{face} = 0.01$

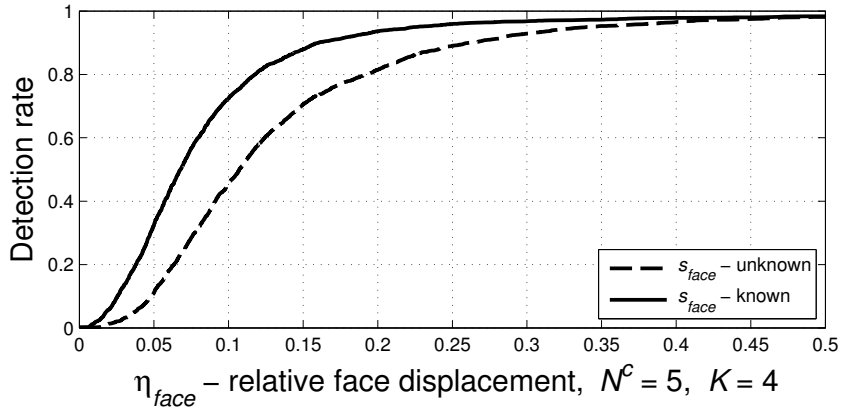


Figure 3.21: Cumulative distributions of η_{face} with $K = 4$ and $N^c = 5$ of LBP and NNC based face detection algorithm for two cases: s_{face} - known and s_{face} is determined by algorithm; tested on **all** frontal facial images of the color FERET

dow, which is probably the most significant issue to be resolved according to the proposed ceiling analysis. Possible approaches to improve the detection rates are:

- The design of more sophisticated face/non-face classifier;
- Increase of the regioning factor K;
- Design of the advanced merging of multiple detections.

The first approach is covered in the next sessions. The growth of the regioning factor K increases the dimensionality of the feature space, which reduces the computational performance of the algorithm and is not considered to be a good choice. The last option is beyond the scope of this document, but is partially covered in the publication of the author [85] (Nikisins et al.).

3.6.3 Results for Artificial Neural Network - based face detection

The first step of the algorithm is LBP transformation of the input image. The parameters of LBP operator are set to the values determined in the Section 3.6.1 in order to reduce the space of variables to be optimized. The main aspects of the proposed LBP and ANN based face detection algorithm to be evaluated in this section are:

- the regioning factor K of the regioning grid (Figure 2.2),
- the structure of the classifier, which is an ANN with variable number of neurons in the hidden layer.

Both of the above aspects are evaluated jointly, because they are strictly related and have a direct impact on possible classification problems and computational complexity of the system. Small value of K and simple structure of the neural network results in high bias problem, while the opposite assumption leads to the high variance of the classifier and increases the computation time. The purpose is to find a compromise between these issues.

The evaluation of the classifier is based on the learning curves, which are the dependencies of the values of the cost function for the Train and Cross Validation sets from the number of training examples utilized in the learning process. The idealized learning curves of an ANN are schematically displayed in the Figure 3.22. The are two main issues that might appear in the system with classifier:

- **high bias** problem - the hypothesis of the classifier is oversimplified,
- **high variance** problem - the classifier performs well on the training data but does not generalize well on the previously unseen data.

In order to obtain the learning curves the face/non-face LBP histograms are split into two sets: Train set and Cross Validation (CV) set. The size M of the training set is iteratively incremented and the values of the cost function J are calculated for Training data set J_{train} and CV set J_{CV} after each epoch. In the case of high bias problem the values of J_{train} and J_{CV} are high and J_{CV} does not decrease when M is growing, see Figure 3.22 (a). If the gap between J_{train} and J_{CV} is significant and the value of J_{train} is small, then the system suffers from the high variance problem, Figure 3.22 (b). In the ideal case the J_{CV} decreases monotonically when M is growing and the final values of J_{train} and J_{CV} are small, Figure 3.22 (c).

Expansion of the feature space and extension of the complexity of the classifier are possible solutions of the high bias problem. The opposite activities resolves the high variance, however the better way is to add regularization ($\lambda > 0$) or increase the number of training examples M if possible.

The first sequence of experiments is performed with $K = 2$ (sliding window is divided into K^2 regions), then the length of the feature vector is $N = 64$. The number of neurons in the hidden layer s_{L-1} of the ANN is varied from 1 to 500. The gap between J_{train} and J_{CV}

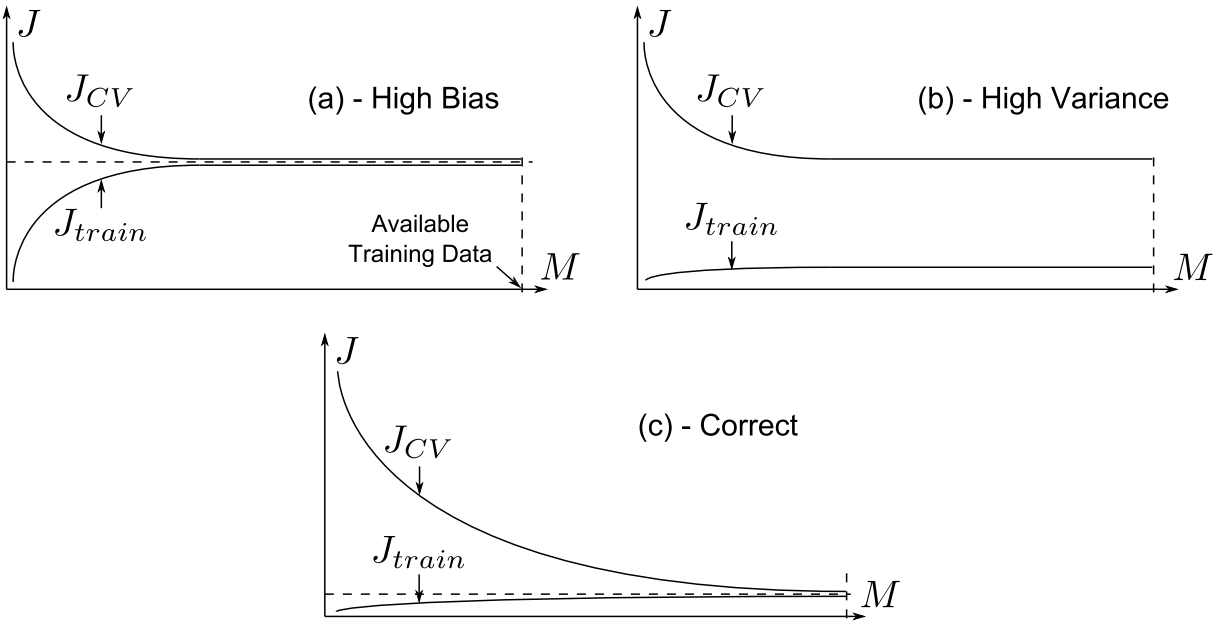


Figure 3.22: Idealized learning curves of an ANN for different problems in the classifier

curves is stabilized for the values $M > 2000$ and the value of $J_{train}(M > 2000)$ is relatively high regardless how many neurons are used in the hidden layer, Figure 3.23 (a). This observation indicates the *high bias* problem in the classifier. The augment of s_{L-1} does not improve the performance significantly, therefore the expansion of the feature space dimensionality is an appropriate choice in this situation. The value $K = 2$ is not considered for further processing.

For $K = 3$ and $K = 4$ the corresponding lengths of the feature vector are $N = 144$ and $N = 256$. In both cases the values of J_{train} are decreased significantly and the gap between J_{train} and J_{CV} curves decreases when M is growing, see Figure 3.23 (b) and (c). These K values are potentially a good choice for regioning factor, however evaluation of an ANN structure is needed in order to obtain high classification precision and computational efficiency.

The number of neurons in the hidden layer s_{L-1} is estimated according to the methodology described in [62]. The number of hidden neurons is iteratively incremented and the learning of an ANN is performed on the training set for each s_{L-1}. The value of the cost function for the cross-validation set J_{CV} is calculated next. The regularization parameter λ is set to zero at this stage. The dependencies J_{CV} for different s_{L-1} and K are plotted in the Figures 3.24 and 3.25. The curves $J_{CV}(s_{L-1})$ in Figures 3.24 and 3.25 are the average of 7 epochs of ANN training, the labels "o" represent the average value of J_{CV} and vertical bars stand for the value of standard deviation. The best choice of parameter s_{L-1} corresponds to the minimum value of J_{CV}. An explicit minimum of J_{CV} is observed for $s_{L-1} = 10$ both for $K = 3$ and $K = 4$.

The regularization parameter λ is evaluated next in the similar way. The number of hidden units in an ANN is set to $s_{L-1} = 10$. The dependencies $J_{CV}(\lambda)$ in Figures 3.26 and 3.27 are the average of 10 epochs of ANN training. The best choice of parameter λ corresponds to the minimum of J_{CV}, which is observed for $\lambda = 0$ both for $K = 3$ and $K = 4$.

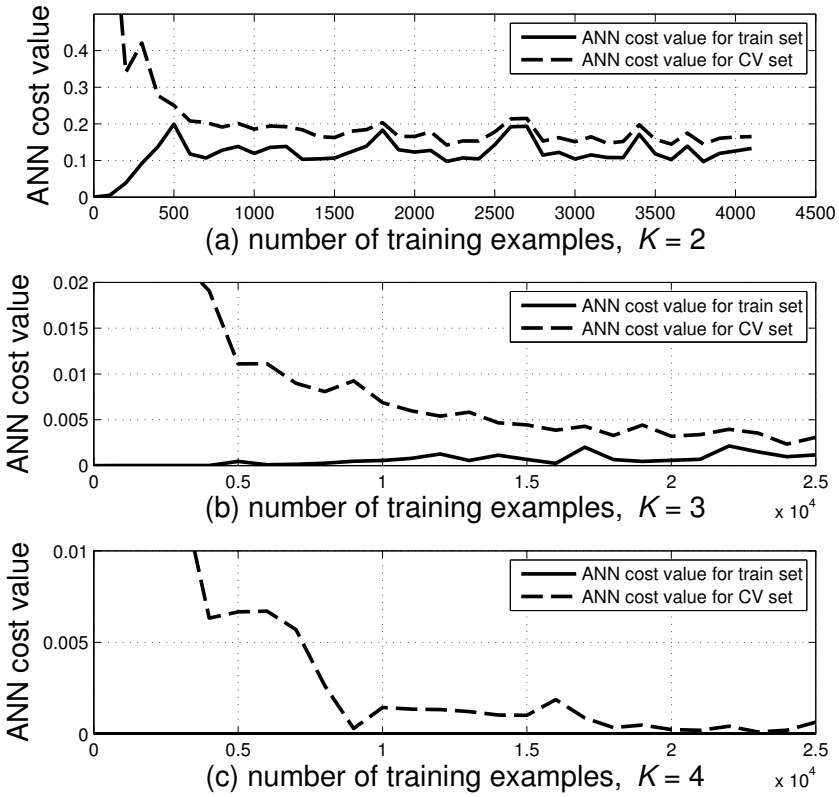


Figure 3.23: The learning curves for an ANN with $s_{L-1} = 5$ neurons in the hidden layer for different dimensionality of the feature space

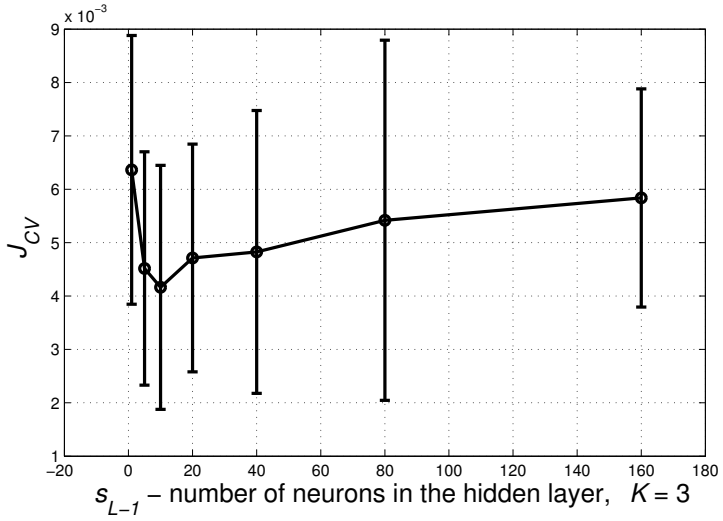


Figure 3.24: The dependence $J_{CV}(s_{L-1}, K = 3)$; vertical bars represent the value of standard deviation

Once the best values $s_{L-1} = 10$ and $\lambda = 0$ are determined both for $K = 3$ and $K = 4$ the algorithm is tested on all frontal face images of the color FERET database. The corresponding cumulative distribution functions of η_{face} are displayed in the Figures 3.28 and 3.29. The value of η_{face} is calculated according to the Equation (3.11) and the ground-truth coordinates of the eyes are determined according to the Equations (3.14). Two series of experiments are reported

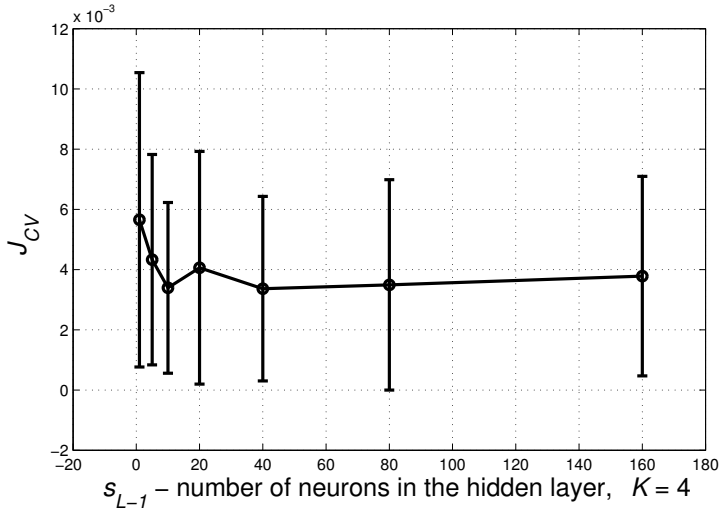


Figure 3.25: The dependence $J_{CV}(s_{L-1}, K = 4)$; vertical bars represent the value of standard deviation

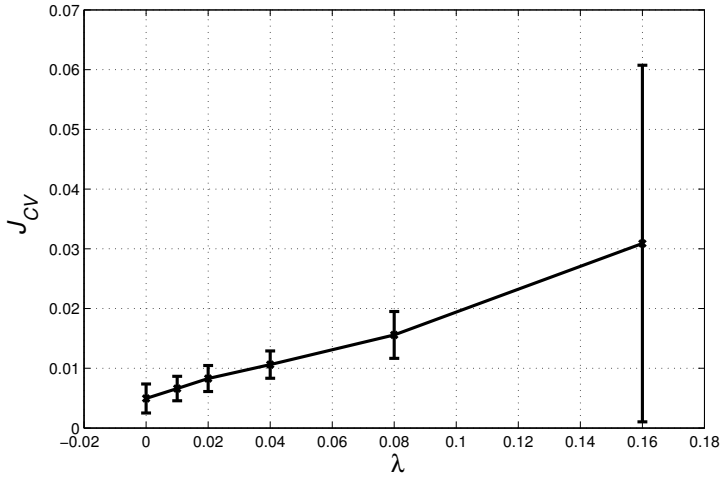


Figure 3.26: The dependence $J_{CV}(\lambda, K = 3)$; vertical bars represent the value of standard deviation

for each value of K:

- the size of the face s_{face} is determined by the algorithm; this scenario is always true in real life applications,
- the size of the face is known and only the position of the object is detected; this approach excludes the error contributed by incorrect size of the face.

The detection rates at $\eta_{face} = 0.25$ are determined from Figures 3.28 and 3.29:

$$P(\eta_{face} = 0.25, s_{face} - \text{unknown}, K = 3) = 76.4\%,$$
$$P(\eta_{face} = 0.25, s_{face} - \text{known}, K = 3) = 85.0\%,$$
$$P(\eta_{face} = 0.25, s_{face} - \text{unknown}, K = 4) = 94.2\%,$$
$$P(\eta_{face} = 0.25, s_{face} - \text{known}, K = 4) = 94.7\%.$$

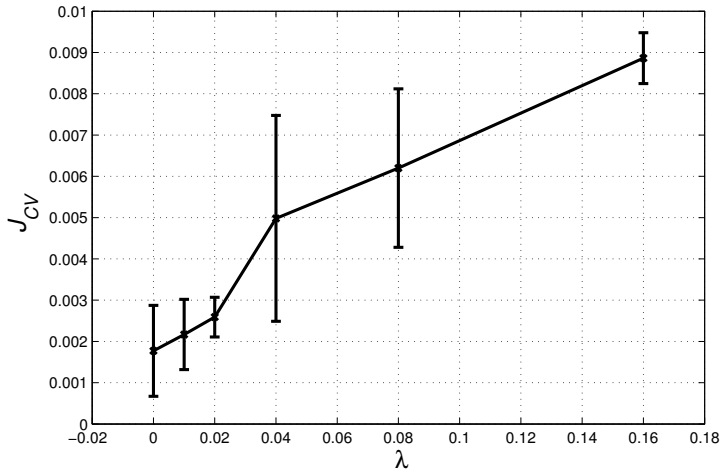


Figure 3.27: The dependence $J_{CV}(\lambda, K = 4)$; vertical bars represent the value of standard deviation

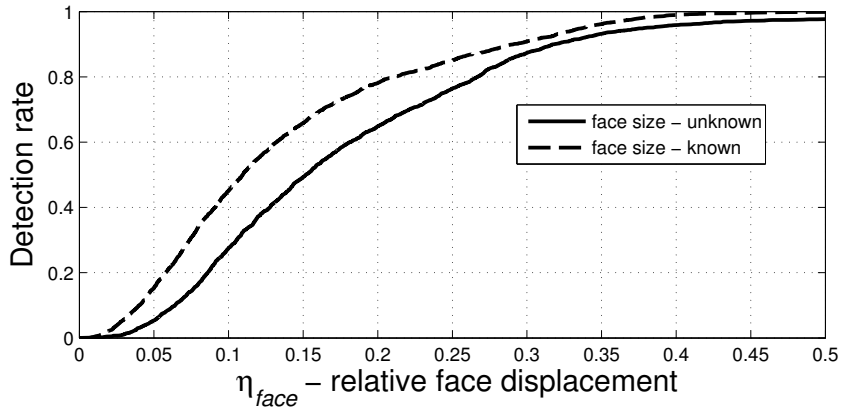


Figure 3.28: Cumulative distributions of η_{face} for LBP and ANN based face detection algorithm for $K = 3$; two cases are observed: s_{face} - known and s_{face} - unknown

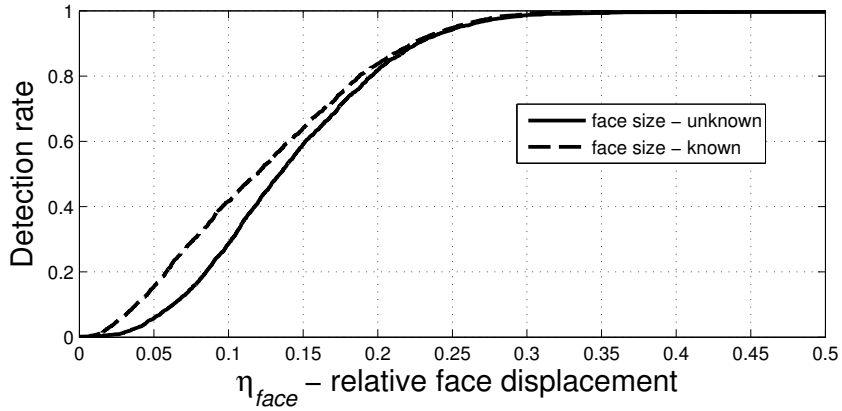


Figure 3.29: Cumulative distributions of η_{face} for LBP and ANN based face detection algorithm for $K = 4$; two cases are observed: s_{face} - known and s_{face} - unknown

The value of $K = 4$ clearly outperforms the experiments with $K = 3$ in terms of detection precision. The assumption that the size of the face is known did not improve the detection rate

significantly:

$$\Delta P(\eta_{face} = 0.25, K = 4) = P(s_{face} - \text{known}) - P(s_{face} - \text{unknown}) = 0.5\%$$

This observation means that regioning of detectable object with $K = 4$ also makes the algorithm robust against errors caused by incorrect size of the face.

3.6.4 Results for Support Vector Machine - based face detection

The first step of the SVM-based face detection algorithm is LBP transformation of the input image. The parameters of LBP operator are set to the values determined in the Section 3.6.1 in order to reduce the space of variables to be optimized. The main aspects of the proposed LBP and SVM based face detection algorithm to be evaluated in this section are:

- the regioning factor K of the regioning grid (Figure 2.2),
- the structure of the classifier, which is an SVM.

Similar to an ANN based face detection algorithm both of the above aspects are related and should be evaluated jointly.

The design of the SVM classifier is a challenging task and the resulting system should satisfy two main aspects:

- high classification precision, which is clearly needed for high discriminative power of the detector,
- small number of support vectors (SV), which has a direct impact on the execution time.

A compromise should be found between above conditions as they are partially mutually exclusive in case of nonlinearly separable data sets. The first assumption about the classifier selection is that it should be non-linear. This choice is based on the analysis of experimental results in previous sections, where the non-linear classifiers outperform the linear models, see Figures 3.24 and 3.25. In general, the RBF kernel is a reasonable choice for non-linear SVM [26], Equation (2.43). The first reason for that is the number of hyper-parameters which influences the complexity of model selection process. For example, the polynomial kernel has more hyper-parameters than the RBF kernel. Finally, the RBF kernel has fewer numerical difficulties. The value of the kernel lies in the range $(0, 1]$ in contrast to polynomial kernels of which values may be infinitely large for high degrees. The RBF kernel is not suitable when the number of features is very large, but low dimensionality of the feature space is one of our goals.

The two parameters to be adjusted for an RBF kernel are:

- $C > 0$ - the penalty of the error term,

- γ - determines the area of influence of the support vector over the data space, see Equation (2.43) for details.

The common strategy for the estimation of acceptable (C, γ) is to separate the data into three sets: training, cross-validation and test. The training set is utilized in the learning process while the cross-validation is needed to find the best parameters (C, γ). The final results are usually reported for the test set. In the case of SVM the accuracy measure for the CV set is the percentage of data which are correctly classified: P^{CV}. The cross-validation procedure can prevent the over-fitting problem.

A grid-search on C and γ using cross-validation set is selected in order to evaluate the model. Various pairs of (C, γ) values are tried and the one with high cross-validation accuracy and low number of SV is selected. Exponentially growing sequences of C and γ are utilized in order to identify good parameters:

$$\begin{aligned} C &= \{2^{\boldsymbol{n}^C}\}, \boldsymbol{n}^C = (-5, -3, -1, 1 \dots 25), \\ \gamma &= \{2^{\boldsymbol{n}^\gamma}\}, \boldsymbol{n}^\gamma = (-13, -11, -9, -7 \dots 11), \end{aligned} \tag{3.15}$$

where the values of degree n^C and n^γ are iteratively increased by a step of 2.

A grid-search along the space of parameters C and γ is a time consuming operation, therefore the sizes of train and cross-validation sets are reduced: $M_{train} = 4000$ and $M_{CV} = 4000$. Once the best parameters of (C, γ) are selected the training of SVM is performed with a complete training set.

The first sequence of experiments is performed with the regioning factor $K = 3$. For each pair of parameters (n_i^γ, n_j^C) , Equation (3.15), the corresponding accuracy measure for the CV set $P_{i,j}^{CV}$ and number of support vectors $N_{i,j}^{SV}$ are calculated. These values are substituted into the matrices:

$$\begin{aligned} \boldsymbol{P}^{CV} &= \{P_{i,j}^{CV}\},\ i = 1, \dots, 13;\ j = 1, \dots, 16, \\ \boldsymbol{N}^{SV} &= \{N_{i,j}^{SV}\},\ i = 1, \dots, 13;\ j = 1, \dots, 16, \end{aligned}$$

where the number of elements in vector $\boldsymbol{n}^\gamma$ is 13 and in vector $\boldsymbol{n}^C$ is 16.

The matrices $\boldsymbol{P}^{CV}$ and $\boldsymbol{N}^{SV}$ with $K = 3$ are displayed in the Figure 3.30. The maximum accuracy on the cross-validation set $\max(\boldsymbol{P}^{CV})$ and the minimum number of support vectors $\min(\boldsymbol{N}^{SV})$ are plotted with a circle markers "o" in Figure 3.30. The ideal scenario is to select parameters (n_i^γ, n_j^C) corresponding to the point of intersection between segments $\max(\boldsymbol{P}^{CV})$ and $\min(\boldsymbol{N}^{SV})$, however this is not the case due to an absence of intersection. The compromise between the values of $\boldsymbol{P}^{CV}$ and $\boldsymbol{N}^{SV}$ is needed.

For this purpose the columns of matrix $\boldsymbol{P}^{CV}$ are stacked into a single vector, which is then sorted:

$$[\boldsymbol{p}^{CV}, \boldsymbol{idx}] = \text{sort}(\{\boldsymbol{P}_{1:13,1}^{CV}; \boldsymbol{P}_{1:13,2}^{CV}; \dots; \boldsymbol{P}_{1:13,16}^{CV}\}), \tag{3.16}$$

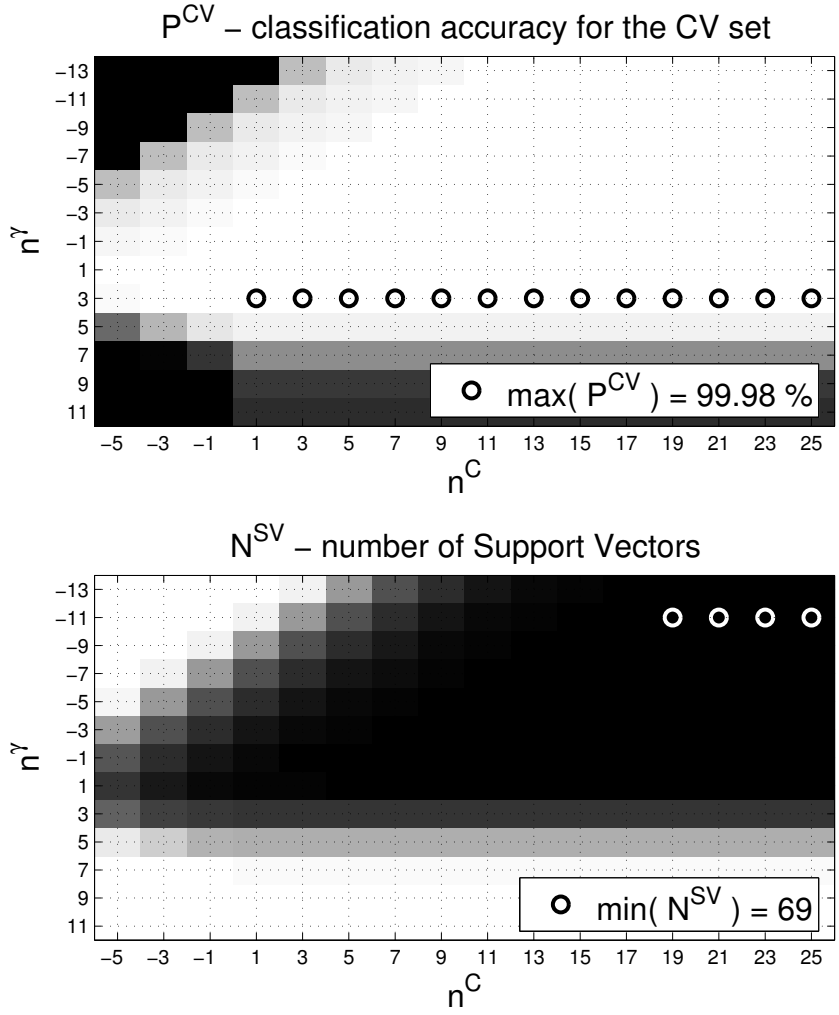


Figure 3.30: Full range images of matrices $\boldsymbol{P^{CV}}$ and $\boldsymbol{N}^{SV}$ with regioning parameter $K = 3$; $M_{train} = 4000$ and $M_{CV} = 4000$

where the elements of a vector $\boldsymbol{p}^{CV}$ are arranged in ascending order and $\boldsymbol{idx}$ is a vector of permutation indexes. The size of sorted vector is now $\boldsymbol{p}^{CV} \in \mathbb{R}^{13 \cdot 16}$.

Next, the matrix $\boldsymbol{N}^{SV}$ is vectorized and elements of resulting vector are permuted according to indexes in vector $\boldsymbol{idx}$:

$$\begin{aligned} \boldsymbol{n}^{SV} &= \{\boldsymbol{N}^{SV}_{1:13,1}; \boldsymbol{N}^{SV}_{1:13,2}; \ldots; \boldsymbol{N}^{SV}_{1:13,16}\}, \\ \boldsymbol{n}^{SV} &\leftarrow \boldsymbol{n}^{SV}_{\boldsymbol{idx}}, \end{aligned} \tag{3.17}$$

where the last equation means permutation: the elements of a vector $\boldsymbol{n}^{SV}$ are updated ($\leftarrow$) with entries specified by indexes $\boldsymbol{idx}$.

The last 100 elements of vectors $\boldsymbol{p}^{CV}$ and $\boldsymbol{n}^{SV}$ and corresponding $\boldsymbol{n}^{C}$ and $\boldsymbol{n}^{\gamma}$ values are plotted in Figure 3.31. The region of interest (ROI) in Figure 3.31 is located between two black vertical lines. The classification accuracy in ROI is still high $P^{CV}_{ROI} = 99.92\%$ while the number of support vectors is relatively small $N^{SV}_{ROI} = [73, 151]$. The smallest number of support vectors

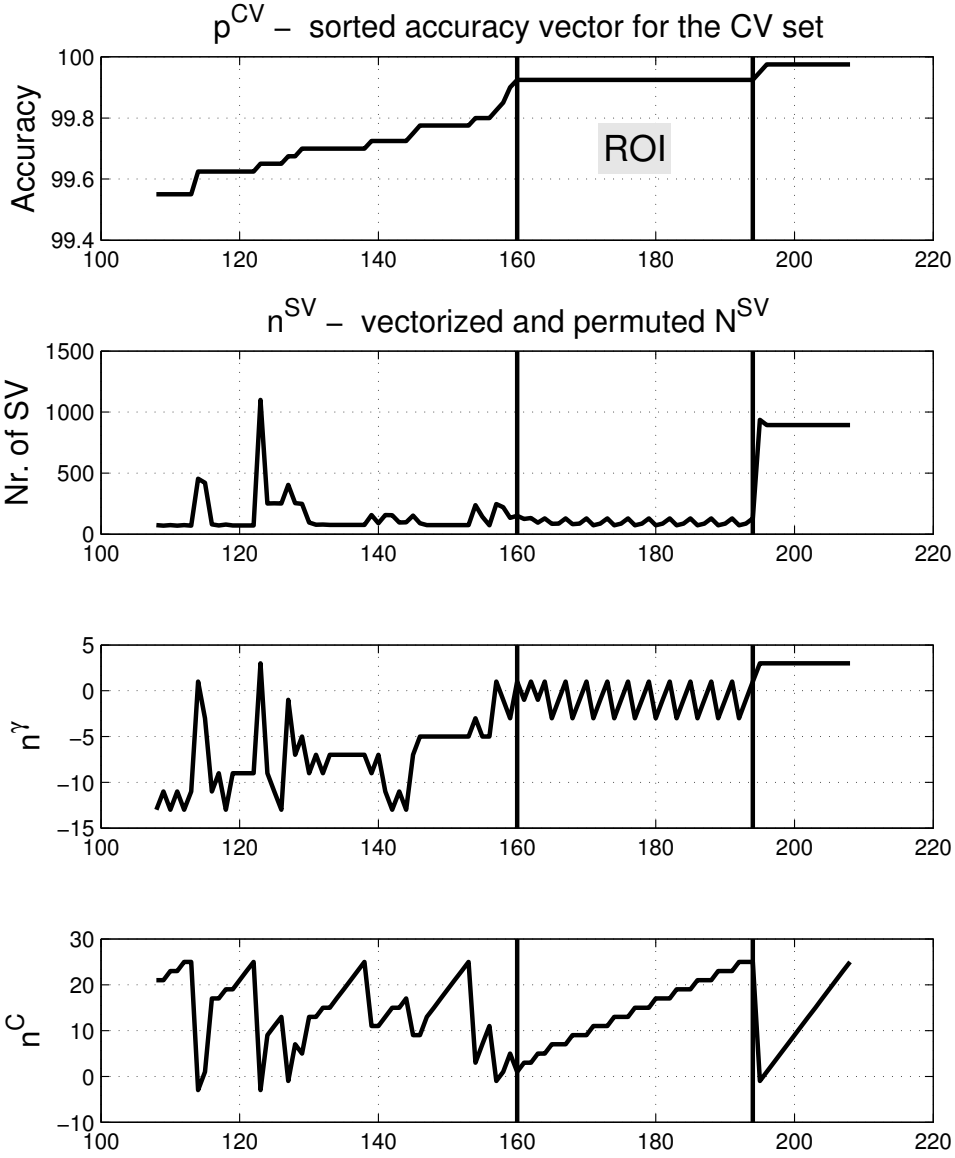


Figure 3.31: Sorted accuracy vector $\boldsymbol{p}^{CV}$ and corresponding $\boldsymbol{n}^{SV}$, $\boldsymbol{n}^{C}$ and $\boldsymbol{n}^{\gamma}$ for $K = 3$

is obtained for learning parameters:

$$n^C = 11, \;\; n^\gamma = -3.$$

Once the best learning parameters $(C = 2^{11}, \gamma = 2^{-3})$ are selected for $K = 3$ the training of SVM is performed with a complete training set. The resulting number of support vectors after the training on the complete data set is $N^{SV}(K = 3) = 115$. The number of support vectors is obviously increased from 73 to 115, but is still in the acceptable range.

The same evaluation process is repeated for $K = 4$. The matrices $\boldsymbol{P}^{CV}$ and $\boldsymbol{N}^{SV}$ for $K = 4$ are displayed in the Figure 3.32.

The last 100 elements of vectors $\boldsymbol{p}^{CV}$ and $\boldsymbol{n}^{SV}$ and corresponding $\boldsymbol{n}^{C}$ and $\boldsymbol{n}^{\gamma}$ values for $K = 4$ are plotted in Figure 3.33. The region of interest (ROI) in Figure 3.31 is located between

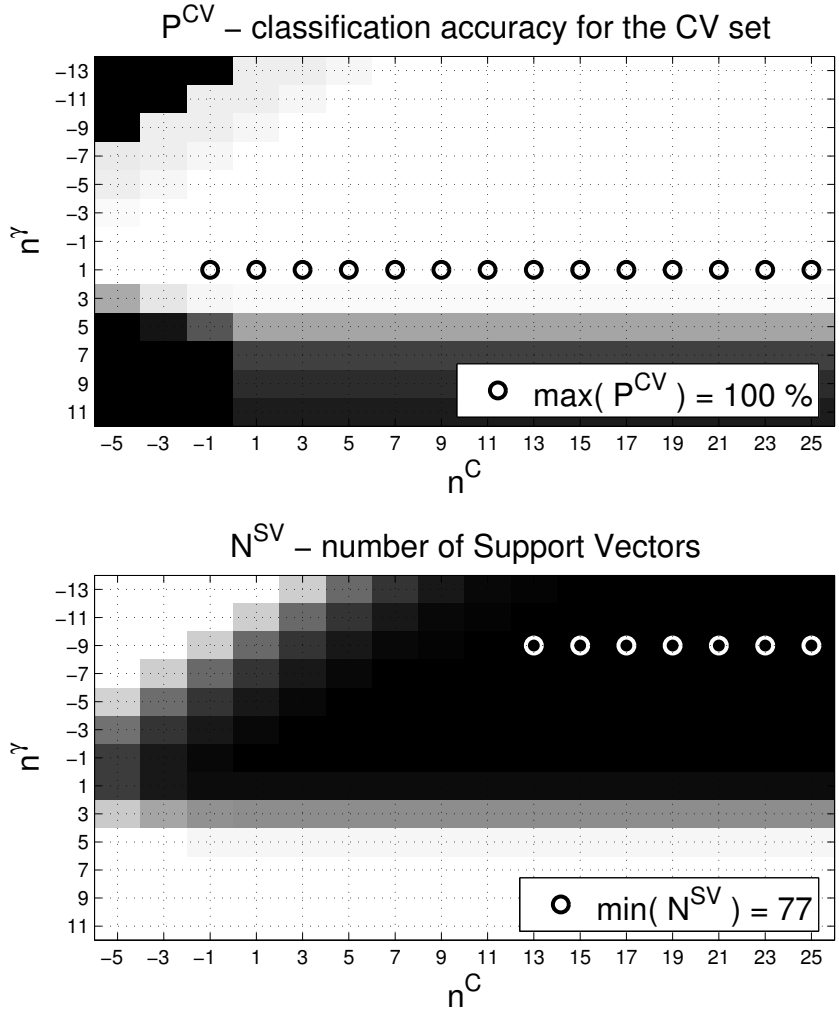


Figure 3.32: Full range images of matrices $\boldsymbol{P^{CV}}$ and $\boldsymbol{N^{SV}}$ with regioning parameter $K = 4$; $M_{train} = 4000$ and $M_{CV} = 4000$

two black vertical lines. The classification accuracy in ROI is still high $P^{CV}_{ROI} = 99.85\%$ while the number of support vectors is smallest possible $N^{SV}_{ROI} = 77$. The corresponding learning parameters: $n^C = 13,\ n^\gamma = -9$.

Once the best learning parameters $(C = 2^{13}, \gamma = 2^{-9})$ are selected for $K = 4$ the training of SVM is performed with a complete training set. The resulting number of support vectors after the training on the complete data set is $N^{SV}(K = 4) = 124$.

The evaluation of the SVM learning parameters is now completed both for $K = 3$ and $K = 4$ and the algorithm is tested on all frontal face images of the color FERET database. The corresponding cumulative distribution functions of η_{face} are displayed in the Figures 3.34 and 3.35. The value of η_{face} is calculated according to the Equation (3.11) and the ground-truth coordinates of the eyes are determined according to the Equations (3.14). Two series of experiments are reported for each value of K:

- the size of the face s_{face} is determined by the algorithm; this scenario is always true in real life applications,

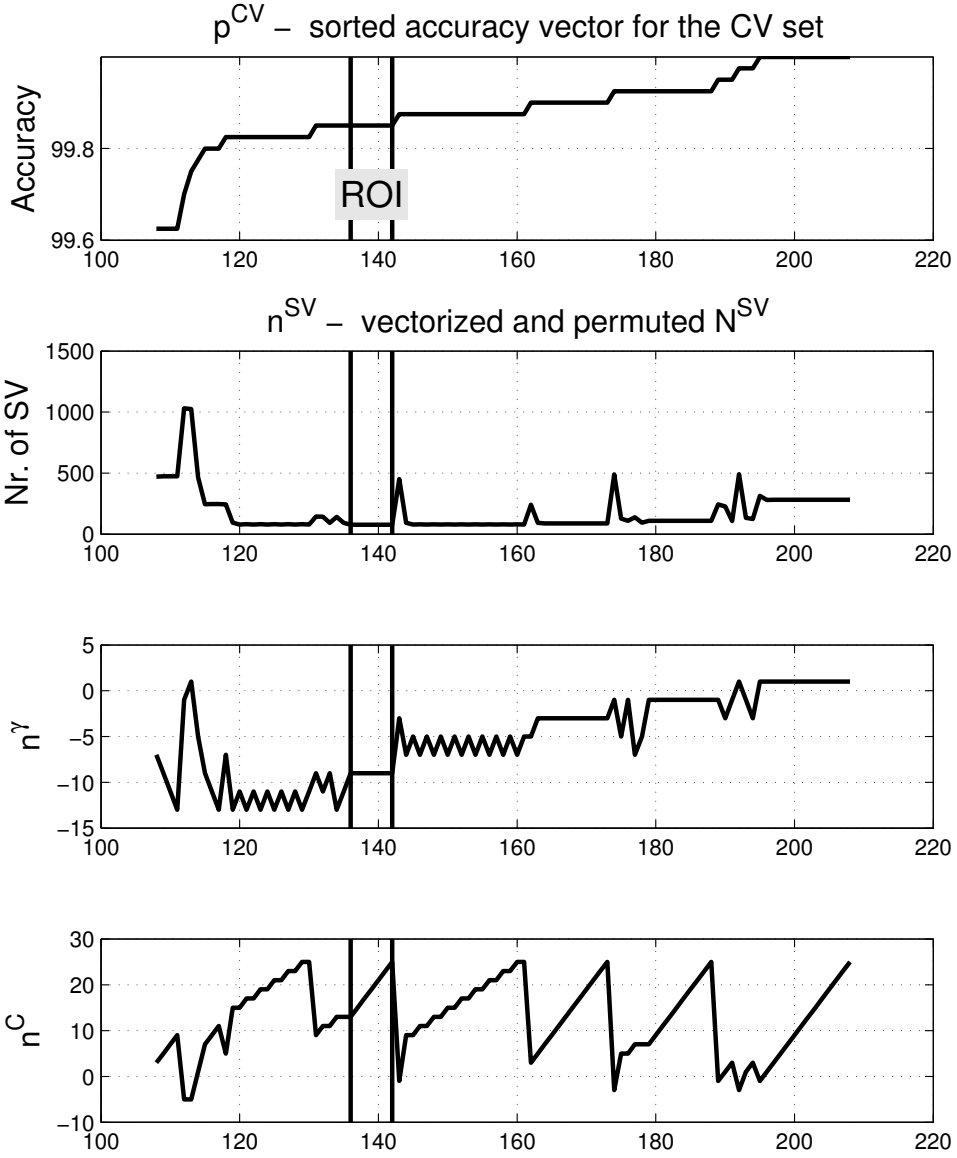


Figure 3.33: Sorted accuracy vector $\boldsymbol{p}^{CV}$ and corresponding $\boldsymbol{n}^{SV}$, $\boldsymbol{n}^{C}$ and $\boldsymbol{n}^{\gamma}$ for $K = 4$

- the size of the face is known and only the position of the object is detected; this approach excludes the error contributed by incorrect size of the face.

The detection rates at $\eta_{face} = 0.25$ are determined from Figures 3.34 and 3.35:

$$P(\eta_{face} = 0.25, s_{face} - \text{unknown}, K = 3) = 76.2\%,$$
$$P(\eta_{face} = 0.25, s_{face} - \text{known}, K = 3) = 78.7\%,$$
$$P(\eta_{face} = 0.25, s_{face} - \text{unknown}, K = 4) = 98.2\%,$$
$$P(\eta_{face} = 0.25, s_{face} - \text{known}, K = 4) = 99.7\%.$$

The value of $K = 4$ clearly outperforms the experiments with $K = 3$ in terms of detection precision. The assumption that the size of the face is known slightly improved the detection

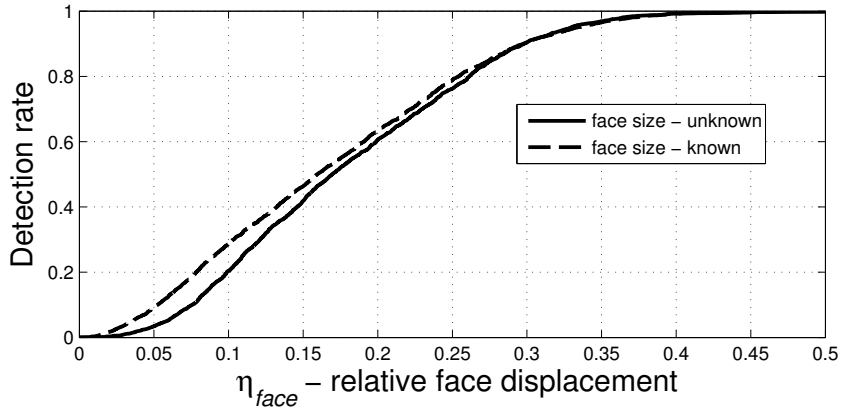


Figure 3.34: Cumulative distributions of η_{face} for LBP and SVM based face detection algorithm for $K = 3$; two cases are observed: s_{face} - known and s_{face} - unknown

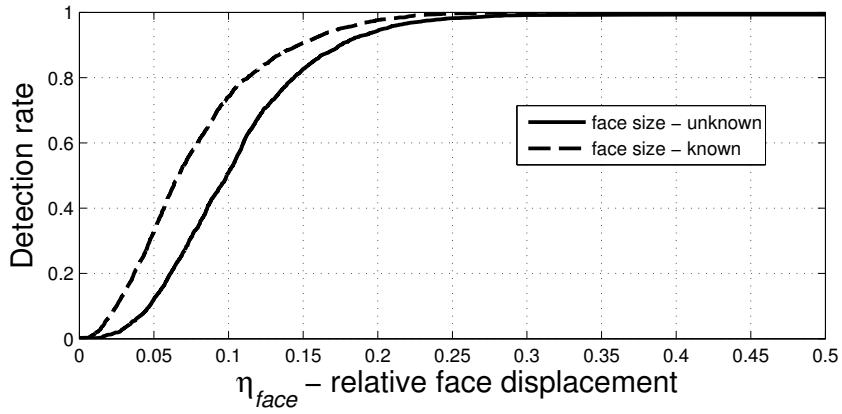


Figure 3.35: Cumulative distributions of η_{face} for LBP and SVM based face detection algorithm for $K = 4$; two cases are observed: s_{face} - known and s_{face} - unknown

rate:

$$\Delta P(\eta_{face} = 0.25, K = 4) = P(s_{face} - \text{known}) - P(s_{face} - \text{unknown}) = 1.5\%$$

This observation means that more sophisticated merging technique is the primary field of improvement in the LBP and SVM based face detection algorithm, because other aspects conduct only 0.3% of error.

3.6.5 Comparison of face detection algorithms

In this section a comparison of the proposed algorithms with one of the state-of-the-art face detection techniques is proposed. The comparative study of the face detection algorithms is a challenging task due to some reasons, which are briefly summarized here:

- The databases utilized in the experiments may differ. One of the most popular databases in the field of face detection is MIT+CMU. However this dataset is not suitable for our

Table 3.1:
Comparison of face detection algorithms

Method:	Parameters:	P($\eta_{face} \leq 0.25$)	Detection time (seconds)
LBP+SVM	$K = 5$, $N^{SV} = 144$	99.7%	0.67
Haar-like features [117]		99.5%	0.02
LBP+SVM	$K = 4$, $N^{SV} = 124$	98.2%	0.45
LBP+ANN	$K = 4$, $s_{L-1} = 10$	86.9%	0.22

experiments due to relatively low resolution of facial images. The proposed algorithms are optimized to operate with facial images of the sizes 150×150 and up.

- The separation of available data into the train and the test sets may be different or even unclear. The random principle for data segregation is sometimes utilized, which makes the experiments unrepeatable.

- The methodology for the evaluation of face detection algorithms is not standardized, thus the criteria in the performance analysis may differ.

The scope of experiments in this research is limited to:

- color FERET database due to its popularity among face recognition researchers,

- localization task, because it is often relevant for biometric systems.

Of all the face detection algorithms currently in use, the one introduced in [117] is the best known and most widely used. For this reason this algorithm is selected for comparison to the proposed localization methodologies. The open source implementation of the algorithm from OpenCV [19] library is utilized in experiments. In order to make the comparison correct the experimental conditions are exactly the same as the ones described in sections 3.4 and 3.5. Both the size and the location of the face are determined by the algorithm. The detection rates at $\eta_{face} = 0.25$ and average detection times are summarized in Table 3.1. The detection time of a single face in the images of color FERET database may vary from image to image, which is especially expressed for Haar-based face detection algorithm. Thus the detection times in Table 3.1 are an average of face detection times for all frontal images in the color FERET database. The algorithms were tested on the same computer with I5-2500K processor and 4GB of RAM, however the proposed LBP-based algorithms are implemented in Matlab, which by default is slower then C/C++ implementation of the Haar-based face detection algorithm in OpenCV library.

The detection rate of the proposed face detection approach which is based on the LBP and SVM combination slightly outperforms the Haar feature based algorithm, see Table 3.1 for details.

3.6.6 Conclusions

In this chapter an overview of significant face detection approaches, including previous LBP-based detection techniques, are reviewed. A novel cluster of LBP-based face detection algorithms is proposed next. The advantage of this methodology is the flexibility of the algorithm, which allows to adjust the trade-off between the dimensionality of the feature space and the complexity of the classifier. Another positive moment is the absence of the down-sampling stage, which is often incorporated in the detection algorithms in order to localize object of various scales. The proposed methods are tested in terms of localization performance. The precision which is comparable to state-of-the-art algorithms (Table 3.1) is obtained in low-dimensional feature space (several hundreds of features) and with simple classifier (Artificial Neural Network with 10 Neurons in the hidden layer or Support Vector Machine (SVM) with 100-200 Support Vectors). While the localization precision is high enough, the computational time is in range of seconds and is still a challenging issue for the proposed approach. This fact can be explained by the absence of special techniques for the reduction of the number of scanning positions. Some of possible improvements of scanning process are described below:

- *Adjustment of scanning parameters*. Such factors as the step of the sliding window or the number of expected scales of the detectable object significantly impacts the computational time, however it is also affects the localization accuracy. The empirical knowledge about the task and physical setup of the system can help to find the desired trade-off.

- *Reduction of the search region*. Fast preprocessing methods can effectively discard sub-windows before computationally expensive processing by the classifier. These methods are usually based on color, variance or texture analysis.

- *Adaptive adjustment of scanning parameters*. The adjustment of scanning parameters is possible based on the confidence scores in the previous scanning positions.

- *Task specific optimizations*. The biometric systems are often designed to operate in localization mode, thus only one face is present in the input image. Additionally, the user is usually interested to cooperate with the system. Based on this information the optimal selection of the starting scanning position and of the scale of the sliding window is possible.

- *Software and hardware level optimizations*. The source code and compilation optimizations also affect the execution time. The proposed detection methodologies are histogram-based, thus the effective parallelization of the process is possible. However parallelization requires special hardware solutions, such as multi-core DSP, FPGA or graphical cards.

Above methodologies can significantly speed up the introduced detectors, however these aspects are out of the scope of this research.

Chapter 4

EYE LOCALIZATION - BASED FACE ALIGNMENT

The knowledge of the precise position of eye regions and eye pupils is an important aspect in many applications, such as face recognition, iris recognition, eye-tracking, medicine, advertising sector, human computer interaction, digital photography and others. The resulting performance of the above systems depends on the precision of the eye localization stage. Eye localization is the second module in the automatic face recognition system, which is schematically introduced in Figure 1.1. It is shown in literature that the face alignment, which is usually based on eye detection, has a large impact on recognition accuracy [74], [93], [22]. The term eye detection is more general than localization. In the detection task the number of desired objects is *unknown*, while the localization principle implies that the number of detectable objects is *known*. Proposed automatic face recognition system processes one facial image at a time, therefore the number of detectable objects (two eyes) is known and the the task is limited to eye localization only. The eye localization module is designed to deal with frontal face images, because the scenario of voluntary cooperation between the user and the system is investigated.

The eye localization task can be viewed as a special case of the *detection of object class*. The purpose is to find the locations and the sizes of the objects in the image, which belong to the specific class. However, in the case of eye localization, the problem is easier, because some information about the input image is known. The input image is obviously a face which greatly reduces the diversity of the non-eye class. Also the size of the detectable object can be specified as a fraction of the size of the face which is determined in the face detection stage. Thus, the number of scans of the input facial image with the sliding window is one. Additionally, the empirical knowledge about the face can be used as a supporting information in the eye localization task, because at a given resolution, whole faces contain more information than the eyes alone.

The amount of algorithms in the field of eye localization is significant. The eye localization solutions combine various approaches which are based both on the principles described in the face detection section and some unique methods which utilize empirical aspects of the task.

Similar to the face detection, eye localization can be divided into two main groups: video - based eye localization and eye localization in digital images. In video sequence the time domain is present, which is used in localization tasks in different manners. In digital images only the spatial and quantitative information about the scene is introduced. The scope of this research is limited to the task of eye localization in digital images.

4.1 Related work

Many eye detection algorithms are proposed in scientific papers in the last decades. Eye detection methods can be divided into two categories based on the physical aspects of the imaging device: active and passive eye detectors [53]. The active detection methods usually use IR illumination and utilize physical properties of the eyes in this spectrum [43] and [135]. The advantages of active methods are that they are very accurate and robust, but the negative property is that they need special lighting sources and have poor precision in an outdoor environment [119]. Passive methods detect eyes from images within visible light spectrum and normal illumination. The scope of this research is limited to passive methods.

Similar to the face detection taxonomy of [124] the passive eye localization methods can be divided into two main categories: *template-based* and *appearance-based* eye detectors. The *feature-based* eye localization approach is not popular due to insufficient statistical information about eye features in regular images.

Template-based approaches are robust in a wide range of pose and expression variability. Probably most widely-used techniques in the field of template based eye localization are *deformable face models* [77].

Deformable face model-based face analysis is a popular paradigm with a wide spectrum of applications. Researchers and industry utilize this principle in facial recognition, emotion recognition, head pose estimation and tracking, lip reading, medical analysis and *face feature detection*. The first step of the algorithms is to construct a face model from a training set. The training set consists of 2D images or 3D face scans. The face model is then fit to the input image by adjusting the parameters of the model. These parameters are then used in whatever the application is. Perhaps the most well known face models are 2D Active Appearance Models (AAM) [27] and 3D Morphable Models (3DMM) [15]. AAMs and 3DMMs are rather similar. Both consist of a linear shape model and a linear appearance (texture) model [77]. The main difference between them is the shape component which is 2D for an AAM and 3D for 3DMM. 3D models are, in general, preferable to 2D models, since they have more compact parametrization, more robust fitting and requires less iterations to converge [77].

The next cluster of template-based eye localization methods utilizes the *geometrical features* of the eye. The first obvious observation about the eye is the circular shape of the pupil. The eye localization algorithms which are based on Hough transform for circle detection are introduced in [86] (Nikisins et al.) and [35]. However this algorithm has an obvious limitation in the

case of half-opened eyes when the pupils are partially covered. The Discriminative Generalized Hough transform is utilized in [42] in order to overcome the limitations by using the well-defined shape information about the whole eye. Another geometrical approach is proposed in [113], where authors use isophote curvatures for eye localization and tracking. However, the accuracy drops significantly in the presence of large head rotations. This is due to the fact of the lost symmetry and thus the algorithm delivers increasingly poor performance. This observation is later corrected by the authors in [114].

The *pictorial structures* for object recognition are introduced in [38]. The scope of the paper is broader than eye localization, however the locations of the eyes can be easily extracted as a particular result of the research. The face is represented by a collection of parts arranged in a deformable configuration. Each part stores local visual properties of the object. The deformable configuration is characterized by spring-like connections between certain pairs of parts. The best alignment of such a model to an input image is found by minimizing an energy function that measures both a match cost for each part and a deformation cost for each pair of connected parts. This research is later extended and applied to the eye localization task in [109].

A wide cluster of object detection methods is based on *advanced correlation filters* [59]. They have been receiving increasing attention in recent years due to their mathematical simplicity and computational efficiency. The pattern of interest in the input image is searched for by cross-correlating the input image with one or more templates and examining the resulting correlation plane for correlation peaks. The main issue in this approach is a proper design of the templates. Popular examples of correlation filters include Minimum Average Correlation Energy Filters [72], Distance Classifier Correlation Filters [71], Unconstrained Minimum Average Correlation Energy Filters [99], Average of Synthetic Exact Filters [17] and Principal Directions of Synthetic Exact Filters [105]. The authors in [105] utilize the principles of advanced correlation filters for the eye localization task and compare the performance of the most popular filters in terms of localization precision and execution time.

Appearance-based approaches perform scanning of the input image with a small overlapping windows with the purpose of searching the most likely eye candidates. Most of the modern appearance based approaches rely on statistical classifiers, which are optimized using the sets of labeled eye and non-eye training examples. The block-diagram of the appearance-based eye detection system is displayed in Figure 4.1.

Similar to the face detection task the concept of a *sliding window* is the key idea of appearance based eye-localization methods. The difference with the face detection is that the size of the sliding window is usually of a constant size, which is specified by the size of the input face image. The rough size of the eyes are estimated using anthropometric relations. The downsampling of the input image might also be incorporated into the algorithm in order to speed up the calculations. However the downsampling causes the loss of statistical information about the object of interest thus the performance of the detector usually degrades. Preprocessing of the subwindows is sometimes performed before the classification stage. Preprocessing might

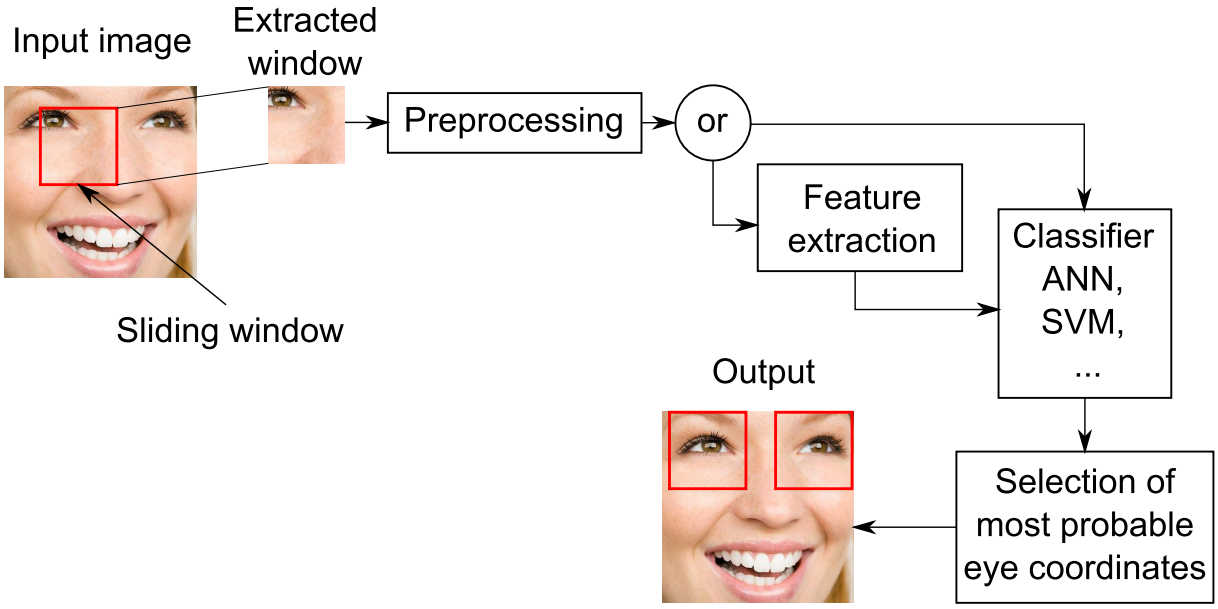


Figure 4.1: The block-diagram of appearance-based eye localization system with a sliding window concepts (semantically similar to [97])

include the subtraction of linear gradient function, histogram equalization, geometrical transformations and other techniques, which are usually addressed to resolve the problem of variable lighting conditions and variations in head poses. The classifier is the final block of the eye localization system, which rates the input pattern as either *eye* or *non-eye*. The classifier processes either the pattern in the sub-window or the features that are extracted from it. The appearance-based methods mainly differ in the choice of preprocessing steps, selected features and classifier. Some of the most significant approaches are discussed below.

The most popular appearance-based eye localization approaches are based on the principles introduced in [118]. The authors introduced the *boosting* technique to the computer vision problem, which involves training a series of simple classifiers with increasing discriminative power and then combining their outputs. The hypothesis of the boosting classifier is described in the Equation (3.1).

One of the first significant papers in the field of a boosting-based eye localization is introduced in [119]. Statistical discriminative features are learned in [119] to characterize eye patterns. Then the probabilistic classifiers are learned to separate eyes and non-eyes. Multiple simple classifiers are then combined in AdaBoost to form a robust detector. The provided experimental results on the challenging FRGC 1.0 database [92] show high localization accuracy. Additionally authors analyze the performance of the face recognition system which based on the proposed automatic eye localization. Experiments demonstrate that the proposed eye localization method can be incorporated into a fully automatic face recognition system.

Later an extension to AdaBoost classifier namely 2D Cascaded AdaBoost is introduced in [87] with application to the eye localization task. Original AdaBoost classifier using bootstrap on the negative examples only. The method in [87] bootstraps both the negative and positive

examples in a similar way. 2D structure provides a number of advantages: facilitates the training on large-scale datasets; easily deals with the significant variations within the class; accelerates the training and testing procedures. Experimental results on four public face databases verified the effectiveness of the introduced method.

The effectiveness of AdaBoost eye localization in mobile devices is described in [41] where authors implemented the algorithm in Nokia N90 mobile phone.

A transition form boosting to SVM classifier is proposed in [69]. The classification approach is constructed as follows: the Adaboost is first used for feature selection instead of being a classifier. A subset of 100 pixel-pattern-based texture features (PPBTF) [125] is selected by Adaboost. SVM classifiers with RBF kernel function are then trained on the features selected by AdaBoost. The experimental results on FERET database demonstrated real time operation.

An overview of appearance-based eye localization approaches with various classifiers is presented in [37]. Three approaches to the task are investigated: a regression approach aiming to directly minimize errors in the predicted eye positions, a simple Bayesian model of eye and non-eye appearance, and a discriminative eye detector trained using AdaBoost. The advantage of the paper is the usage of identical images in the training and testing stages that allows unbiased comparison of the proposed techniques.

4.2 Eye localization using Local Binary Patterns

Local Binary Pattern operator was originally introduced as a texture descriptor, but the discriminative power and computational simplicity of the LBP enhanced the scope of the operator in many computer vision fields. Some of the most significant directions are face detection and face recognition. An obvious extension of the LBP-based object detection field is eye localization task, however this field has not been studied enough. This fact can be explained with a lack of statistical information about the detectable object. The resolution of the eyes in face images is usually not high and the LBP histograms, that are often used as descriptors, might be unstable. The number of papers in this field is very limited, but some of the most significant results are discussed below.

Authors in [73] addressed the problem of locating facial features in frontal face images under different lighting conditions. The research is based on well-known Active Shape Model (ASM) method proposed in [28]. Authors introduced the use of Local Binary Patterns (LBP) for the improvement of the robustness of ASM to illumination changes. In original ASM the shape of a face is represented by a set of landmark points and the landmarks itself are described with local gray-level structures. These structures are called local appearance models. In [73] the landmarks are the centers of a squared regions. The square is divided into four regions from which the LBP histograms are extracted and concatenated into a single feature histogram representing the local appearance models. Introduced experiments performed on the standard and darkened image sets of the XM2VTS database [79] demonstrate that LBP-ASM approach gives superior performance

compared to the state-of-the-art ASM.

Later inspired by the ideas of Viola and Jones about Haar-like features (Figure 3.2) the authors in [94] developed a robust approach for eye detection using Haar-like features extracted from LBP images. The images are first preprocessed by LBP operator with parameters $(P = 8, R = 1)$ and then the Haar-like features are extracted and utilized in AdaBoost for designing a cascade classifier. The training of the classifier is based on the bootstrap strategy. Thus, the system is first trained with eye and random non-eye examples and after the run of the eye detector the training set is updated with all those non-eye patterns that were wrongly classified. Additionally, authors considered negative training samples extracted also from the facial regions because it has been shown that this can enhance the performance of the system. The experimental results with LBP and Haar-like combination for eye localization outperforms the Haar-like features alone.

In [85] (Nikisins et al.) an appearance-based eye localization algorithm which combines the LBP and NNC was introduced. An LBP transformed image of facial region is iteratively scanned with a sliding window of the size equal to the expected eye dimensions. At each position of the window the squared Euclidean distance between the LBP histogram of the corresponding region and the histogram of eye model is calculated. These distances are then substituted into a single distance matrix. The minimums in a distance matrix refer to the most probable locations of the eyes. The approach in [85] (Nikisins et al.) also takes into consideration the empirical information about the face: interocular distance and angle between the eye - line and the horizontal line. This information is utilized as a support data in order to exclude mis-detections.

The idea of [85] (Nikisins et al.) is later enhanced in [83] (Nikisins et al.). Authors utilized the spatially enhanced LBP histograms and an ANN classifier for eye localization. The regioning of detectable object improved the localization precision. Introduced optimization principles and discriminative power of LBP significantly reduced the number of features used to describe the classes in comparison to [85] (Nikisins et al.). The parameters of LBP operator are $(P = 4, R = 4)$, thus the length of the LBP histogram of each region is equal to 16. The experimental results show that an ANN with only 5 neurons in the hidden layer and spatially enhanced LBP histograms with regioning parameter $K = 2$ is a good compromise for high localization precision and computational efficiency.

4.2.1 Artificial Neural Network - based eye localization

The idea to use the combination of Local Binary Patterns and Artificial Neural Network for the eye detection problem is proposed in [83] (Nikisins et al.).

The first stage of the algorithm, that is the detection of eye regions, belongs to appearance-based approaches, see Figure 4.1. An important aspect of eye localization task, in contrast to face detection, is the scanning of the input image with only one size of the sliding window. In this research the size of the eye is considered to be equal to a constant fraction of the face size.

Similar to face detection the down-sampling of the input facial image is absent, and therefore statistical data about the object is not lost on a very first stage of the algorithm.

The LBP and ANN based eye localization task can be divided into two steps:

- *Localization of eye regions* - in this stage the squared eye regions are detected in the face image. The center of the squared region is an approximate position of the eye.
- *Localization of eye pupils* in the eye images - in this step the centers of eye pupils are detected. Eye pupils are considered to be a good reference points in the face.

Localization of eye regions is discussed first. The general structure of the LBP and ANN based eye localization algorithm is schematically displayed in the Figure 4.2. The first step of the algorithm is to calculate the LBP transformation of the input facial image, Figure 4.2 (2). The LBP transformed image is scanned with the sliding window of the size equal to the expected eye dimensions s_{eye} (the size of the eye is considered to be equal to a constant fraction of the face size):

$$s_{eye} = 0.8 \cdot d_{eye} = 0.4 \cdot s_{face}. \tag{4.1}$$

At each position of the sliding window the representation of the object is calculated. The spatially enhanced histogram is selected as the representative feature vector, see section 2.1.1 for details. Authors in [85] (Nikisins et al.) compute an ordinary histogram of the LBP image in the sliding window as a representation of the pattern. Such approach can be viewed as a particular case of the spatially enhanced histogram with $K = 1$. However, in this case the spatial information about the object is completely lost and therefore the overall performance of the detector degrades. The spatially enhanced histogram is implemented in the algorithm in order to introduce the spatial information about the eye in the feature vector. The length of the feature vector is now equal to $N = K \cdot 2^P$, instead of $N = 2^P$ bins in the ordinary histogram, however this issue can be overcome by varying the parameter P of the LBP operator. The normalization (Figure 4.2, block (4)) of the feature vector is needed at each scanning position due to variable size of the detectable object in order to get a coherent description of the eye, Equation (2.5).

At each scanning position (x, y) of the sliding window the probability of being an eye pattern is calculated by the pre-trained ANN based on the input representation of the object. The probability matrix is obtained after the scanning:

$$\boldsymbol{P}_{x,y} = h_w(\boldsymbol{h}_{x,y}), \tag{4.2}$$

where $\boldsymbol{P}_{x,y}$ is the value of the probability at the position (x, y), which is equal to the hypothesis value h_w of an ANN for the corresponding spatially enhanced LBP histogram $\boldsymbol{h}_{x,y}$. An output value $h_w(\boldsymbol{h}_{x,y})$ of an ANN is calculated according to the methodology described in the subsection 2.3.2. In contrast to the face detection task the size of the sliding window is fixed and is determined according to the Equation (4.1).

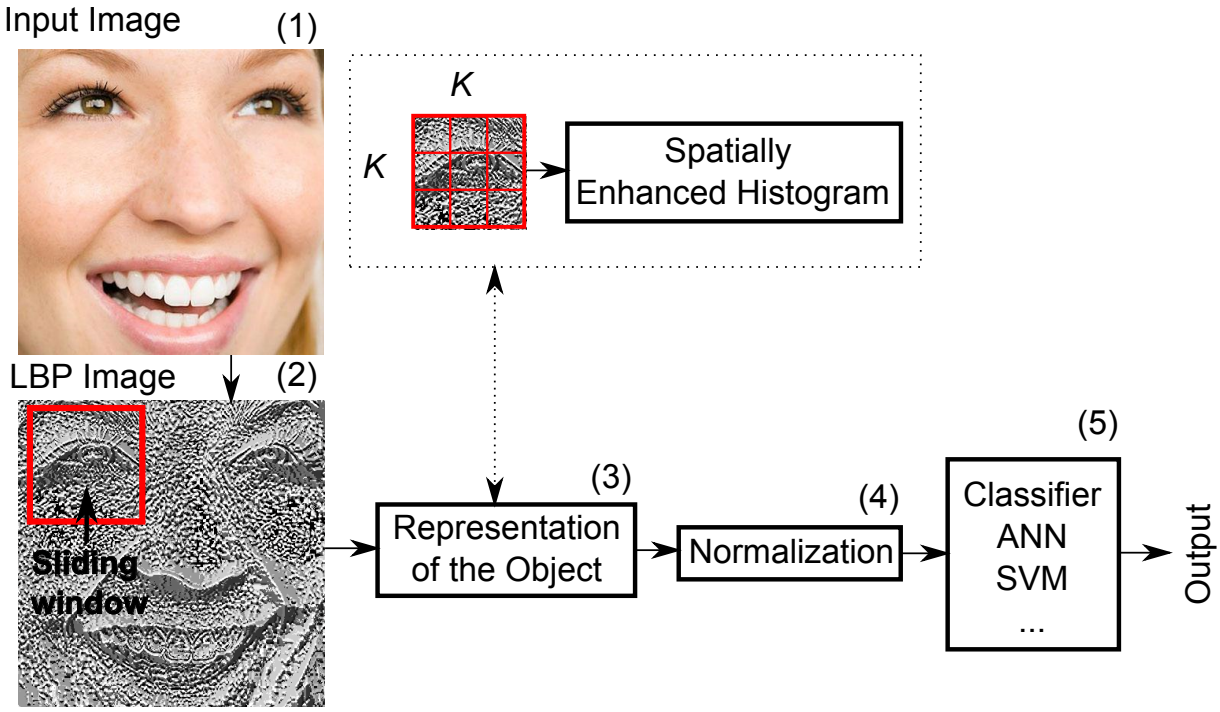


Figure 4.2: The block scheme of LBP and ANN (or SVM) based eye localization algorithm

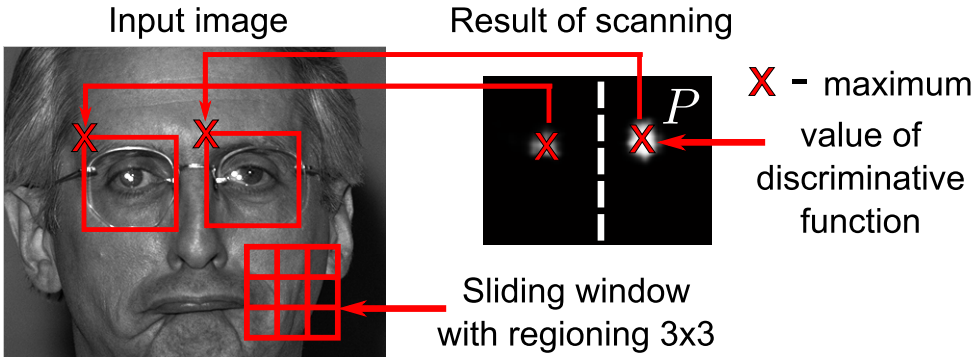


Figure 4.3: An example of the probability matrix $\boldsymbol{P}$ with following scanning parameters: $K = 3$, $\Delta_s = 2$ pixels, s_{eye} according to Equation (4.1)

An example of the probability matrix is displayed in the Figure 4.3, where scanning is performed with the sliding window of the size equal to the expected size of the eye. The step for the positions of the sliding window Δ_s in Figure 4.3 is equal to 2 pixels. The original probability matrix $\boldsymbol{P}$ is next employed in computation of the positions of eye regions.

The output matrix $\boldsymbol{P}$ of an ANN is divided into left and right halves: $\boldsymbol{P} = \{\boldsymbol{P}^L, \boldsymbol{P}^R\}$. The maximum in each matrix $\boldsymbol{P}^L$ and $\boldsymbol{P}^R$ is next detected. The coordinates of maximums $\left(x^{\{L,R\}}_{\max(\boldsymbol{P})}, y^{\{L,R\}}_{\max(\boldsymbol{P})}\right)$ refer to the most likely positions of the eyes, see Figure 4.3.

The coordinates $\left(x^L_{eye}, y^L_{eye}\right)$ of the top left corner of the left eye in the *face image* and semantically the same coordinates $\left(x^R_{eye}, y^R_{eye}\right)$ of the right eye are determined as follows:

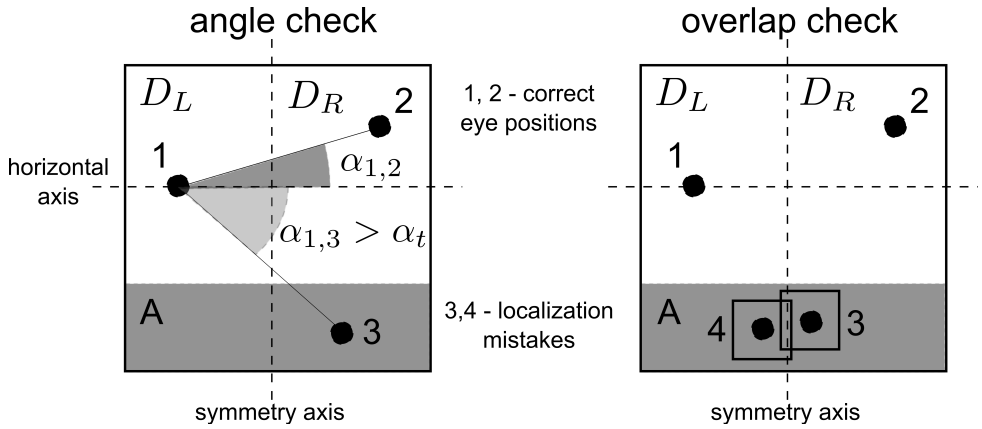


Figure 4.4: Schematic visualization of two-step empirical verification process

$$
\begin{aligned}
\left\{x^L_{\max(\boldsymbol{P})}, y^L_{\max(\boldsymbol{P})}\right\} &= \text{find}(\boldsymbol{P}^L = \max(\boldsymbol{P}^L)),\\
\left\{x^R_{\max(\boldsymbol{P})}, y^R_{\max(\boldsymbol{P})}\right\} &= \text{find}(\boldsymbol{P}^R = \max(\boldsymbol{P}^R)),\\
x^L_{eye} &= \Delta_s(x^L_{\max(\boldsymbol{P})} - 1) + R + 1,\\
y^L_{eye} &= \Delta_s(y^L_{\max(\boldsymbol{P})} - 1) + R + 1,\\
x^R_{eye} &= \Delta_s(x^R_{\max(\boldsymbol{P})} - 1) + R + 1,\\
y^R_{eye} &= \Delta_s(y^R_{\max(\boldsymbol{P})} - 1) + R + 1.
\end{aligned}
\tag{4.3}
$$

The sliding widow with a fixed size is employed in the detection of eye regions. Thus the merging of detection results is not needed in contrast to face detection algorithms.

To avoid possible localization mistakes two empirical verifications are performed next [83] (Nikisins et al.): the slope of the interocular line must be less then predetermined threshold: $\alpha_{eye} < \alpha_t$; the detected eye regions should not overlap. These principles are schematically displayed in Figure 4.4. Matrix $\boldsymbol{D}$ in the Figure 4.4 represents either the probability matrix in the case of an ANN or the discriminative matrix in the case of SVM. The matrices $\boldsymbol{D_L}$ and $\boldsymbol{D_R}$ are the corresponding halves of the matrix $\boldsymbol{D}$.

The first stage is *angle check* (Figure 4.4): suppose, that the first pair of detected points is $(1, 3)$, then the condition $\alpha_{1,2} > \alpha_t$ is satisfied. The point with highest probability is selected next from the detected set $(1, 3)$, if this point is 1, then all values in region A are set to zero. The optional maximum is determined in $\boldsymbol{D_R}$ - point 2. If the condition $\alpha_{1,2} < \alpha_t$ is satisfied, then the coordinates of the eyes correspond to points $(1, 2)$.

The second stage called *overlap check* is performed if the angle check is passed without mistakes (Figure 4.4): if the detected eye regions are very close to each other, points $(3, 4)$, then these points are considered as a mistake and all values in region A are set to zero. The next pair of points is detected in regions $\boldsymbol{D_L}$ and $\boldsymbol{D_R}$ - $(1, 2)$ and the coordinates of the eyes correspond to this set.

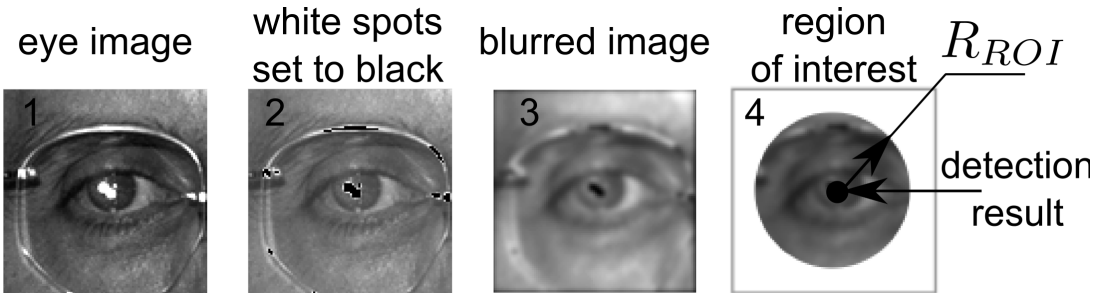


Figure 4.5: The process of eye pupil detection in the eye image

4.2.2 Localization of eye pupils

Localization of eye pupils is the second part of general eye localization algorithm. Previously described methodology for the detection of eye regions is quite insensitive to small detection offsets, a few more steps are needed to achieve the desired localization performance. These steps are based on the detection of eye pupils in the segmented eye images and are schematically displayed in Figure 4.5.

First, bright spots in the eye image are set to black, which is needed to reduce the effect of light - striking, Figure 4.5, image 2. Suppose that I^{eye} is an input eye image, then after the first stage the image is updated as follows:

$$I^{eye}(\text{find}(I^{eye} = 255)) = 0, \tag{4.4}$$

where 255 is the maximum possible intensity in the 8-bit input image. The resulting image is blurred with Gaussian low-pass filter. The size of the filter in both directions is equal to the tenth part of the s_{eye} and the standard deviation is set to $\sigma = 1.5$. Next, the disc shaped region of interest (ROI) is selected. The radius of the ROI R_{ROI} is selected according to the localization precision of the proposed eye region detector. The optimal value of the R_{ROI} will be evaluated in Section 4.5. Coordinates of the minimum (Figure 4.5, image 4) define the position of the eye pupil center in the eye image. This is the final step of pupil localization procedure.

The training of an ANN classifier is performed on eye and non-eye spatially enhanced LBP histograms, which are extracted from the color FERET database. The process of SVM training and tuning is discussed in results section.

4.2.3 Support Vector Machine - based eye localization

The proposed idea to use the combination of LBP and SVM for eye localization task is novel and is an extension of the research presented in [83] (Nikisins et al.).

The LBP-SVM eye localization task can be divided into two steps:

- *Localization of eye regions* - in this stage the squared eye regions are detected in the face image. The center of the squared region is an approximate position of the eye.

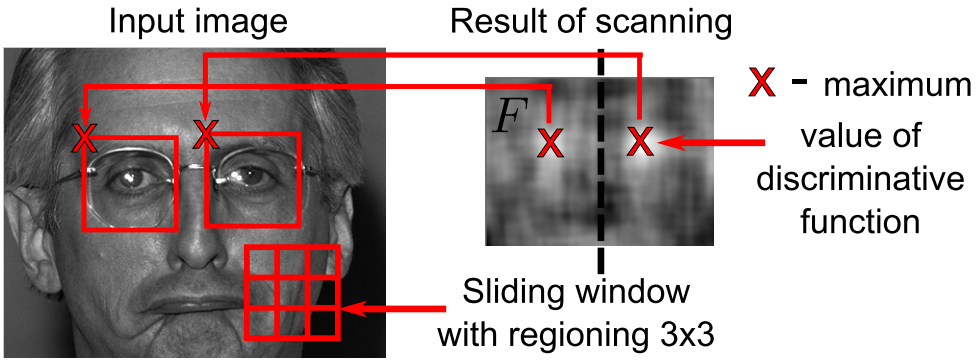


Figure 4.6: An example of the discriminative matrix $\boldsymbol{F}$ with following scanning parameters: $K = 3$, $\Delta_s = 2$ pixels, s_{eye} according to Equation (4.1)

- *Localization of eye pupils* in the eye images - in this step the centers of eye pupils are detected. Eye pupils are considered to be a good reference points in the face.

Localization of eye regions is very similar to the LBP and ANN based eye detection approach, see Figure 4.2. The only difference is in the classification module, which is now the Support Vector Machine, Figure 4.2 block (5). The advantage of this setup is the robustness of the classifier. SVM is well founded in statistical learning theory and has been successfully applied in various computer vision problems. Blocks (1) - (4) in the Figure 4.2 are discussed in details in the subsection 4.2.1 and only the classification module is described here.

At each scanning position (x, y) of the sliding window the value of discriminative function is calculated by the pre-trained non-linear SVM classifier based on the input representation of the object. The matrix with values of discriminative function is obtained after the scanning:

$$\boldsymbol{F}_{x,y} = f(\boldsymbol{h}_{x,y}), \tag{4.5}$$

where $\boldsymbol{F}_{x,y}$ is the value of discriminative function at the position (x, y). The value of $\boldsymbol{F}_{x,y}$ is calculated according to the methodology described in the subsection 2.3.3, Equation (2.45). In contrast to the face detection task the size of the sliding window is fixed and is determined according to the Equation (4.1).

An example of the discriminative matrix is displayed in the Figure 4.6, where scanning is performed with the sliding window of the size equal to the expected size of the eye. The step for the positions of the sliding window Δ_s in Figure 4.6 is equal to 2 pixels. The original discriminative matrix $\boldsymbol{F}$ is used in computation of the positions of eye regions.

Further processing of the result is still a challenging task, because some regions of the face might be misclassified as an eye. The output matrix $\boldsymbol{F}$ of the SVM is divided into left and right halves: $\boldsymbol{F} = \{\boldsymbol{F}^L, \boldsymbol{F}^R\}$. The maximum in each matrix $\boldsymbol{F}^L$ and $\boldsymbol{F}^R$ is next detected. The coordinates of maximums $\left(x^{\{L,R\}}_{\max(\boldsymbol{F})}, y^{\{L,R\}}_{\max(\boldsymbol{F})}\right)$ refer to the most likely positions of the eyes, see Figure 4.6.

The coordinates $\left(x^L_{eye}, y^L_{eye}\right)$ of the top left corner of the left eye in the *face image* and se-

mantically the same coordinates $\left(x^R_{eye}, y^R_{eye}\right)$ of the right eye are determined as follows:

$$\begin{aligned}
\left\{x^L_{\max(\boldsymbol{F})}, y^L_{\max(\boldsymbol{F})}\right\} &= \text{find}(\boldsymbol{F}^L = \max(\boldsymbol{F}^L)),\\
\left\{x^R_{\max(\boldsymbol{F})}, y^R_{\max(\boldsymbol{F})}\right\} &= \text{find}(\boldsymbol{F}^R = \max(\boldsymbol{F}^R)),\\
x^L_{eye} &= \Delta_s(x^L_{\max(\boldsymbol{F})} - 1) + R + 1,\\
y^L_{eye} &= \Delta_s(y^L_{\max(\boldsymbol{F})} - 1) + R + 1,\\
x^R_{eye} &= \Delta_s(x^R_{\max(\boldsymbol{F})} - 1) + R + 1,\\
y^R_{eye} &= \Delta_s(y^R_{\max(\boldsymbol{F})} - 1) + R + 1.
\end{aligned} \tag{4.6}$$

The sliding widow with a fixed size is employed in the detection of eye regions. Thus the merging of detection results is not needed in contrast to face detection algorithms.

Similar to an ANN-based eye localization algorithm two empirical verifications are performed so as to avoid possible localization mistakes: the slope of the interocular line must be less then predetermined threshold: $\alpha_{eye} < \alpha_t$; the detected eye regions should not overlap. These principles are discussed in section 4.2.1.

The second stage of the algorithm, which is the *localization of eye pupils*, is exactly the same as the one described in section 4.2.1.

The training of SVM classifier is performed on eye and non-eye spatially enhanced LBP histograms, which are extracted from the color FERET database. The process of SVM training and tuning is discussed in results section.

4.3 Eye localization: performance evaluation

In contrast to face detection problem, the number of detectable objects in eye localization task is known since the examined region is limited to the face image by the first stage of the automatic face recognition algorithm. This task is usually called localization and is in general easier than the detection problem.

Given the ground-truth coordinates of the eye centers (by human expert), the eye localization accuracy is usually expressed as a statistics of the error distribution made over each eye (usually the maximum), measured as the Euclidean distance [22]. In order to make these statistics accessible to the research community it is necessary to standardize the error by normalizing it over the scale of the face.

One popular error measure has been introduced in [52], and it has been already adopted by many research in the field of eye localization. This error evaluation approach can be considered as a worst case analysis. Let C_L and C_R be the true positions of the left and right eyes correspondingly and let $\tilde{C}_L$ and $\tilde{C}_R$ be the left and right eye positions estimated by the eye localization algorithm, see Figure 4.7 for details. The criteria for the evaluation of the detector performance

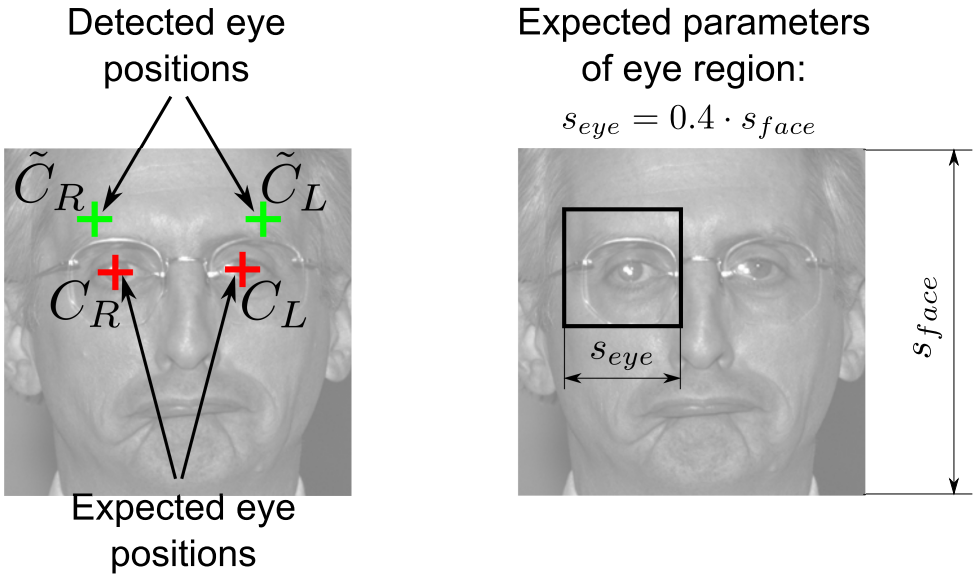


Figure 4.7: The displacement of the detected eye positions from the expected coordinates (1); the expected parameters of the face (2)

can be written as follows:

$$\eta_{eye} = \frac{\max(d(C_L, \tilde{C}_L), d(C_R, \tilde{C}_R))}{d(C_L, C_R)}, \tag{4.7}$$

where the notation $d(a, b)$ stands for the value of Euclidean distance between points a and b. Exactly the same criteria is utilized in the performance evaluation of the face detection task, see Equation (3.11).

Authors in [70] and [22] clearly show that the precision of eye localization is critical for the design of robust face recognition system, even if it does not affect all face recognition methods in the same way. The acceptable value for η_{eye} lies in the range $\eta_{eye} \leq 0.1$, which is a more strict criteria than in the face detection task.

More recently authors in [96] developed a new error measure which considers four kinds of error: the displacement error both in horizontal and vertical directions, the scale and the rotation error. However this approach is not considered in the simulation results since it is not convenient for comparative purposes.

The distribution of the proposed error measure η_{face} for all faces in the test set is converted into empirical cumulative form. Such representation is called Empirical Cumulative Distribution Function (ECDF). This approach provides a unified measure of eye localization error which is accepted by many researchers.

4.4 Experimental setup

The core of the introduced eye localization methodologies is the classifier: ANN or SVM. The robustness of the classifier to appearance variability is achieved by incorporating different as-

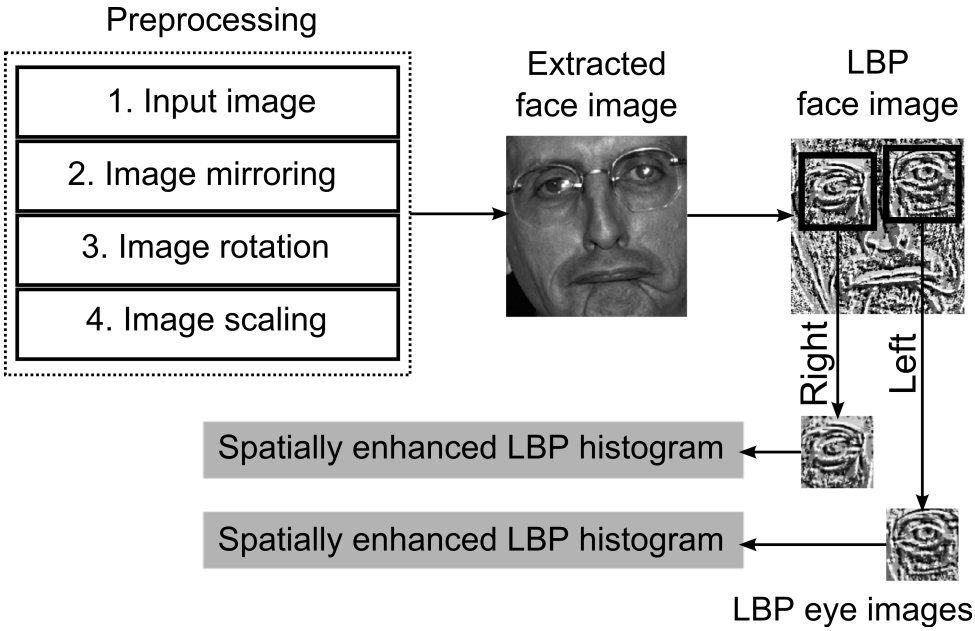


Figure 4.8: The process of forming an artificial training data for the eye class

pects of the scene into the training data. Similar to the face detection task the amount of issues in the eye localization is significant and caused by three main sources (Section 3.5): changes in appearance caused by physiological and emotional factors of the face; partial occlusions in the image; imaging problems.

Most of the above mentioned aspects with some exclusions are introduced in the utilized training data, which is extracted from the color FERET database. This database covers most of the issues that may appear in the real life eye images with some exclusions: the database is collected in the semi-controlled environment with fixed lighting conditions, the hardware setup is not changed and partial occlusions are only present in the form of glasses and hairstyle. The stated task is limited to frontal frontal face detection, therefore only the frontal subsets are selected from the database (**fa** and **fb** subsets) and only insignificant off-plane rotations which are natural for humans are present in the training set. An example of frontal face images for the first five persons in the color FERET database is displayed in the Figure 3.11. The number of frontal faces in the database is equal to 2722, which is usually not sufficient for training of the classifier. This set is artificially augmented in order to get enough training data. The process of training data forming is schematically displayed in the Figure 4.8 and consists of the following steps:

- The preprocessing is similar to the one described in Section 3.5 (Figure 3.12). The first preprocessing operation is mirroring of the input image, which separates the training and test data. All results of the algorithms performance will be reported for *non-modified* FERET database in order to make the results accessible for the research community. The size of the test set is equal to the number of frontal faces in the color FERET database $M_{test} = 2722$. Then the rotation of the input image by the angles $\alpha = (-10, -5, 0, 5, 10)$

and scaling by the factors $Scale = (0.8, 1, 1.2)$ are performed. These operations enhance the amount of available training data and introduce the robustness of the algorithm to the rotation and scale of the detectable object.

- The face region is then extracted from the modified input image. The bounding box of the face is described with four coordinates, see Figure 3.12 for details, which are calculated according to the Equation (3.12). The LBP transformation of the face image is computed next. The evaluation of parameters of LBP operator is discussed in later section.

- The eye regions are extracted from the LBP facial image. The bounding box of the eye region is determined with four coordinates by analogy with Figure 3.12:

$$\begin{aligned}
X_{start}^{R} &= \max\left(\{1, \text{round}\left(X(C_R) - 0.4 \cdot d_{eye}\right)\}\right), \\
X_{end}^{R} &= X_{start}^{R} + \text{round}\left(0.8 \cdot d_{eye}\right), \\
X_{end}^{L} &= \min\left(\{X_{max}, \text{round}\left(X(C_L) + 0.4 \cdot d_{eye}\right)\}\right), \\
X_{start}^{L} &= X_{end}^{L} - \text{round}\left(0.8 \cdot d_{eye}\right), \\
Y_{start}^{\{L,R\}} &= \max\left(\left\{1, \text{round}\left(Y(C_{\{L,R\}}) - 0.4 \cdot d_{eye}\right)\right\}\right), \\
Y_{end}^{\{L,R\}} &= Y_{start}^{\{L,R\}} + \text{round}\left(0.8 \cdot d_{eye}\right),
\end{aligned} \tag{4.8}$$

  where the indexes and superscripts L and R represent the corresponding parameters for the left and right eyes. The notations $X(C_R)$ and $X(C_L)$ stand for the X coordinates of the left and right eye pupils in the LBP face image and $Y(C_R)$, $Y(C_L)$ are the corresponding Y coordinates, see Figure 3.10 for details; X_{max} is the width of the input LBP image in pixels. The size of the eye image in this case is equal to $0.8 \cdot d_{eye}$, where d_{eye} is the value of interocular distance. The coefficient 0.8 is selected empirically so that the eye region includes all components of detectable object.

- The spatially enhanced LBP histogram is calculated for each eye according to the methodology described in Section 2.1.1.

The initial set of LBP histograms of frontal eye images is now augmented to a set of 81660 training examples. This set is sufficient for the training stage of the proposed NNC-based face detection algorithm, however it should be supplemented by the non-face training examples if the training of an ANN or SVM is performed. The methodology of calculating the non-face patterns is described below.

The process of forming the non-eye training data is schematically displayed in the Figure 4.9. The nature of eye localization task is limited to the detection of the desired object in the facial image. This assumption decreases the variation of the data in the non-eye class. However the first stage of the proposed automatic face recognition system is LBP-based face detection that is prone to detection offsets due to integral and spatially depleted nature of the face descriptive vector which in fact is a histogram. Thus, the facial region that is extracted from the input

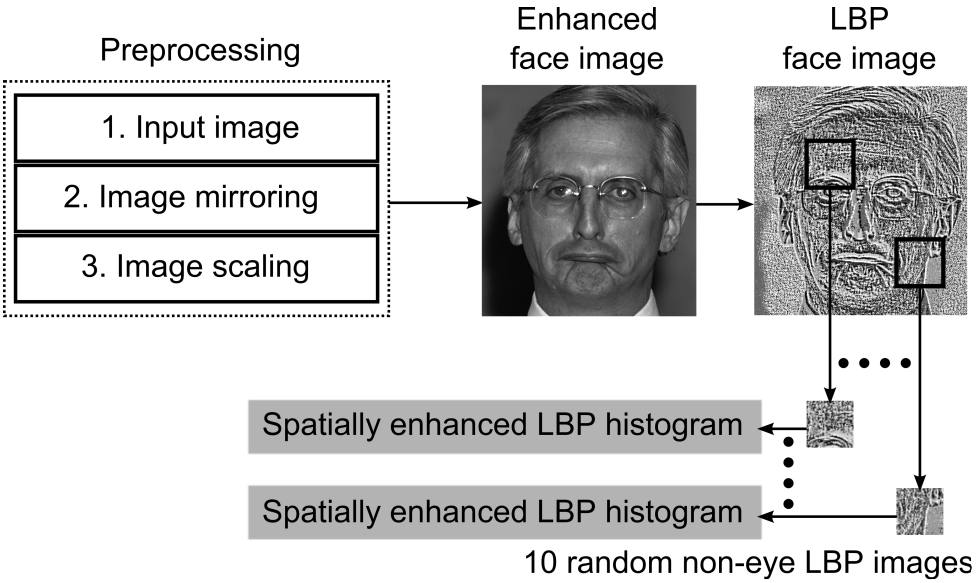


Figure 4.9: The process of forming an artificial training data for the non-eye class

image is increased in all directions, see Figure 4.9 for details. This approach enriches the non-eye class with patterns that have only partial overlap in the face region. Ten random regions are extracted from each LBP representation of the face. The spatially enhanced LBP histograms are calculated for each non-eye LBP image according to the methodology described in Section 2.1.1. The preprocessing stage of the non-eye data calculation algorithm is limited to two steps: image mirroring and scaling. The rotation of the input image is not performed. These operations are performed for all frontal face images in the color FERET database and 81660 non-eye patterns are obtained as a result.

The resulting number of face and non-face LBP histograms is equal to 163320. Some machine learning techniques require few separate data sets for the attenuation of classifier parameters, which are usually called Training, Cross Validation (CV) and Test sets. The initial 163320 histograms are split into Training and Cross Validation sets with corresponding proportions 70% and 30%. The Test set contains *non modified* frontal face images from the color FERET database. The resulting sizes of the sets are: $M_{train} = 114324$, $M_{CV} = 48996$ and $M_{test} = 2722$.

4.5 Simulation results

Evaluation of the proposed face detection algorithms is performed on a color FERET database. The modified images from the database are utilized in the process of the classifier training and optimization of the parameters of the algorithms. The LBP transformation is at the core of the proposed methods, therefore the evaluation of LBP operator parameters is discussed in the following subsection. The classifier-specific details of the algorithms and corresponding results are introduced later.

4.5.1 Evaluation of parameters of Local Binary Patterns

The first stage of the proposed eye localization algorithms is in general similar to the face detection technique. The number of parameters to be optimized is significant, therefore it is important to evaluate some of the settings before the global optimization of the system. Special technique for the evaluation of parameters of LBP operator is introduced in 3.6.1 and is based on the following equation:

$$F = \frac{(M-1)/M \sum_{i=1}^{M} \sum_{j=1}^{M} d(\boldsymbol{h}_i^e, \boldsymbol{h}_j^{ne})}{\sum_{i=1}^{M} \sum_{j=1}^{M} d(\boldsymbol{h}_i^e, \boldsymbol{h}_j^e) + \sum_{i=1}^{M} \sum_{j=1}^{M} d(\boldsymbol{h}_i^{ne}, \boldsymbol{h}_j^{ne})}, \tag{4.9}$$

where the notation $d(\boldsymbol{h}_i^e, \boldsymbol{h}_j^{ne})$ now stands for the value of Euclidean distance between spatially enhanced LBP histogram of the eye $\boldsymbol{h}_i^e$, and spatially enhanced LBP histogram of the non-eye $\boldsymbol{h}_j^{ne}$.

The encoding of patterns in the eye and non-eye classes is based on the spatially enhanced LBP histograms $(\boldsymbol{h}_i^e, \boldsymbol{h}_j^{ne})$, which are determined by four main parameters:

- P - number of sampling points in the LBP label;
- R - radius of the LBP label in pixels;
- Structure of the LBP label. Possible values: "+" - shaped or "x" - shaped, see Figure 2.5 for details;
- K - number of columns and rows in the regioning grid, Figure 2.2.

The LBP structure ("+" - shaped and "x" - shaped) and parameters P and R are evaluated in this subsection. The regioning factor K is optimized later in conjunction with the classifier due to aspects which are discussed in the next sessions. The parameters P and K have a direct impact on the dimensionality of the feature space: $N = 2^P \cdot K^2$. The dimensionality of the feature space N in logarithmic scale for different values of P and K is plotted in Figure 3.15. In order to introduce the spatial information about the object in the LBP histogram the value of K should meet the following constraints : $K \geq 2$, which results in a highly dimensional feature space with $P = 8$: $N \geq 1024$, thus, similar to the face detection, the number of sampling points is limited to $P = 4$.

The function in the Equation (4.9) now depends on the following parameters: $F(R, K, \text{structure})$. The eye histograms $\boldsymbol{h}^e$ are calculated for all frontal images ($M = 2722$) in the color FERET database and non-eye histograms $\boldsymbol{h}^{ne}$ are extracted from the same images according to the methodology from Section 4.4. The histograms are calculated for all possible combinations in the sets: $R = (1, 2, \ldots, 10)$ and $K = (2, 3, 4)$. The maximum value of regioning factor K is limited to 4 due to relatively small resolution of the eye images in the FERET database.

The dependencies $F(R, K, \text{structure})$ are displayed in the Figure 4.10. The absolute values of F are not needed for the evaluation, only the locations of the maximums are of the importance.

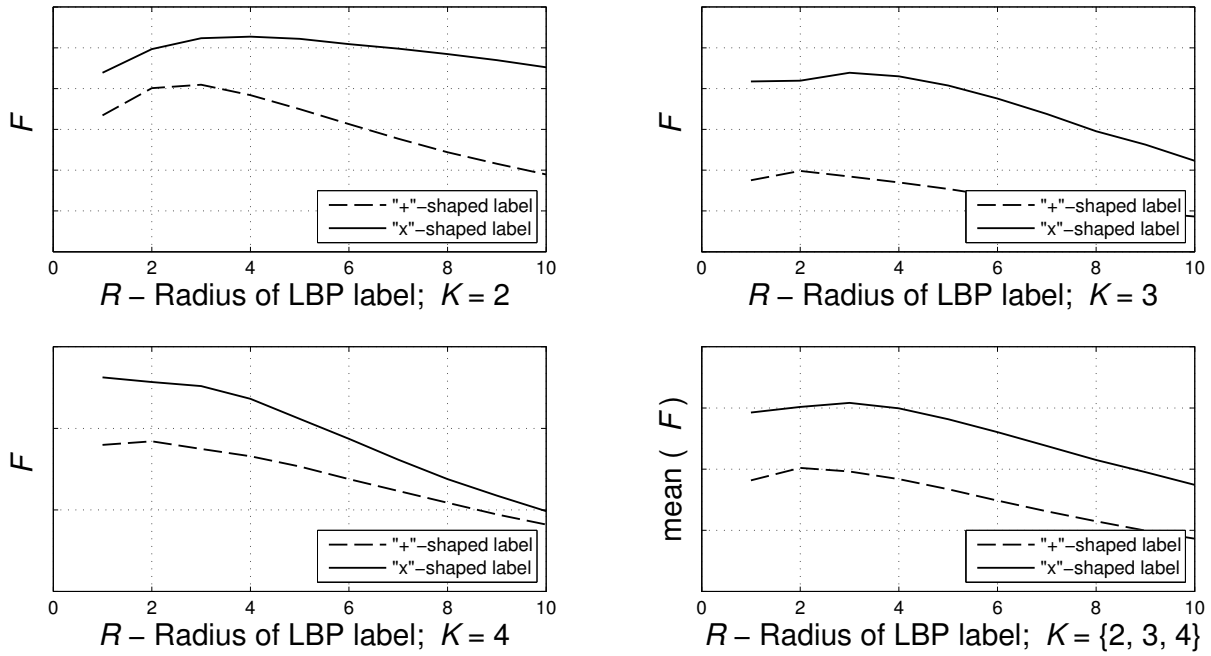


Figure 4.10: Dependencies F for the evaluation of radius R and structure of LBP operator in the eye localization task

The explicit maximums of the curves F in the Figure 4.10 for different values of regioning factor K are observed for different parameters of LBP operator. The selection of a single set of parameters R and "structure" is preferable for all values of K in order to obtain coherent simulation results. For this purpose the dependencies F of different K are averaged and the resulting curves mean(F) are displayed in the bottom-right plot of the Figure 4.10. The maximum of the mean(F) curves is observed for $\{K = 3, \text{structure} = \text{"x"}\}$, which is a good choice according to the proposed methodology.

The selected parameters of the LBP operator are summarized here:

- $P = 4$ - number of sampling points in the LBP label;
- $R = 3$ - radius of the LBP label in pixels;
- Structure of the LBP label: "x" - shaped, see Figure 2.5 for details.

4.5.2 Results for Artificial Neural Network - based eye localization

The first step of the algorithm is LBP transformation of the input image. The parameters of LBP operator are set to the values determined in the Section 4.5.1 in order to reduce the space of variables to be optimized. The main aspects of the proposed LBP and ANN based eye localization algorithm to be evaluated in this section are:

- the regioning factor K of the regioning grid (Figure 2.2),

- the structure of the classifier, which is an ANN with variable number of neurons in the hidden layer (s_{L-1}),
- the radius of the region of interest R_{ROI}, see Figure 4.5 for details.

First two of the above tasks refer to the *"localization of eye regions"* stage, while the last one refers to *"localization of eye pupils"*.

The parameters of K and s_{L-1} are evaluated jointly, because they are strictly related and have a direct impact on possible classification problems and computational complexity of the system. Small value of K and simple structure of the neural network results in high bias problem, while the opposite assumption leads to the high variance of the classifier and increases the computation time. The process of ANN adjustment is described in details in section 3.6.3 and is not covered here. Application of the methodology from section 3.6.3 yields the following parameters of an ANN in the eye localization task:

- acceptable values of the regioning factor: $K = (2, 3)$,
- the number of neurons in the hidden layer: $s_{L-1} = 10$.

The number of neurons in the hidden layer is relatively small, thus the high bias problem is not relevant for the system and the regularization parameter is set to $\lambda = 0$. The value of $K = 4$ is not considered in further experiments due to low classification precision for this data. This observation could possibly be explained with insufficient statistical information about each region in the sliding window due to low resolution of the detectable object. Poor statistical data destabilizes the LBP histograms. The increase of the resolution of the facial image can help to overcome the stated limit for K value.

The value of R_{ROI} depends on the localization precision of the first stage of the algorithm. The more accurate detection of eye regions, the smaller value of R_{ROI} is allowed. Thus, the R_{ROI} is selected after the *localization of eye regions* is performed.

Once the evaluation of ANN learning parameters and training of the classifier are completed both for $K = 2$ and $K = 3$ the *localization of eye regions* is tested on all frontal face images in the color FERET database. The advantage of the algorithm is that the size of the eye region is considered to be equal to the constant fraction of the size of the face, which is know after the face detection stage. In real life systems the size of the face always has some error. This error in general degrades the performance of eye localization stage, however in this section the size of the eye is considered to be known without additional errors. The influence of mis-detections in the face detection block on other stages of automatic face recognition system is discussed in later sections. The size of the eye is defined according to the Equation (4.1). The step of the sliding window is $\Delta_s = 5$ pixels which is a small fraction $(1/12)$ of the minimum expected size of the eye and should not degrade the precision of the detector significantly. This value of Δ_s is used both in ANN and SVM based eye localization algorithms. The value of η_{eye} is calculated

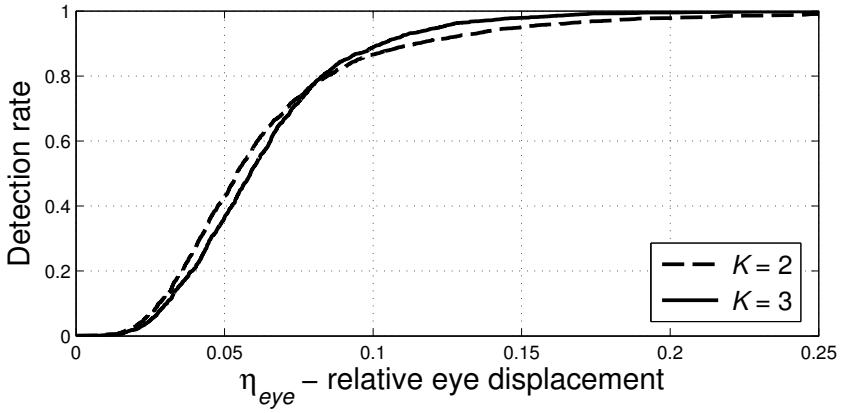


Figure 4.11: Cumulative distributions of η_{eye} for LBP and ANN based eye region localization algorithm for $K = (2, 3)$

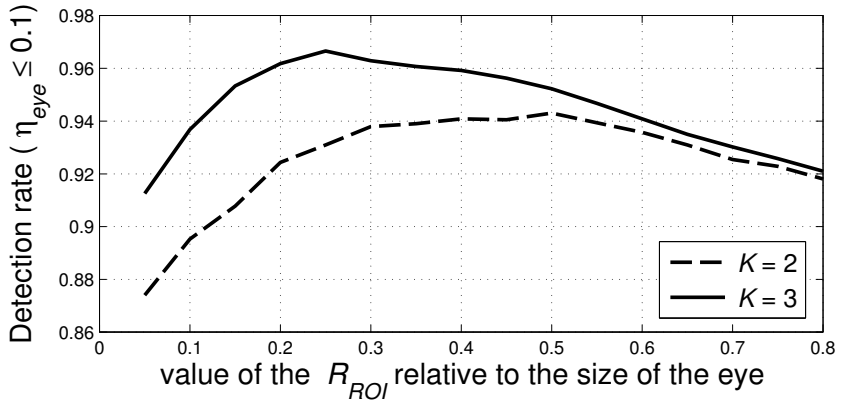


Figure 4.12: Detection rates for different values of relative radius $\hat{R}_{ROI}$; LBP and ANN based eye localization

according to the methodology described in section 4.3 and the ground-truth coordinates remain unchanged.

The cumulative distributions of η_{eye} for eye region localization stage are displayed in the Figure 4.11 both for $K = 2$ and $K = 3$. The acceptable values of relative eye displacement are $\eta_{eye} \leq 0.1$ [83] (Nikisins et al.) where the corresponding detection rates in the Figure 4.22 are:

$$P(\eta_{eye} \leq 0.1, K = 2, \text{localization of eye regions}) = 86.6\%,$$
$$P(\eta_{eye} \leq 0.1, K = 3, \text{localization of eye regions}) = 88.9\%.$$

The detection rates after the first stage of the proposed algorithm are relatively low, thus the *detection of eye pupils* is needed. First, the value of R_{ROI} (Figure 4.5) is evaluated in order to achieve highest localization rate of pupils in both eyes at $\eta_{eye} \leq 0.1$. In real life systems the size of detectable object in the input image is variable, therefore the relative value of the R_{ROI} is introduced: $\hat{R}_{ROI} = R_{ROI}/s_{eye}$. $\hat{R}_{ROI}$ is the value of the radius of the ROI relative to the size of the eye s_{eye}.

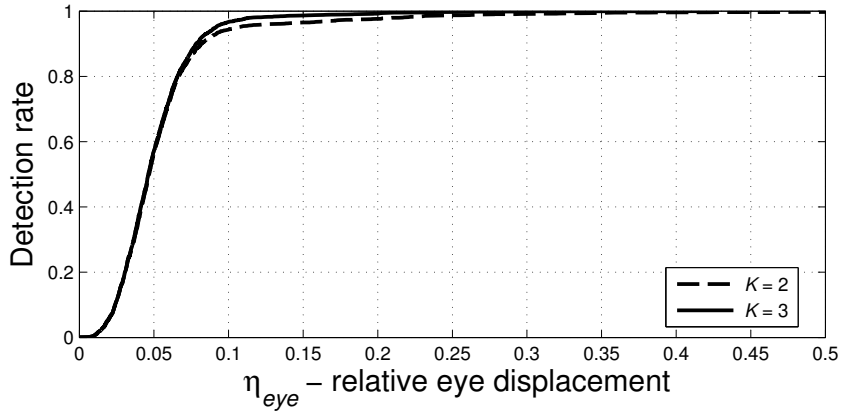


Figure 4.13: Cumulative distributions of η_{eye} for LBP and ANN based eye localization algorithm after pupil detection stage for $K = (2, 3)$; range of detection rate is $[0, 1]$

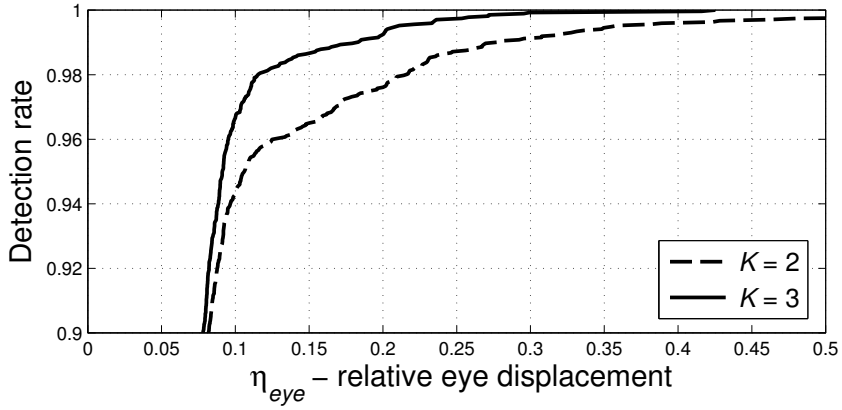


Figure 4.14: Cumulative distributions of η_{eye} for LBP and ANN based eye localization algorithm after pupil detection stage for $K = (2, 3)$; range of detection rate is $[0.9, 1]$

The dependencies $P(\hat{R}_{ROI}, \eta_{eye} \leq 0.1)$ both for $K = 2$ and $K = 3$ are displayed in the Figure 4.12. The value of $\hat{R}_{ROI}$ is selected in the range:

$$\hat{R}_{ROI} = (0.05, 0.10, \ldots, 0.80).$$

An explicit maximum in the plot of Figure 4.12 for $K = 2$ is observed for $\hat{R}_{ROI}(K = 2) = 0.5$ with corresponding $P = 94.3\%$ and for $K = 3$ the maximum is at $\hat{R}_{ROI}(K = 3) = 0.25$ with $P = 96.7\%$ These $\hat{R}_{ROI}$ values are the best choice in further evaluations.

Once the best values of $\hat{R}_{ROI}$ are determined the detection of eye pupils is added to the eye localization algorithm. The resulting cumulative distributions of η_{eye} are displayed in the Figure 4.13. The curve for $K = 3$ is slightly higher than the one for $K = 2$ but that is not clear enough from Figure 4.13. To clarify, the top part of ECDF is magnified in the Figure 4.14. The detection rates are:

$$P(\eta_{eye} \leq 0.1, K = 2, \text{localization of eye regions + pupil detection}) = 94.3\%,$$
$$P(\eta_{eye} \leq 0.1, K = 3, \text{localization of eye regions + pupil detection}) = 96.7\%.$$

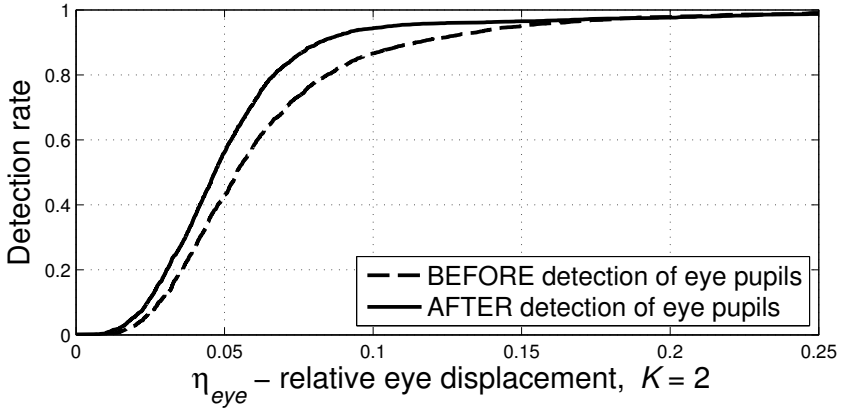


Figure 4.15: Cumulative distributions of η_{eye} for LBP - ANN eye localization algorithm before and after pupil detection stage for $K = 2$

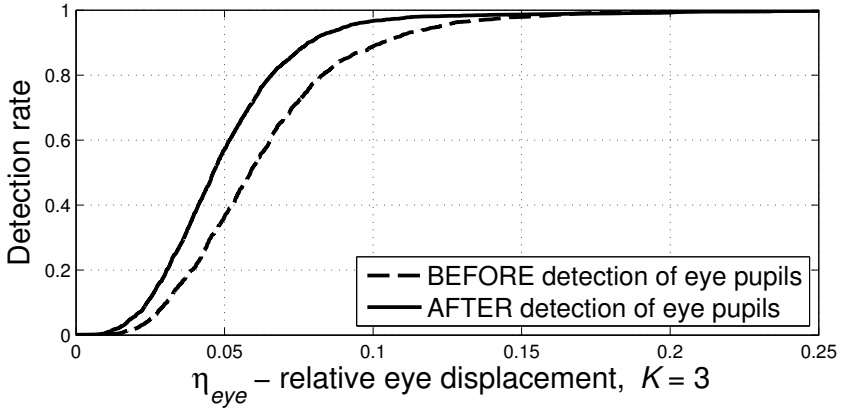


Figure 4.16: Cumulative distributions of η_{eye} for LBP - ANN eye localization algorithm before and after pupil detection stage for $K = 3$

Table 4.1:
Summary of the detection rates for LBP and ANN based eye detection algorithm

P($\eta_{eye} \leq 0.1$)	Localization of eye regions	Localization of eye pupils
$K = 2$	86.6%	94.3%
$K = 3$	88.9%	96.7%

The performance of the algorithm with other regioning parameters (values of $K = 4$ and above) is low, therefore these experiments are not described. Reduced performance can be explained with insufficient statistical information about each region when the number of cells in the regioning grid is high.

The influence of each stage of eye localization algorithm on the detection rate is displayed in the Figure 4.15 for $K = 2$ and in the Figure 4.16 for $K = 3$. The localization accuracy of LBP-ANN eye region detector alone is not sufficient due to integral nature of the utilized descriptor. Thus, the proposed detection of eye pupils is performed in the spatial domain and resulting gain in precision is significant. The results are summarized in Table 4.1.

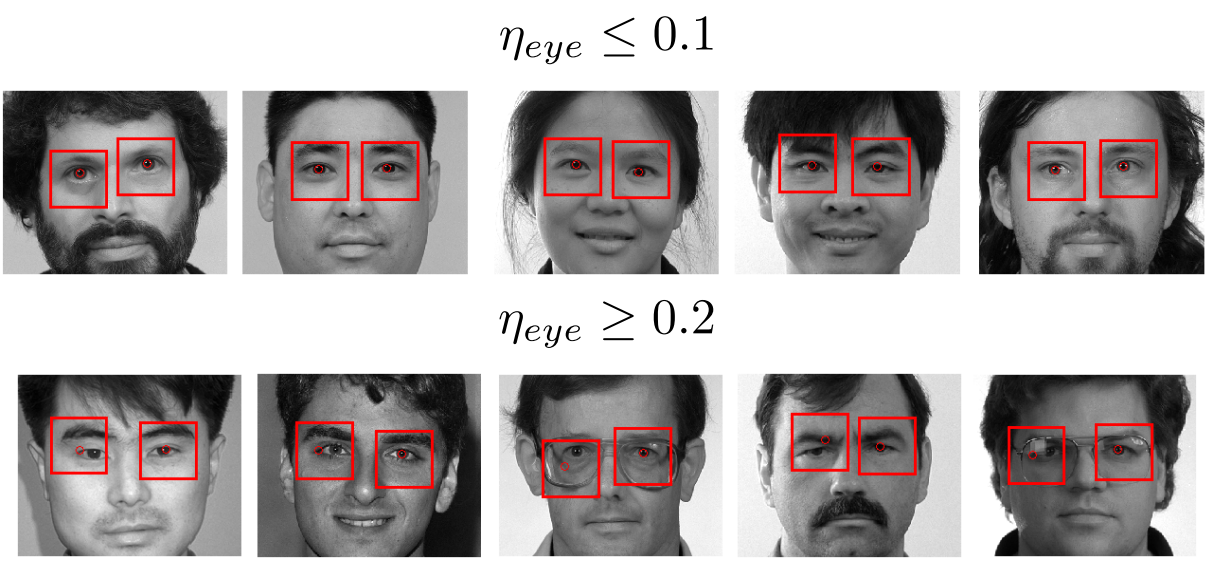


Figure 4.17: The examples of correct ($\eta_{eye} \leq 0.1$) and incorrect ($\eta_{eye} \geq 0.2$) eye detection results; LBP and ANN based eye localization

Five *randomly* selected correct ($\eta_{eye} \leq 0.1$) and incorrect ($\eta_{eye} \geq 0.2$) eye detections are displayed in the Figure 4.17.

In general the mis-detections are caused by the following factors:

- insufficient localization precision of the first stage of the algorithm: detection of eye regions. Possible positions of the eye pupil are limited by R_{ROI};
- detection of eye pupils is based on the following observation: eye pupils are always the darkest part of the eye. This assumption is sometimes violated by the presence of other dark objects in the eye image, for example glasses;
- classification errors: non-eye region is classified as being the object of interest.

As seen in the Figure 4.17, first two of the above mentioned factors cause the main localization problems in the LBP-ANN based eye detection algorithm.

4.5.3 Results for Support Vector Machine - based eye localization

The first step of the proposed SVM-based eye localization algorithm is LBP transformation of the input face image. The parameters of LBP operator are set to the values determined in the Section 4.5.1 in order to reduce the space of variables to be optimized. The main aspects of the proposed LBP and SVM based eye localization algorithm to be evaluated in this section are:

- the regioning factor K of the regioning grid (Figure 2.2),
- the structure of the classifier, which is an SVM,
- the radius of the region of interest R_{ROI}, see Figure 4.5 for details

First two of the above points refer to the *"localization of eye regions"* stage, while the last one refers to *"localization of eye pupils"*. Similar to SVM based face detection algorithm first two of the above aspects are related and should be evaluated indivisibly. The value of R_{ROI} depends on the localization precision of the first stage of the algorithm. The more accurate detection of eye regions, the smaller value of R_{ROI} is allowed.

Similar to SVM based face detection the design of the SVM classifier should satisfy two main aspects:

- high classification precision, which is clearly needed for high discriminative power of the detector,
- small number of support vectors (SV), which has a direct impact on the execution time.

A compromise should be found between above conditions as they are partially mutually exclusive in case of nonlinearly separable data sets. The non-linear classifier with RBF kernel is selected for eye detector. This choice is based on the success of this classifier in the face detection stage (Section 3.6.4). The size of the expected feature space is relatively low, thus RBF kernel is a suitable option.

The parameters to be adjusted for an RBF kernel are: $C > 0$ - the penalty of the error term; γ - determines the area of influence of the support vector over the data space, see Equation (2.43) for details. A grid-search on C and γ using cross-validation set is selected in order to evaluate the model. Various pairs of (C, γ) values are tried and the one with high cross-validation accuracy and low number of SV is selected. Exponentially growing sequences of C and γ [26] are utilized in order to identify good parameters:

$$\begin{aligned} C &= \{2^{\boldsymbol{n}^C}\}, \boldsymbol{n}^C = (-5, -3, -1, 1 \dots 25), \\ \gamma &= \{2^{\boldsymbol{n}^\gamma}\}, \boldsymbol{n}^\gamma = (-13, -11, -9, -7 \dots 3), \end{aligned} \tag{4.10}$$

where the values of degree n^C and n^γ are iteratively increased by a step of 2. The sizes of training and cross-validation sets are reduced in the grid search stage to accelerate the evaluation process : $M_{train} = 4000$ and $M_{CV} = 4000$. Once the best parameters of (C, γ) are selected the training of SVM is performed with a complete training set.

The resolution of eye images is not that high as in the case of facial images. Thus the value of $K = 2$ is considered in the first sequence of experiments. For each pair of parameters (n_i^γ, n_j^C) (Equation (4.10)) the corresponding accuracy measure for the CV set $P_{i,j}^{CV}$ and number of support vectors $N_{i,j}^{SV}$ are calculated. These values are substituted into the matrices:

$$\begin{aligned} \boldsymbol{P}^{CV} &= \{P_{i,j}^{CV}\},\ i = 1, \dots, 9;\ j = 1, \dots, 16, \\ \boldsymbol{N}^{SV} &= \{N_{i,j}^{SV}\},\ i = 1, \dots, 9;\ j = 1, \dots, 16, \end{aligned}$$

where the number of elements in vector $\boldsymbol{n}^\gamma$ is 9 and in vector $\boldsymbol{n}^C$ is 16.

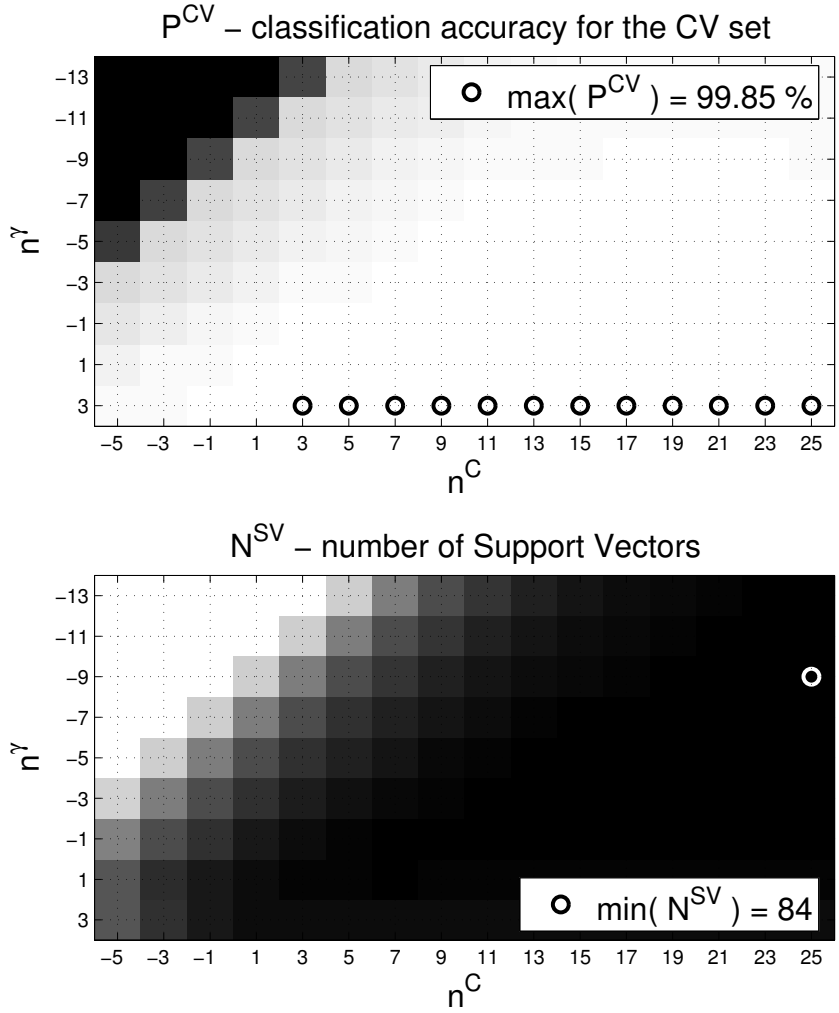


Figure 4.18: Full range images of matrices $\boldsymbol{P^{CV}}$ and $\boldsymbol{N^{SV}}$ in the eye localization task with regioning parameter $K = 2$; $M_{train} = 4000$ and $M_{CV} = 4000$

The matrices $\boldsymbol{P}^{CV}$ and $\boldsymbol{N}^{SV}$ with $K = 2$ are displayed in the Figure 4.18. The maximum accuracy on the cross-validation set max($\boldsymbol{P}^{CV}$) and the minimum number of support vectors min($\boldsymbol{N}^{SV}$) are plotted with a circle markers "o" in Figure 4.18. The ideal scenario is to select parameters (n_i^γ, n_j^C) corresponding to the point of intersection between segments max($\boldsymbol{P}^{CV}$) and min($\boldsymbol{N}^{SV}$), however this is not the case due to an absence of intersection. Thus, the compromise between the values of $\boldsymbol{P}^{CV}$ and $\boldsymbol{N}^{SV}$ is needed.

The methodology for the evaluation of the compromise between $\boldsymbol{P}^{CV}$ and $\boldsymbol{N}^{SV}$ is described in section 3.6.4, Equations (3.16) - (3.17). The last 100 elements of vectors $\boldsymbol{p}^{CV}$ and $\boldsymbol{n}^{SV}$, that are calculated according to this methodology, and corresponding $\boldsymbol{n}^C$ and $\boldsymbol{n}^\gamma$ values are plotted in Figure 4.19. The region of interest (ROI) in Figure 4.19 is located between two black vertical lines. The classification accuracy in ROI is still high $P_{ROI}^{CV} = 99.55\%$ while the number of support vectors is relatively small $N_{ROI}^{SV} = 105$. The values of n^C and n^γ that are closest to

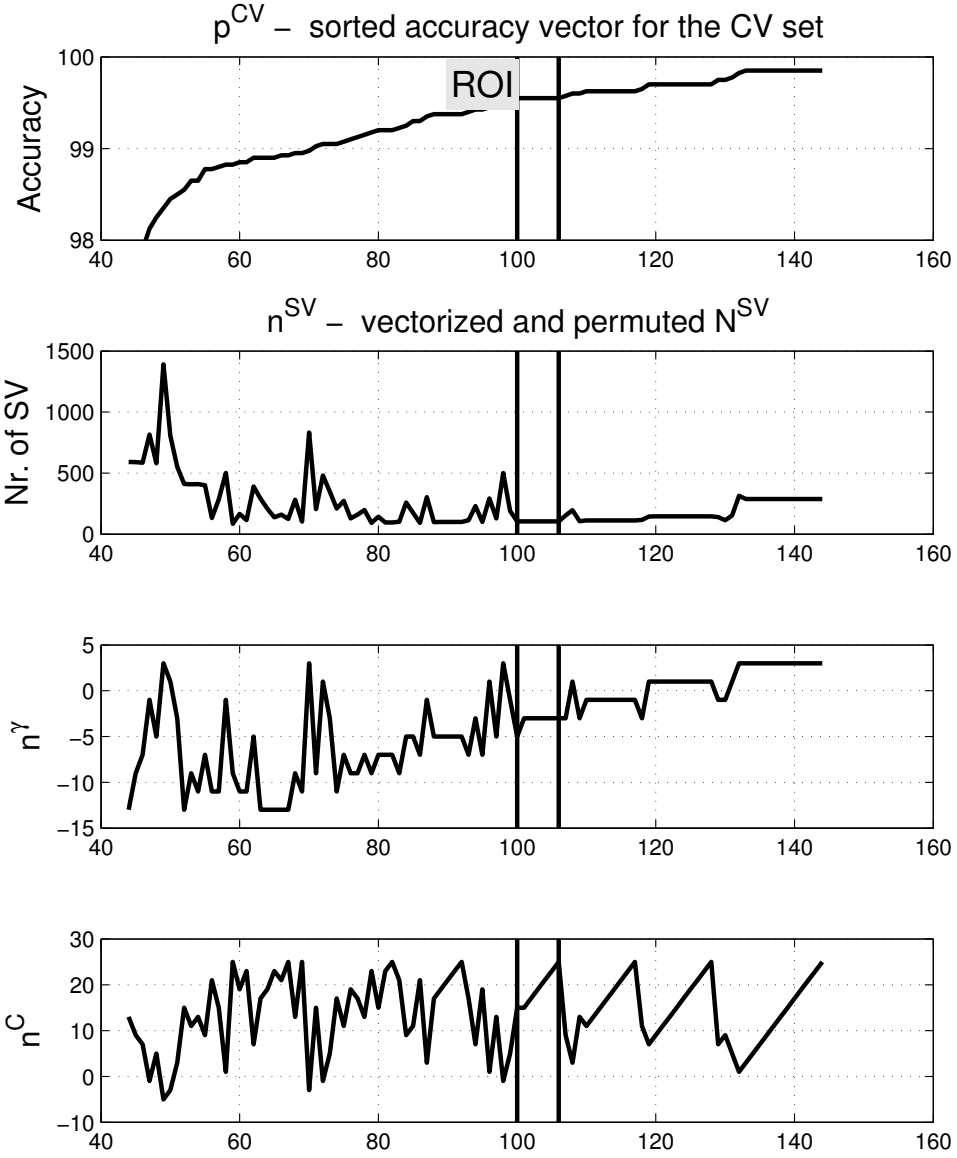


Figure 4.19: Sorted accuracy vector $\boldsymbol{p}^{CV}$ and corresponding $\boldsymbol{n}^{SV}$, $\boldsymbol{n}^{C}$ and $\boldsymbol{n}^{\gamma}$ for $K = 2$ in the eye localization task

one [26] in the region of interest are:

$$n^C = 15, \;\; n^\gamma = -3.$$

Once the best learning parameters $(C = 2^{15}, \gamma = 2^{-3})$ are selected for $K = 2$ the training of SVM is performed with a complete training set. The resulting number of support vectors after the training with a complete data set is $N^{SV}(K = 2) = 206$. The corresponding accuracy for the CV set is $P^{CV} = 99.88\%$. The number of support vectors is obviously increased from 105 to 206.

The same evaluation process is repeated for $K = 3$. The matrices $\boldsymbol{P}^{CV}$ and $\boldsymbol{N}^{SV}$ for $K = 3$ are displayed in the Figure 4.20.

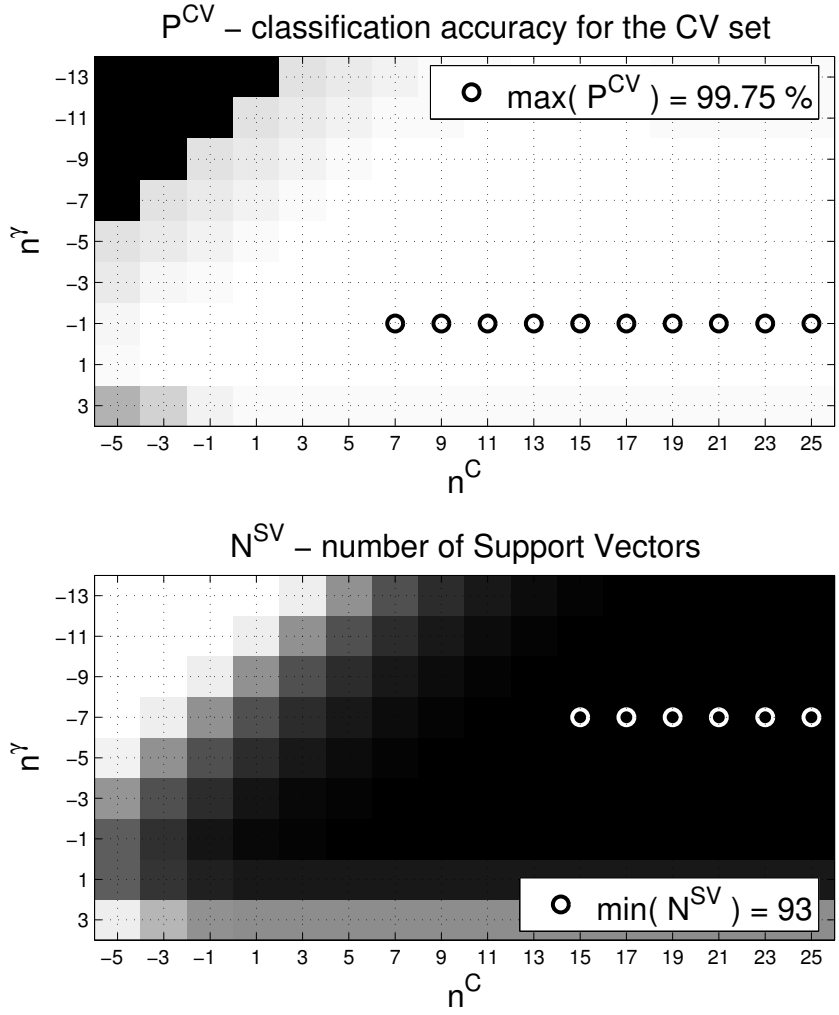


Figure 4.20: Full range images of matrices $\boldsymbol{P^{CV}}$ and $\boldsymbol{N^{SV}}$ in the eye localization task with regioning parameter $K = 3$; $M_{train} = 4000$ and $M_{CV} = 4000$

The last 100 elements of vectors $\boldsymbol{p}^{CV}$ and $\boldsymbol{n}^{SV}$ and corresponding $\boldsymbol{n}^{C}$ and $\boldsymbol{n}^{\gamma}$ values for $K = 3$ are plotted in Figure 4.21. The region of interest (ROI) in Figure 4.21 is located between two black vertical lines. The classification accuracy in ROI is still high $P^{CV}_{ROI} = 99.58\%$ while the number of support vectors is still small $N^{SV}_{ROI} = 112$. The values of n^C and n^γ that are closest to 1 in the region of interest are: $n^C = 11,\ n^\gamma = -3$.

Once the best learning parameters $(C = 2^{11}, \gamma = 2^{-3})$ are selected for $K = 3$ the training of SVM is performed with a complete training set. The resulting number of support vectors after the training on the complete data set is $N^{SV}(K = 3) = 216$. The corresponding accuracy for the CV set is $P^{CV} = 99.89\%$.

Further augment of regioning parameter K results in a high amount of support vectors. The smallest number of SV for $K = 4$ is $N^{SV} > 2000$. This observation could possibly be explained with insufficient statistical information about each region in the sliding window due to low resolution of the detectable object. Poor statistical data destabilizes the LBP histograms. The increase of the resolution of the facial image can help to overcome the stated issue.

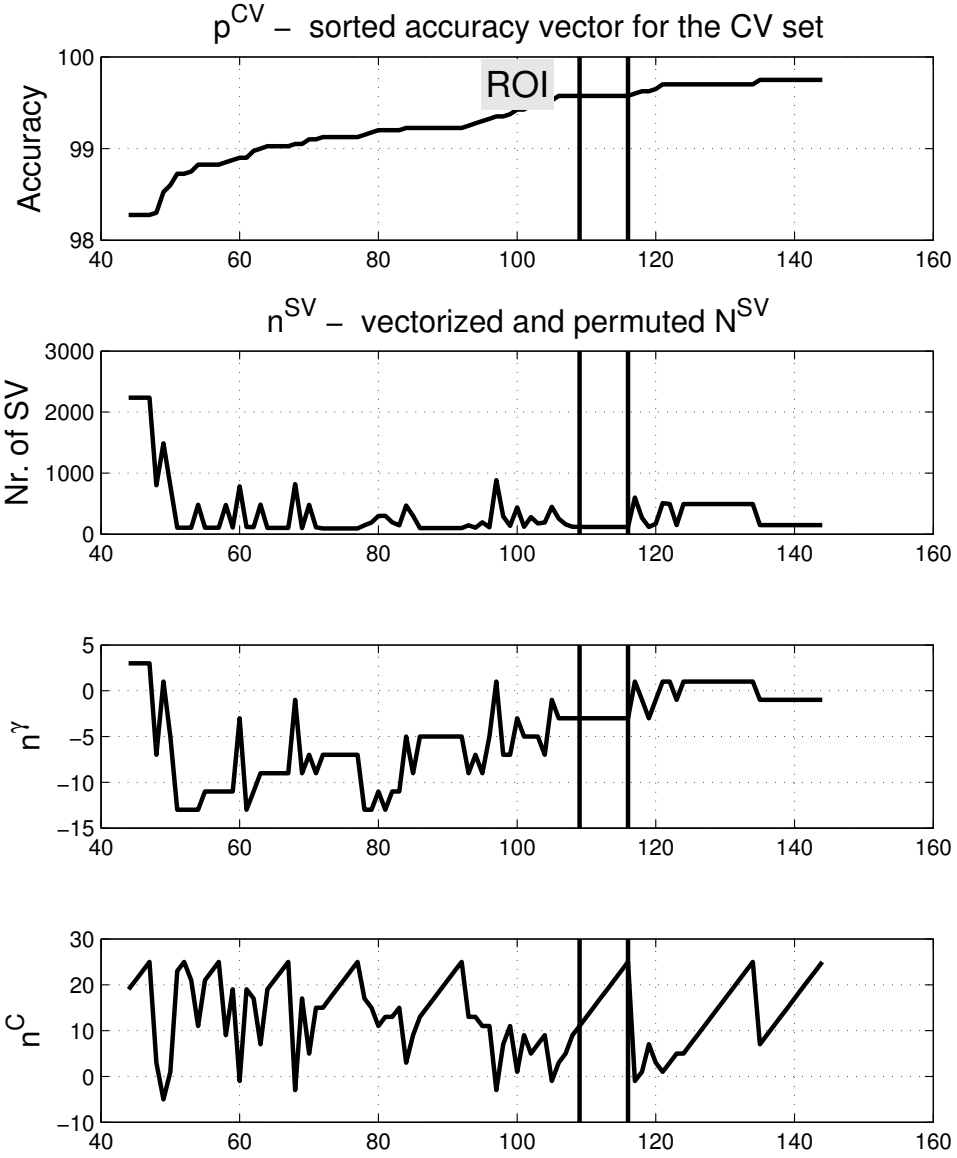


Figure 4.21: Sorted accuracy vector $\boldsymbol{p}^{CV}$ and corresponding $\boldsymbol{n}^{SV}$, $\boldsymbol{n}^{C}$ and $\boldsymbol{n}^{\gamma}$ for $K = 3$ in the eye localization task

The evaluation of SVM learning parameters is now completed both for $K = 2$ and $K = 3$ and the *localization of eye regions* is now tested on all frontal face images in the color FERET database. The advantage of the algorithm is that the size of the eye region is explicitly defined by the size of the face which is know after the face detection stage. In real life systems the size of the face always has some error. This error in general degrades the performance of eye localization stage, however in this section the size of the eye is considered to be known without additional errors. The influence of mis-detections in the face detection block on other stages of automatic face recognition system is discussed in later sections. The size of the eye is defined according to the Equation (4.1). The value of η_{eye} is calculated according to the methodology described in section 4.3 and the ground-truth coordinates remain unchanged.

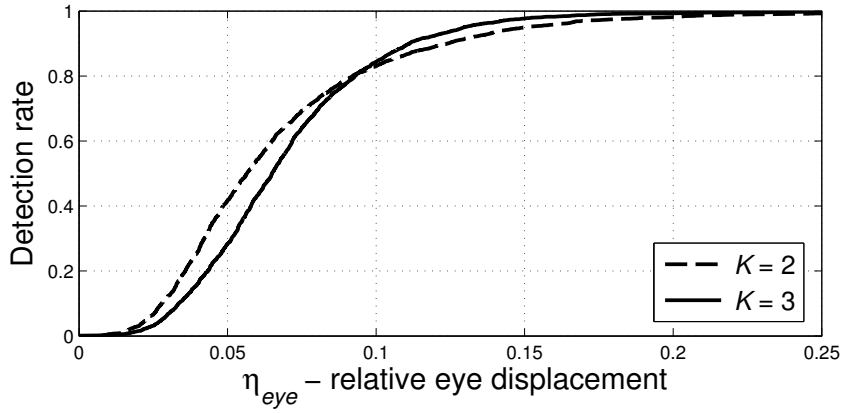


Figure 4.22: Cumulative distributions of η_{eye} for LBP and SVM based eye region localization algorithm for $K = (2, 3)$

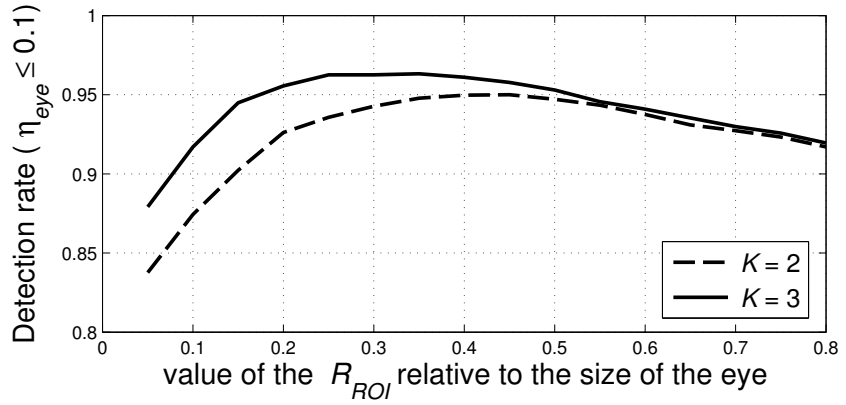


Figure 4.23: Detection rates for different values of relative radius $\hat{R}_{ROI}$; LBP and SVM based eye localization

The cumulative distributions of η_{eye} for eye region localization stage are displayed in the Figure 4.22 both for $K = 2$ and $K = 3$. The acceptable values of relative eye displacement are $\eta_{eye} \leq 0.1$ [83] (Nikisins et al.) where the corresponding detection rates in the Figure 4.22 are:

$$P(\eta_{eye} \leq 0.1, K = 2, \text{localization of eye regions}) = 83.0\%,$$
$$P(\eta_{eye} \leq 0.1, K = 3, \text{localization of eye regions}) = 84.4\%.$$

The detection rates after the first stage of the proposed algorithm are relatively low, thus the *detection of eye pupils* is needed. First, the value of R_{ROI} (Figure 4.5) is evaluated in order to achieve highest localization rate of pupils in both eyes at $\eta_{eye} \leq 0.1$. The size of detectable object in the input image is variable, therefore the relative value of the R_{ROI} is introduced: $\hat{R}_{ROI} = R_{ROI}/s_{eye}$. $\hat{R}_{ROI}$ is the value of the radius of the ROI relative to the size of the eye s_{eye}.

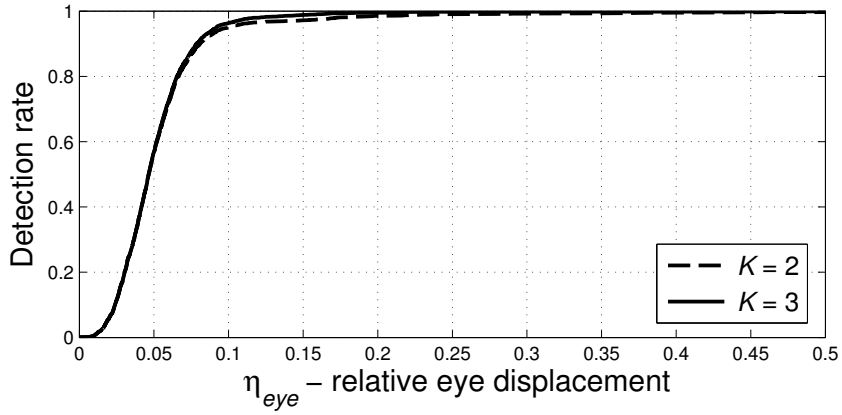


Figure 4.24: Cumulative distributions of η_{eye} for LBP and SVM based eye localization algorithm after pupil detection stage for $K = (2, 3)$; range of detection rate is $[0, 1]$

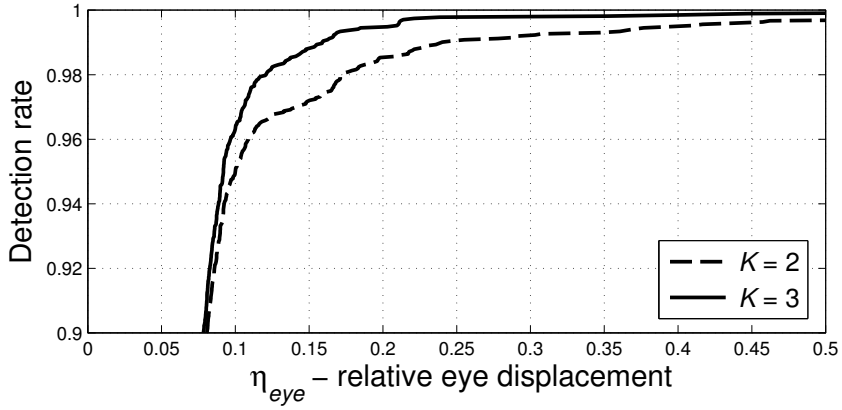


Figure 4.25: Cumulative distributions of η_{eye} for LBP and SVM based eye localization algorithm after pupil detection stage for $K = (2, 3)$; range of detection rate is $[0.9, 1]$

The dependencies $P(\hat{R}_{ROI}, \eta_{eye} \leq 0.1)$ both for $K = 2$ and $K = 3$ are displayed in the Figure 4.23. The value of $\hat{R}_{ROI}$ is selected in the range:

$$\hat{R}_{ROI} = (0.05, 0.10, \ldots, 0.80).$$

An explicit maximum in the plot of Figure 4.23 for $K = 2$ is observed for $\hat{R}_{ROI}(K = 2) = 0.45$ with corresponding $P = 95.0\%$ and for $K = 3$ the maximum is at $\hat{R}_{ROI}(K = 3) = 0.35$ with $P = 96.3\%$ These $\hat{R}_{ROI}$ values are the best choice in further evaluations.

Once the best values of $\hat{R}_{ROI}$ are determined the detection of eye pupils is added to the eye localization algorithm. The resulting cumulative distributions of η_{eye} are displayed in the Figure 4.24. The curve for $K = 3$ is slightly higher than the one for $K = 2$ but that is not clear enough from Figure 4.24. To clarify, the top part of ECDF is magnified in the Figure 4.25. The detection rates are:

$$P(\eta_{eye} \leq 0.1, K = 2, \text{localization of eye regions + pupil detection}) = 95.0\%,$$
$$P(\eta_{eye} \leq 0.1, K = 3, \text{localization of eye regions + pupil detection}) = 96.3\%.$$

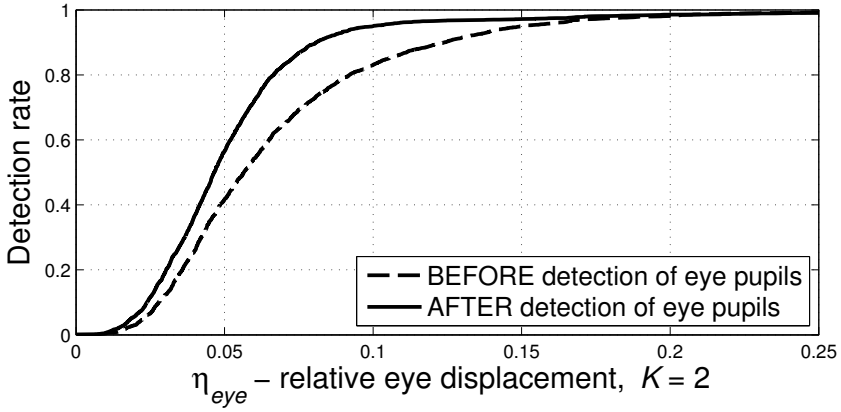


Figure 4.26: Cumulative distributions of η_{eye} for LBP - SVM eye localization algorithm before and after pupil detection stage for $K = 2$

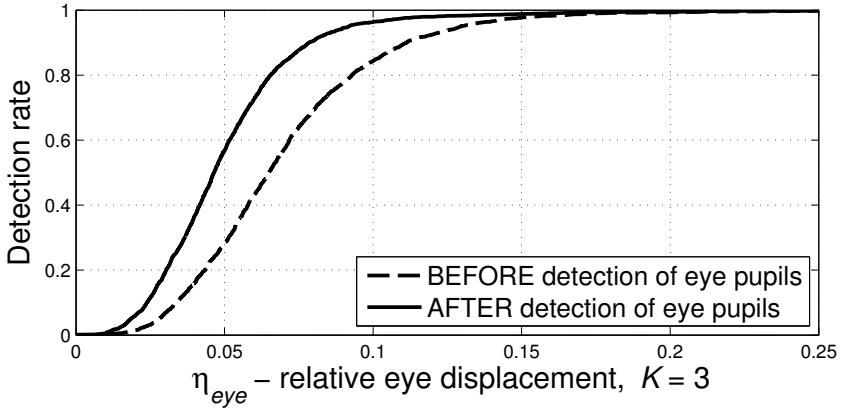


Figure 4.27: Cumulative distributions of η_{eye} for LBP - SVM eye localization algorithm before and after pupil detection stage for $K = 3$

Table 4.2:
Summary of the detection rates for LBP and SVM based eye detection algorithm

P($\eta_{eye} \leq 0.1$)	Localization of eye regions	Localization of eye pupils
$K = 2$	83.0%	95.0%
$K = 3$	84.4%	96.3%

The influence of each stage of eye localization algorithm on the detection rate is displayed in the Figure 4.26 for $K = 2$ and in the Figure 4.27 for $K = 3$. The need of pupil detection stage is obvious. The localization accuracy of LBP-SVM eye region detector alone is not sufficient due to integral nature of the utilized descriptor. The detection of eye pupils is performed in the spatial domain, thus the resulting gain in precision is significant. The results are summarized in Table 4.2.

First five *randomly* selected correct ($\eta_{eye} \leq 0.1$) and incorrect ($\eta_{eye} \geq 0.2$) eye detections are displayed in the Figure 4.28. As seen in the Figure 4.28, glasses are the main reason of possible mis-detections. The estimation of reported detection rates is based on the ground truth data provided by human-expert for the FERET database, however this data is sometimes prone

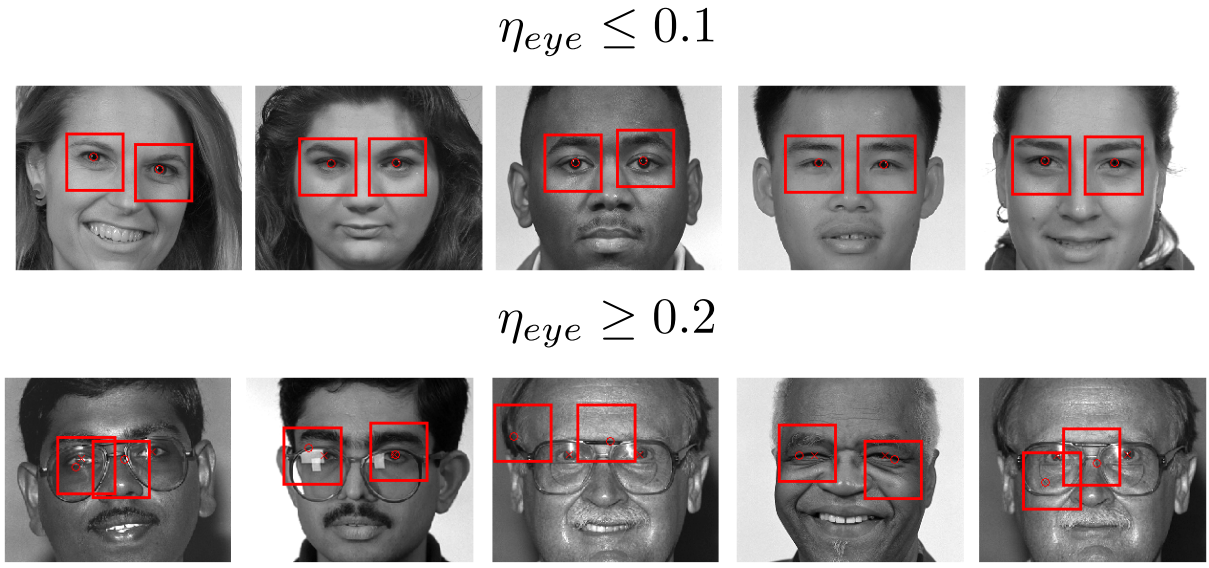


Figure 4.28: The examples of correct ($\eta_{eye} \leq 0.1$) and incorrect ($\eta_{eye} \geq 0.2$) eye detection results; LBP and SVM based eye localization

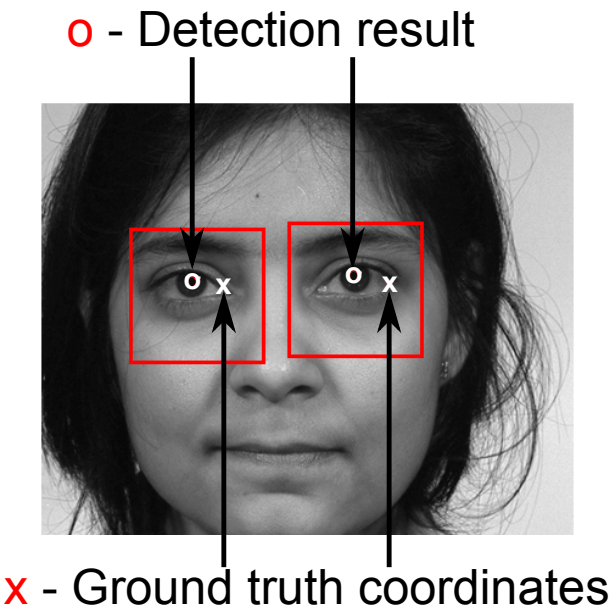


Figure 4.29: An example of incorrect ground truth data

to mis-localizations. An example of explicitly erroneous ground truth information is displayed in the Figure 4.29, where the algorithm performed better than the human-expert.

4.5.4 Comparison of eye localization algorithms

In this section a comparison of the proposed eye localization techniques with other popular methodologies is introduced. The comparative study of the eye localization algorithms is a challenging task due to the factors which are briefly summarized in Section 3.6.5. In contrast to unpopularity of color FERET database among face detection researchers this facial dataset is often utilized in the experiments with eye localization algorithms. This fact can be explained

Table 4.3:
Comparison of eye localization algorithms

Method:	Parameters:	P($\eta_{eye} \leq 0.1$)	Detection time
LBP+ANN	$K = 3$, $s_{L-1} = 10$	96.7%	0.450 seconds
LBP+SVM	$K = 3$, $N^{SV} = 216$	96.3%	0.970 seconds
PSEF [105]		83.0%	0.001 seconds [105]
ASEF [17]		66.1%	
Haar-like features [105], [117]		44.7%	0.046 seconds [105]

with relatively high resolution of face images which is usually needed for eye detectors.

The proposed algorithms are compared with eye localization methods which are based on Haar-like features [117] and correlation filters [17], [105]. The detection rates at $\eta_{eye} = 0.1$ and average detection times are summarized in Table 4.3 (tested on FERET database).

The localization time of both eyes in the facial images may vary from image to image, which is especially expressed for Haar-based eye localization algorithm. Thus the localization times for LBP+ANN and LBP+SVM algorithms in Table 4.3 are an average of eye localization times for all frontal face images in the color FERET database. The localization times for PSEF and Haar-based eye localization algorithms are derived from [105] (average time, tested on FERET). The proposed algorithms (LBP+ANN and LBP+SVM) were tested on the same computer (I5-2500K processor and 4GB of RAM) in the Matlab environment. The parameters of the testing platform are not available in [105] (PSEF and Haar-based eye localization algorithms).

While the Haar-like features are effective in face detection task (Section 3.6.5), this method significantly inferior in accuracy to our eye localization algorithm. Thus the cluster of proposed detection principles has a better generalization to various tasks. The correlation filters for object detection are very effective in terms of computational time both for learning and detection stages, however the localization accuracy is still not high enough (see Table 4.3 for details).

4.5.5 Conclusions

In this chapter an overview of significant eye localization approaches, including previous LBP-based detection techniques, are reviewed. A novel cluster of LBP-based eye localization algorithms is proposed. The introduced localization algorithms consists of two main stages: localization of eye regions and detection of eye pupils. The first stage is an extension of proposed face detection methods to another cluster of detectable objects. The second stage is needed for further gain in the localization precision. The experiments clearly show that the proposed method outperforms a number of state-of-the-art eye localization approaches, Table 4.3. High localization accuracy (Table 4.3) is obtained in low-dimensional feature space (144 LBP features) and with simple classifier (Artificial Neural Network with 10 Neurons in the hidden layer or Support Vector Machine (SVM) with 100-200 Support Vectors). Similar to face detection, the scope of the experiments in this research is limited to the task of eye localization in frontal face

images taken under semi-controlled lighting conditions. Partial occlusions are presented in the test images in the form of glasses, which is the main reason of erroneous detections. The time for eye localization is measured in the range of seconds (for images in FERET dataset, average desktop PC with I5 quad-core processor and 4 GB of RAM) and depends on the parameters of the system. This fact can be explained by the absence of special techniques for the reduction of the number of scanning positions. Some of possible improvements of scanning process are described in 3.6.6.

Further improvements in localization accuracy are also possible and potential supplements to the algorithm are briefly summarized here:

- *Reduction of the search space*. The eye localization stage is the second step in automatic face recognition process and the location of the face is already estimated by the face detector. This information can help to restrict the area of possible eye locations in the input face image.

- *Utilize information about other facial features*. The analyzed input image is known to be face, thus the knowledge of relative positions of the facial features (eyes, nose, moth and others) can help to improve accuracy.

- *Shape information*. The second stage of eye localization algorithm, namely detection of eye pupils, is based on the intensity information only. The shape of eye pupils is explicitly circular which can also benefit as an additional data in more complicated localization approaches.

Above mentioned methodologies are based on the knowledge about the analyzed input image, which in our case is a face. The first principle can both improve the accuracy and speed up the detector and is extremely easily implementable. However these aspects are out of the scope of this research.

Chapter 5

FACE RECOGNITION

Face recognition is one of the most studied fields in computer vision over a couple of last decades. An extensive research in this field led to the fact that nowadays face recognition is one of the most successful applications of image analysis and understanding [104]. Face recognition is of interest not only for computer science researchers, but also for neuroscientists and psychologists. It is the general opinion that advances in computer vision research will provide useful insights to neuroscientists and psychologists into how human brain works, and vice versa. Face recognition has a wide range of applications, such as access control systems, border control, forensics, banking sector, human computer interaction, patient monitoring, image database investigations, video indexing and others.

According to the taxonomy in [5] the term *recognition* in biometric applications is a generic term and does not necessarily imply the description of the specific purpose of the system. All biometric systems perform recognition, but the recognition task can be divided in two groups:

- *verification* occurs when the biometric system attempts to confirm the claimed identity of an individual by comparing a submitted sample to one or more previously enrolled templates;
- *identification* occurs when the system attempts to determine the identity of an individual. A biometric data is collected and compared to all the templates in the database.

The scope of this research is limited to *identification* task, which in general is more complicated than the verification problem.

Face recognition techniques can be divided into three categories based on the face data acquisition principles: algorithms that operate on intensity images; those that deal with video sequences; and those that imply other sensory data, for example 3D information, infra-red or thermal imagery. The algorithms in this research are designed to operate with intensity images only.

5.1 Related work

Many face recognition algorithms have been proposed in scientific papers in the last decades. A comprehensive survey of many algorithms is provided in [134] and [108] with more recent review in [49]. The problem of person identification attracted researchers with different scientific backgrounds: psychology, pattern recognition, machine learning, artificial intelligence, computer vision, computer graphics and others. Due to this fact that the literature on face recognition is vast and diverse, thus it is difficult to classify these systems based purely on what types of techniques they use for feature representation or classification. To have a clear and high-level categorization, authors in [134] introduced the idea to follow a guideline suggested by the psychological study of how humans use holistic and local features. Specifically, the following categorization was offered:

- *Holistic matching methods.* Holistic methods use the whole face region as the raw input to a recognition system. One of the most widely used representations of the face region is eigenpictures [56] which are based on Principal Component Analysis.
- *Feature-based (structural) matching methods.* These methods typically utilize local features of the face, their locations and local statistics. Popular facial features are eyes, nose, and mouth.
- *Hybrid methods.* In this approach both local features and the whole face region are used to recognize a face. These methods could potentially offer the best of the two types of the above methods. The idea of hybrid approach is the most similar to the human perception system.

Holistic matching methods can be subdivided into two groups: *statistical* and *Artificial Intelligence (AI) approaches* [49].

Statistical holistic matching methods. In the naive version of the holistic approaches, the image is simply represented as a 2D array of intensity values and recognition is performed by direct comparisons between the input face and all the other faces in the database. The classification in performed in a space of very high dimensionality. Obviously this approach is computationally very expensive and can operate under very limited circumstances.

Authors in [56] were the first to utilize Principal Components Analysis (PCA) to economically represent face images. They demonstrated that any face can be efficiently represented along the eigenpictures coordinate space. Any face can then be approximately reconstructed by using just a small collection of eigenpictures and the corresponding coefficients along each eigenpicture.

Authors in [112] realized that projections along eigenpictures in [56] could be used as features to recognize faces. They introduced Eigenfaces that correspond to the eigenvectors associated with the highest eigenvalues of the known face (patterns) covariance matrix. The faces

are recognized by comparing the projections along the eigenfaces to those of the face images of the known individuals. This approach drastically reduces the dimensionality of the original space. Authors show that the method appears to be fairly robust to lighting variations but its performance degrades with scale changes. The research have been lated extended in [91] and tested on a large database.

Authors in [8] propose the use of Fisher's Linear Discriminant Analysis in face recognition, which maximizes the ratio of the between-class scatter and the within-class scatter and performs better for classification than PCA. However, some recent work [75] shows that when the training data set is small, PCA can outperform LDA and also that PCA is less sensitive to different training sets.

An alternative approach which utilizes difference images is proposed in [80]. The difference image for two face images is defined as the signed arithmetic difference in the intensity values of the corresponding pixels in those images. Two classes of difference images are introduced: intrapersonal (for the same person) and extrapersonal (for different people). It is assumed that both these classes originate from discrete Gaussian distributions within the space of all possible difference images. The probability that the difference image belongs to the intrapersonal class is given by Bayes Rule.

Significant amount of extensions to the standard Eigenfaces and the Fisherfaces approaches have been developed since their introduction. Some of these examples are proposed in [127] and [106] for PCA-based methods and in [133] and [132] for LDA-based approaches. All these methods purportedly obtain better recognition results than the original techniques.

AI-based holistic matching methods. AI approaches utilize various tools of machine learning, such as Neural Networks and and Support Vector Machines, to recognize faces. Some examples of methods that belong to this category are given below.

Authors in [63] reported a 96.2 % recognition rate on the ORL database [2] using a hybrid neural network solution. The combination includes local image sampling, a self-organizing map [57] neural network for dimensionality reduction and invariance to small changes in the image sample, and a convolutional neural network, which provides partial invariance to translation, rotation, scale and deformation. The Eigenfaces method achieved 89.5 % recognition accuracy for the same data set.

A system in which a face image is first decomposed with a wavelet transform to three levels is introduced in [65]. The Fisherfaces method is then applied to each of the three low-frequency sub-images. The individual classifiers are then fused using the RBF neural network. The system was tested on the FERET database and was shown to outperform the individual classifiers and the direct Fisherfaces method.

SVM has also been utilized for face recognition by many researchers and has been shown to yield good results [66], [33].

Feature-based (structural) matching methods first process the input image to detect and extract discriminative facial features such as the eyes, mouth, nose, as well as other fiducial

marks. Then the geometric relationships among those facial points are computed, thus reducing the input facial image to a vector of geometric features. Standard statistical pattern recognition techniques are then employed to recognize faces using these measurements.

Geometrical parameters of the face are intuitive for our perception, thus early face recognition approaches was mostly based on these techniques. The first significant example is introduced in [54]. The author employed simple image processing methods to extract a vector of 16 facial parameters - which were ratios of distances, areas and angles and used a simple Euclidean distance measure for matching. A peak performance of 75 % was achieved on a database with 20 different people using 2 images per person (one for reference and one for testing).

Later, authors in [30] reported a recognition performance of 95 % on a more challenging database of 685 images (a single image for each individual) using a 30-dimensional feature vector. The disadvantage of the method is that the facial features were manually extracted, so it is reasonable to assume that the recognition performance would degrade significantly if an automated feature extraction method had been adopted. In general, current algorithms for automatic feature extraction do not provide a high degree of accuracy and require considerable computational capacity [63].

Another popular feature-based approach is the elastic bunch graph matching method proposed in [122]. A graph for an individual face is generated as follows: a set of fiducial points on the face are chosen. Each fiducial point is a node of a full connected graph, and is labeled with the Gabor filters responses applied to a window around the fiducial point. Each arch is labeled with the distance between the correspondent fiducial points. A representative set of such graphs is combined into a face bunch graph. Once the system has a face bunch graph, graphs for new face images can then be generated automatically by Elastic Bunch Graph Matching. Recognition is performed by comparing the graph of the face to those of all the known face images and picking the one with the highest similarity value. Though the mentioned method was among the best performing ones in the most recent FERET evaluation [93], it does suffer from the serious issue of requiring the graph placement for the first 70 faces to be done manually before the elastic graph matching becomes adequately dependable.

The advantage of the feature-based schemes is relative invariance to size, orientation and/or lighting conditions [30]. Other benefits include the compactness of representation of the face images and high speed matching due to low dimensionality of the feature space. The major disadvantage of these approaches is the computational complexity of automatic feature detection and the loss of textural information about the face.

Hybrid methods use both holistic and local features. For example, the modular eigenfaces approach in [91] uses both global eigenfaces and local eigenfeatures.

An interesting appearance model based method for automatic face recognition was introduced in [61]. To identify a face, both shape and gray-level information are used. The shape is incorporated in the form of Active Shape Model. For an input image all three types of information, including extracted shape parameters, shape-free image parameters, and local profiles, are

used to compute a Mahalanobis distance for classification.

Another significant approach in this field is introduced in [48]. It is based on principles of component-based detection/recognition [47] and 3D morphable models [16]. The SVM classifier is used in the recognition process.

5.2 Local Binary Patterns based face recognition

Authors in [9] applied the LBP histograms in face recognition and achieved promising results on the FERET database. The developed histogram-based representation contained information on three different levels: pixel level - encoded in the form of LBP labels; region/block level - LBP histograms of face regions; image level - introduced in the form of spatially enhanced histograms. Authors also described the idea of unequal importance of different face regions in the recognition process and introduced an empirical method to assign weights to each region. For classification, a nearest neighbor classifier was used with Chi-square dissimilarity measure.

Following the idea of [9] authors in [129] underlined some limitations of the method. First, the position and size of each region are fixed which limits the size of the feature space. Second, the weighting technique is empirical and thus not optimal. To overcome these limitations, they propose to shift and scale a scanning window over pairs of images, extract the local LBP histograms and compute the dissimilarity measure between the corresponding local histograms. The dissimilarity measures are labeled as positive features if both images are from the same identity and as negative otherwise. Classification is performed with AdaBoost learning, which solves the feature selection and classifier design problem. The boosting procedure is utilized in the process of optimal selection of the position, size and weight of the region. Comparative studies with method in [9] on the **(fa/fb)** protocol of the FERET database showed similar results in accuracy but, since fewer regional histograms are used, the dimensionality of the representation space is lower. However, the amount of training time in this approach is prohibitive.

A lot of extensions of the LBP paradigm were developed since it was introduced, for example Multi-scale Block Local Binary Patterns (MB-LBP) [68]. The MB-LBP are computed based on average values of block subregions, instead of individual pixels. Authors claim that MB-LBP code has a number of advantages if compared to LBP: it is more robust than LBP; it encodes not only micro-structures but also macro-structures of image patterns and hence provides a more complete image representation. The computation of MB-LBP is still effective using integral images. Finally, authors apply AdaBoost learning in order to select most effective uniform MB-LBP features and construct face classifiers. Experiments on Face Recognition Grand Challenge database show that the proposed MB-LBP method significantly outperforms LBP based face recognition algorithms.

Another extension, namely Multi-scale Local Binary Patterns (MSLBP) is introduced in [24]. The representation of the face is derived by the Linear Discriminant Analysis (LDA) of multi-scale local binary pattern histograms. In each non-overlapping face region, multi-scale

local binary uniform pattern histograms are extracted and concatenated into a regional feature. The features are then projected on the LDA space to be used as a discriminative facial descriptor. Authors presented the recognition results for the FERET and XM2VTS databases.

5.2.1 Face recognition based on Weighted Local Binary Pattern histograms

Proposed face recognition algorithm is based on the LBP transformation of the input face image. The spatially enhanced histogram of the labeled image $\boldsymbol{I}_L$ (Equation (2.3)) effectively represents both global and regional features of the face, however some advanced optimization and preprocessing steps are needed in order to get high performance face recognition system.

The first phase of the proposed face recognition algorithm is the preprocessing of the input gray-scale image, which consists of two steps:

- input image rotation. This stage performs image upright rotation if the deviation of the eye line from the horizontal $\alpha_{eyeline}$ exceeds the predetermined threshold. Positions of eye pupils $\tilde{C}_L$ and $\tilde{C}_R$ are determined in the previous stages of automatic face recognition algorithm;

- face region extraction. The region of the face in the input image is limited by the bounding box, which is determined by four coordinates, see Figure 3.12 for details. In contrast to face detection task the bounding box is enhanced and is now calculated as follows:

$$
\begin{aligned}
X_{start} &= \max\left(\{1, \text{round}\left(X(C_R) - 0.9 \cdot d_{eye}\right)\}\right), \\
X_{end} &= \min\left(\{X_{max}, \text{round}\left(X(C_L) + 0.9 \cdot d_{eye}\right)\}\right), \\
Y_{start} &= \max\left(\left\{1, \text{round}\left(\frac{Y(C_R) + Y(C_L)}{2} - 1.4 \cdot d_{eye}\right)\right\}\right), \\
Y_{end} &= \min\left(\left\{Y_{max}, \text{round}\left(\frac{Y(C_R) + Y(C_L)}{2} + 1.9 \cdot d_{eye}\right)\right\}\right),
\end{aligned}
\tag{5.1}
$$

where the notations $X(C_R)$ and $X(C_L)$ stand for the X coordinates of the left and right eye pupils in the *rotated* input image and $Y(C_R)$, $Y(C_L)$ are the corresponding Y coordinates, see Figure 5.1 for details; X_{max} and Y_{max} are the width and hight of the input image in pixels. The examples of enhanced facial images that are utilized in the recognition process are displayed in the Figure 5.2.

Once preprocessing steps are completed an LBP transformation is performed as illustrated in Figure 2.2. The transformed image is divided into $K \times K$ regions to save the spatial information about the object. The LBP histogram is calculated for each region according to the Equation 2.1. Region histograms are stacked sequentially into a single face feature histogram, Figure 2.2. The length of the face feature vector is equal to: $N = K^2 \cdot 2^P$.

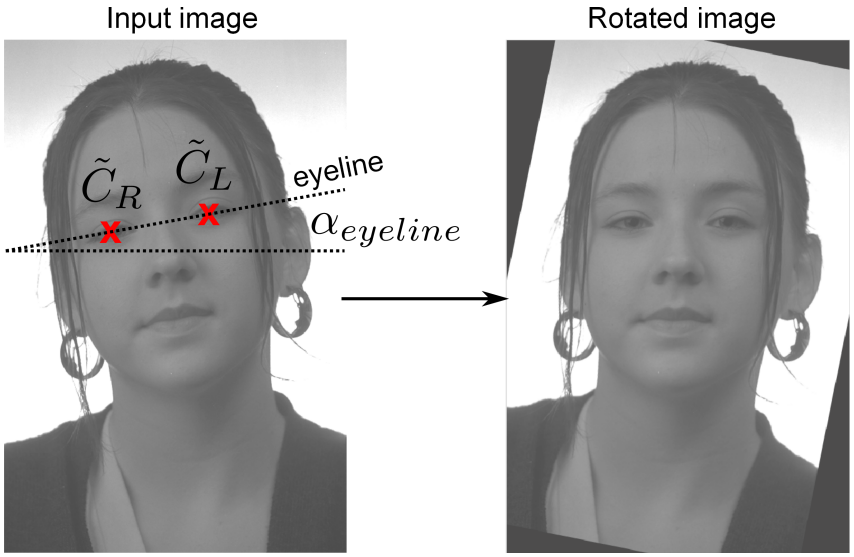


Figure 5.1: Rotation of the input face image by the angle $\alpha_{eyeline}$

Figure 5.2: An example of face images used in the face recognition algorithms

In real life systems the scales of captured objects are different. Therefore the normalization of spatially enhanced LBP histogram is performed according to the Equation (2.5) in order to get a coherent description.

For pattern classification in the face recognition tasks a nearest neighbor classifier is usually used [9]. Among the most popular approaches for similarity/dissimilarity measures are Histogram intersection, Chi-square statistics and Squared Euclidean distance.

A usual problem in face recognition is having plenty of classes and only a small number, possibly even one, training sample per class [9]. This fact degrades the usability of complicated classifiers, such as Neural Networks or SVM, in face recognition. For this reason, we have developed a new learning algorithm, which iteratively adjusts the weights in the wighted nearest neighbor classifier. The introduced weighting algorithm compensates two important classification problems:

- lack of statistical information about the data, which is typical for the nearest neighbor classifier;
- lack of intra-class information, which is needed for sophisticated classifier learning.

The weighting principle is enhanced for the optimization in two levels: block-level weight-

ing and feature-level weighting, see Figure 2.6 for details. These principles are discussed in Section 5.3.

After the learning of the weights in WNNC is completed the weighted histogram intersection is used for similarity measure between the input pattern and face images that are stored in the database. The extensions of histogram intersection to the weighted form are presented in Equations (2.18) and (2.19). According to our experiments histogram intersection is the most robust method for similarity measure and constantly provides the best recognition results.

5.2.2 Face recognition based on Weighted Multi-scale Local Binary Pattern histograms

The Multi-scale Local Binary Pattern based face recognition process is very similar to the one described in Section 5.2.1. The main difference is in the descriptor of the face which is now a Multi-scale Local Binary Pattern histogram. Some modifications are also introduced in the preprocessing stage. The main goal of this section is to achieve a better stability of the descriptor for different scales of the object / face in our case. The approach is based on the observation, that the texture of the material varies for different magnification factors. LBP operator was originally introduced, as a texture descriptor and the result of the LBP transformation clearly depends on the scale of the object. Introduced MSLBP principle partially resolves this challenging aspect. This issue is also addressed in [68], where another extension of the LBP, called Multi-scale Block Local Binary Patterns (MB-LBP), is introduced. In MB-LBP, the comparison between single pixels in LBP is simply replaced with the comparison between average gray-values of the square sub-regions containing neighborhood pixels. This idea makes the descriptor more robust and reduces the noise.

Inspired by the idea of MB-LBP the **mean filtering** [104] of the input face image is performed before further processing. This preprocessing step reduces the negative influence of the texture scale on the stability of the MSLBP histograms. Mean filtering provides slightly better results than average filtering introduced in MB-LBP [68].

Next, the rotation of the input image (see Figure 5.1 for details) and the detection of face region (Equation (5.1)) are performed. These steps are exactly the same as in the weighted LBP based face recognition and are discussed in more details in Section 5.2.1.

The MSLBP transformation of the preprocessed face image is performed next. In fact MSLBP is the same as LBP transformation with various values of the parameter R, see Figures 2.3 and 2.4 for details. The transformed images (Figure 2.3) are divided into $K \times K$ regions in order to save the spatial information about the object. The LBP histogram is calculated for each region. Obtained regional histograms are stacked sequentially into the corresponding feature histograms: $\boldsymbol{h}^{(1)}, \boldsymbol{h}^{(2)}, \ldots, \boldsymbol{h}^{(n_R)}$. The resulting MSLBP histogram is calculated next according to the Equation (2.6). The process of spatially enhanced LBP histogram calculation is displayed in Figure 2.2, in case of MSLBP this process is performed n_R times.

The WNNC methodology is applied in the recognition stage of the algorithm. An iterative approach for the adjustment of the weights in WNNC is described in the next sections. The weighting principle is enhanced for the optimization in two levels: block-level weighting and feature-level weighting, see Figure 2.6 for details.

After the learning of the weights in WNNC is completed the weighted histogram intersection is used for similarity measure between the input pattern and face images that are stored in the database. The extensions of histogram intersection to the weighted form are presented in Equations (2.18) and (2.19). According to our experiments histogram intersection is the most robust method for similarity measure and constantly provides the best recognition results.

5.3 Discriminative feature weighting

A challenging aspect in the field of face recognition is the design of the classifier. The usual identification approaches are based on various Nearest Neighbor Classifiers (NNC) [9], [85] (Nikisins et al.). The use of complicated classifiers, such as Artificial Neural Networks or Support Vector Machine, is often inconvenient or even impossible due to insufficient intra-class information and significant amount of classes / individuals in the database. An algorithm for the weighting of discriminative features (DFW) is developed in [84] (Nikisins et al.) in order to compensate the statistical incompleteness of the NNC by utilizing the information from all classes. Similar approaches of weighting the components in the feature vector are discussed in [31] and [110], however these methods still require significant intra-class data, while the methodology in [84] (Nikisins et al.) needs only two training examples per class. An extension of the DFW principle [84] (Nikisins et al.) is also proposed in this research. The extended version of the algorithm is more stable, predictable and provides better recognition results. The improvement is based on the special procedure of the learning data selection, which allows to maximize the distance between feature vectors, which are most likely to cause the misclassification. The information obtained in the process of weights learning is incorporated in the recognition process by the use of weighted nearest neighbor classifier (WNNC). The DFW principles are utilized in two levels: block-level and feature-level weighting.

The reduction of the learning time is another challenging aspect. This issue is very important in the cases of massive training data sets and highly dimensional feature vectors. Both of these aspects are usually true for biometric applications. This problem is resolved by the introduction of mini-batch principle, which makes the proposed training methodology comparatively fast. The proposed principle of optimal learning data selection slightly degrades the speed of the learning process, but the selection procedure is not involved on every iteration of the algorithm and the final increase of the learning time is not significant.

5.3.1 A mini-batch discriminative feature weighting algorithm in the feature-level

Not all components in the feature vector are equally important during the recognition process. An obvious idea of the proposed algorithm is to enhance/degrade features according to their discriminative importance, see Figure 2.6 (a).

Let $\boldsymbol{w} = (w_1, w_2, \ldots, w_N)$ be the set of weights to be adjusted during the optimization process. For training purposes two data sets are required: $\boldsymbol{X}^{(1)} \in \mathbb{R}^{M \times N}$ and $\boldsymbol{X}^{(2)} \in \mathbb{R}^{M \times N}$, where upper indexes stand for the number of training example, that belongs to the same class, so only two training examples per class are needed; M is the number of classes / persons and N is the length of the feature vector. Rows i in matrices $\boldsymbol{X}^{(1)}$ and $\boldsymbol{X}^{(2)}$ represent two face histograms of the same class: $\boldsymbol{x}_i^{(1)}$ and $\boldsymbol{x}_i^{(2)}$ respectively.

A diagonal matrix $\boldsymbol{V} = \{v_{i,j}\}$ is formed form the elements of vector $\boldsymbol{w}$ in order to perform a linear diagonal transformation of the training data sets:

$$v_{i,j} = \begin{cases} w_i & \text{if } i = j \\ 0 & \text{if } i \neq j. \end{cases} \tag{5.2}$$

$$\begin{aligned} \tilde{\boldsymbol{X}}^{(1)} &= \boldsymbol{X}^{(1)}\boldsymbol{V}, \\ \tilde{\boldsymbol{X}}^{(2)} &= \boldsymbol{X}^{(2)}\boldsymbol{V}. \end{aligned} \tag{5.3}$$

Rows $\tilde{\boldsymbol{x}}^{(1)}$ and $\tilde{\boldsymbol{x}}^{(2)}$ in matrices $\tilde{\boldsymbol{X}}^{(1)}$ and $\tilde{\boldsymbol{X}}^{(2)}$ represent weighted face histograms.

The idea of the optimization process is based on maximization / minimization of the inter-class / intra-class Squared Euclidean distance. Let's introduce the following notations: d_i^{intra} - the value of Squared Euclidean distance between two intra-class histograms $\tilde{\boldsymbol{x}}_i^{(1)}$ and $\tilde{\boldsymbol{x}}_i^{(2)}$, and $d_{i,j}^{inter}$ - the value of Squared Euclidean distance between two inter-class histograms $\tilde{\boldsymbol{x}}_i^{(1)}$ and $\tilde{\boldsymbol{x}}_j^{(2)}$, where $i \neq j$.

If $g_{i,j} = \frac{d_i^{intra}}{d_{i,j}^{inter}} - 1$, then the following statements are true for all combinations of histograms:

$$g_{i,j} < 0, \text{if classification is correct,}$$

$$g_{i,j} \geq 0, \text{if classification is incorrect.}$$

The number of possible $d_{i,j}^{inter}$ values for each i is equal to $(M-1)$. A matrix $\boldsymbol{G}$ can be formed from elements $g_{i,j}$:

$$\boldsymbol{G} = \{g_{i,j}\}, i = 1, \ldots, M, j = 1, \ldots, M-1.$$

At this stage a **mini-batch** principle is introduced. Instead of using all elements of $\boldsymbol{G}$ in further computations, only one $g_{i,j}$ value is randomly selected for each class i in order to accelerate the

learning process and to avoid computational problems:

$$\boldsymbol{g} = \{g_{i,j=rand(1)} | j \neq i, j \in [1, M]\}, i = 1, \ldots, M, \tag{5.4}$$

where $rand(1)$ stands for a single positive randomly selected integer number. The random selection principle can be replaced with more robust learning data selection algorithm, which is described in the next sections.

The proposed cost function can be constructed as follows:

$$J = \frac{1}{M} \sum_{i=1}^{M} c_i, \text{where} \tag{5.5}$$

$$c_i = \frac{1}{1 + \exp(-\alpha g_i)}, \tag{5.6}$$

where g_i are the elements of vector $\boldsymbol{g}$: $\boldsymbol{g} = \{g_i\}, i = 1, \ldots, M$ and α determines the degree of correct classification influence on the magnitude of the cost function. In the case of a batch algorithm the number of summands in the (5.5) is equal to $M(M-1)$. A mini-batch principle reduces the number of summands till M, which is called the batch size. This fact makes the learning algorithm much faster, which is very important in case of a big database.

The next step of the proposed algorithm is based on the iterative re-estimation of weights in the $\boldsymbol{w}$ set in order to minimize a cost function J representing the classification error. A gradient descent method is used to update the parameters w_i at each iteration k:

$$w_{i,k} = w_{i,k-1} - \eta \frac{\partial J}{\partial w_i}|_{w_{k-1}}, \tag{5.7}$$

where η is a learning rate. Partial derivative $\partial J / \partial w_i$ in the (5.7) can be written as follows:

$$\frac{\partial J}{\partial w_i} = \frac{1}{M} \sum_{j=1}^{M} \frac{\partial c_j}{\partial w_i} \tag{5.8}$$

$$\frac{\partial c_j}{\partial w_i} = \frac{\partial c_j}{\partial g_j} \frac{\partial g_j}{\partial w_i} \tag{5.9}$$

According to the (5.6) $\partial c_j / \partial g_j$ can be written:

$$\frac{\partial c_j}{\partial g_j} = \alpha c_j (1 - c_j) \tag{5.10}$$

The partial derivatives $\partial g_j / \partial w_i$ can be obtained from the definition of the proposed mini-batch algorithm:

$$\frac{\partial g_j}{\partial w_i} = \frac{2w_i(S^{(1)}_{j,i}S^{(4)}_{j,i} - S^{(2)}_{j,i}S^{(3)}_{j,i})}{(S^{(3)}_{j,i}w_i^2 + S^{(4)}_{j,i})^2}, \text{where} \tag{5.11}$$

$$S^{(1)}_{j,i} = (x^{(1)}_{j,i} - x^{(2)}_{j,i})^2,$$

$$S^{(2)}_{j,i} = \sum_{n=1,n\neq i}^{N} w_n^2(x^{(1)}_{j,n} - x^{(2)}_{j,n})^2,$$

$$S^{(3)}_{j,i} = (x^{(1)}_{j,i} - x^{(2)}_{k,i})^2,$$

$$S^{(4)}_{j,i} = \sum_{n=1,n\neq i}^{N} w_n^2(x^{(1)}_{j,n} - x^{(2)}_{k,n})^2,$$

$$k = rand(1)|k \neq j.$$

5.3.2 A mini-batch discriminative feature weighting algorithm in the block-level

The methodology of block weighting is in many aspects similar to the technique of feature weighting. This idea was originally introduced in [9] and comes from an obvious guess, that not all face features are equally important during the recognition process. For example, eyes seem to be an important component in the face recognition, while mouth region is strongly dependent on facial expressions [134]. The disadvantage of the idea in [9] is a semi-manual adjustment of the weights, while the proposed methodology is based on the automatic learning process.

Block weighting is schematically displayed in Figure 2.6 (b). Let $\boldsymbol{w} = (w_1, w_2, \ldots, w_m)$ be the set of the weights for each block, where $m = K^2$ is the total number of regions, see Figure 2.2. Previously described sets $\boldsymbol{X}^{(1)}$ and $\boldsymbol{X}^{(2)}$ are needed for the training stage of the algorithm.

In order to perform a block-level weighting a diagonal matrix $\boldsymbol{V} = \{v_{i,j}\}$ is formed from the elements of vector $\boldsymbol{w}$:

$$v_{i,j} = \begin{cases} w_l, \text{ if } i = j = \{n(l-1)+1, \ldots, nl | l = 1, \ldots, m\}, \\ 0, \text{ if } i \neq j, \end{cases}$$

where $n = N/m$ is the number of features per block. A linear diagonal transformation of the training data sets is performed according to the Equation (5.3). Rows $\tilde{\boldsymbol{x}}^{(1)}$ and $\tilde{\boldsymbol{x}}^{(2)}$ in matrices $\tilde{\boldsymbol{X}}^{(1)}$ and $\tilde{\boldsymbol{X}}^{(2)}$ represent the face histograms, which are weighted in a block level.

An iterative method is used to update the weights of each block, Equations (5.7) - (5.11). Elements $S^{(1)}_{j,i}, S^{(2)}_{j,i}, S^{(3)}_{j,i}$ and $S^{(4)}_{j,i}$ in the (5.11) are determined as follows:

$$S^{(1)}_{j,i} = \sum_{l=n(i-1)+1}^{ni} (x^{(1)}_{j,l} - x^{(2)}_{j,l})^2,$$

$$S_{j,i}^{(2)} = \sum_{r=1, r\neq i}^{m} w_r^2 \sum_{l=n(r-1)+1}^{nr} (x_{j,l}^{(1)} - x_{j,l}^{(2)})^2,$$

$$S_{j,i}^{(3)} = \sum_{l=n(i-1)+1}^{ni} (x_{j,l}^{(1)} - x_{k,l}^{(2)})^2,$$

$$S_{j,i}^{(4)} = \sum_{r=1, r\neq i}^{m} w_r^2 \sum_{l=n(r-1)+1}^{nr} (x_{j,l}^{(1)} - x_{k,l}^{(2)})^2,$$

$$k = rand(1) | k \neq j.$$

5.3.3 Stabilized learning data selection algorithm

Suppose, that d_i^{intra} is the value of Squared Euclidean distance between two weighted intra-class (same person) histograms $\tilde{\boldsymbol{x}}_i^{(1)}$ and $\tilde{\boldsymbol{x}}_i^{(2)}$, $i = 1, \ldots, M$ - is the number of the class / person and $d_{i,j}^{inter}$ - the value of Squared Euclidean distance between two inter-class (different persons) histograms $\tilde{\boldsymbol{x}}_i^{(1)}$ and $\tilde{\boldsymbol{x}}_j^{(2)}$ with constraint $i \neq j$.

The previously described principle is based on iterative re-estimation of the block (or feature) weights $\boldsymbol{w} = (w_1, w_2, \ldots, w_m)$ in order to minimize d_i^{intra} and maximize $d_{i,j}^{inter}$ for all classes $i = 1, \ldots, M$. For this purpose the corresponding pairs (d_i^{intra}, $d_{i,j}^{inter}$) for all classes i are utilized at each iteration of the re-estimation process, where j is selected *randomly*. The random selection of j implements a *mini-batch* principle in the learning process with intent to accelerate it, but also makes it noisy and slightly unstable.

Inspired by the ideas of neocortex simulation introduced in [44] a novel methodology of the learning data selection is introduced in this paper. The introduced principle is based on the assumption, that the sequence of the inflow of the learning data is important. The *random* selection of the data can be replaced with the special selection procedure, which is performed after each δ_{iter} iterations of the learning algorithm: $\boldsymbol{n}_{iter} = (1, i \cdot \delta_{iter}), i = 1, \ldots, N_{iter}/\delta_{iter}$, where $\boldsymbol{n}_{iter}$ - vector with the iteration values for which the data selection procedure is performed, N_{iter} - total number of the iterations. For the iteration from the $\boldsymbol{n}_{iter}$ set the *closest* (not random [84] (Nikisins et al.)) histogram $\tilde{\boldsymbol{x}}_j^{(2)}$ is determined for each $\tilde{\boldsymbol{x}}_i^{(1)}$ with constraint $i \neq j$. These histograms are utilized for the calculation of the (d_i^{intra}, $d_{i,j}^{inter}$) pairs and are used in the learning process for the next δ_{iter} iterations. When the next value of the iteration from the $\boldsymbol{n}_{iter}$ set is achieved the estimation of closest histograms is performed again. The Euclidean distance is used for similarity measure. An intuitive explanation of the proposed idea is simple: to maximize the distance between the histograms, which are *most likely* to cause the misclassification (face recognition errors). The proposed principle of learning data selection makes the learning process more stable and predictable, while a *mini-batch* principle and the computational efficiency of the learning algorithm are still represented.

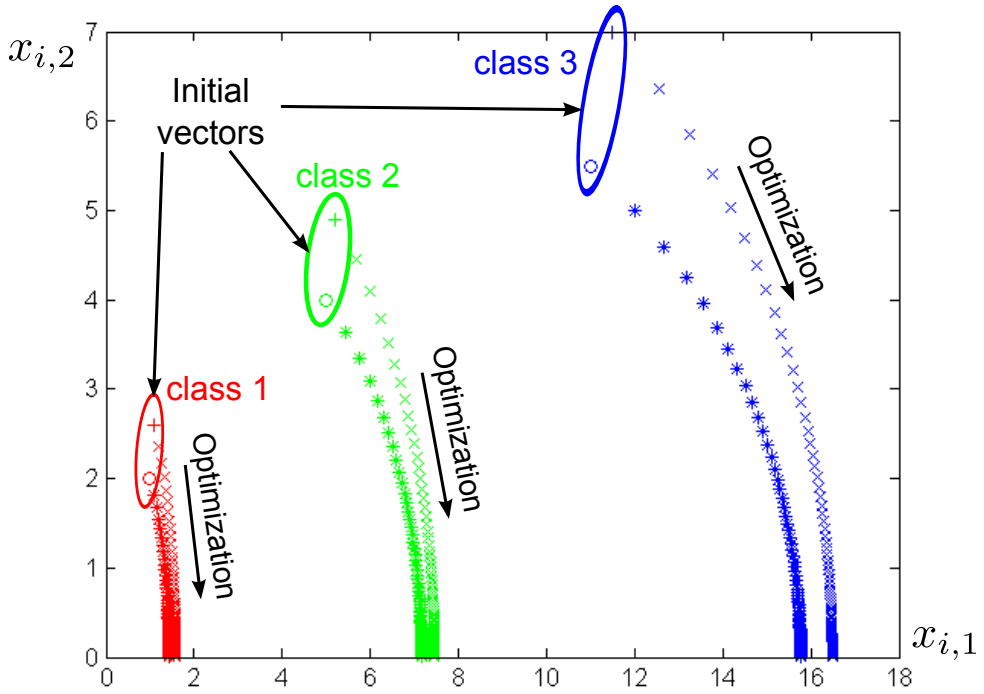


Figure 5.3: Visualization of optimization path of discriminative feature weighting algorithm for 3-class example in 2D feature space

5.3.4 Visual interpretation of the learning process: simple example

An intuitive explanation of proposed discriminative feature weighting algorithm is covered in this subsection. Let's consider a very simple classification example with only three classes and each feature vector has only two parameters/coordinates, thus $M = 3$ and $N = 2$. Each feature vector can be represented as a point in 2D space, see Figure 5.3 for details (points with "o" and "+" markers). For each class only two training examples are given: $\boldsymbol{x}_i^{(1)}$ and $\boldsymbol{x}_i^{(2)}$, where $i = (1, 2, \ldots, M)$ is the number of the class.

The introduced mini-batch discriminative feature weighting algorithm is applied next in the feature-level. In other terms the weights (w_1, w_2) of each parameter $(x_{i,1}, x_{i,2})$ of the feature vectors are iteratively adjusted in order to minimize the cost function in the Equation (5.5). The purpose of the algorithm is to locate the intra-class (same class) training examples as close as possible along with keeping the inter-class (different classes) feature vectors as far as possible. The *initial* intra-class (same class) training examples in the Figure 5.3 are *mostly* scattered from each other in the vertical direction (along $x_{i,2}$ coordinate axis; the class-bounding ellipses are stretched vertically). Thus, the main classification error is incorporated by parameter $x_{i,2}$. The initial weights in the example (Figure 5.3) were $(w_1 = 1, w_2 = 1)$, and after 50 iterations of the weights adjustment algorithm the new values are $(w_1 = 1.43, w_2 = 0)$. The algorithm projected all feature vectors on the horizontal axis $x_{i,1}$ and the error-prone parameter $x_{i,2}$ is excluded from the recognition process (the weight w_2 is set to zero). The optimization path in the Figure 5.3 is displayed with "*" and "x" markers.

Let's consider a more complicated classification example with five classes and 2D feature

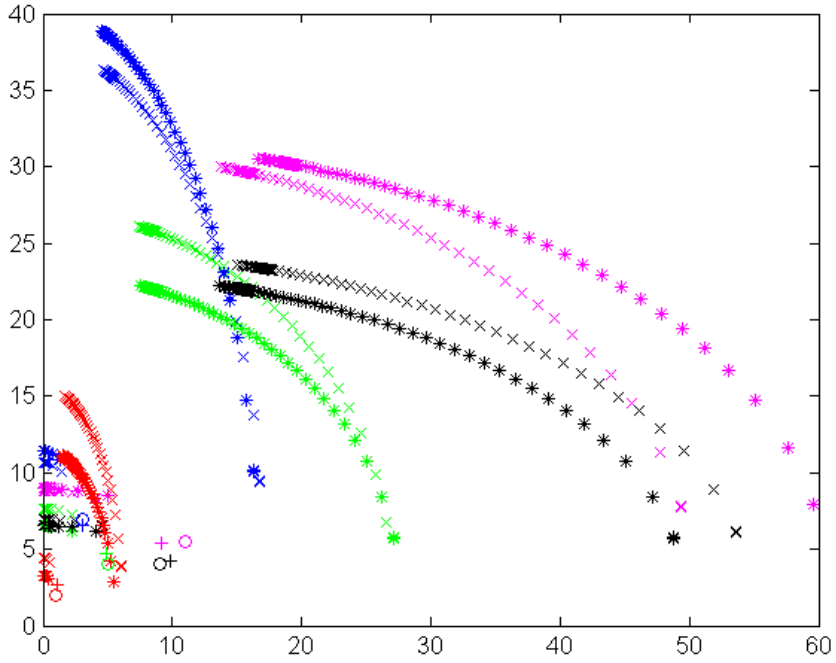


Figure 5.4: Visualization of optimization path of discriminative feature weighting algorithm for 5-class example in 2D feature space

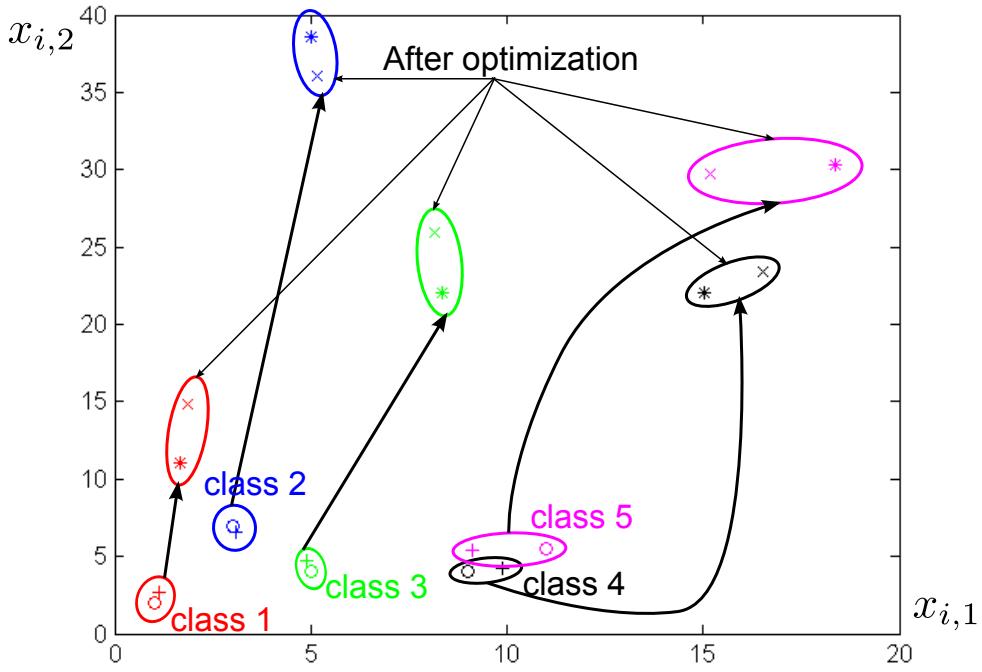


Figure 5.5: Visualization of the result of discriminative feature weighting algorithm for 5-class example in 2D feature space

vectors, thus $M = 5$ and $N = 2$. Still only two training examples are given for each class: $\boldsymbol{x}_i^{(1)}$ and $\boldsymbol{x}_i^{(2)}$, where $i = (1, 2, \ldots, 5)$ is the number of the class. Additionally, the *initial* intra-class (same class) training examples in the Figure 5.5 (points with "o" and "+" markers) are now scattered from each other both in vertical and horizontal directions (the class-bounding

ellipses are stretched vertically for classes 1,2,3 and horizontally for classes 4,5). Thus, the trivial solution with one of the weights equal to zero is not possible.

Again, a mini-batch discriminative feature weighting algorithm is applied in the feature-level. The initial weights in the example (Figure 5.5) were $(w_1 = 1, w_2 = 1)$, and after 200 iterations of the weights adjustment algorithm the new values are $(w_1 = 1.67, w_2 = 5.51)$. The optimization path which is marked with symbols "*" and "x" in the Figure 5.4 is now much more complicated and difficult for perception. However the result of optimization is clear, see Figure 5.5 for details. Suppose, that in the *initial state* samples which are marked with "+" symbol are stored in the database (*gallery* set) and "o"-marked samples are unknown feature vectors that are presented to the classification algorithm (*probe* set). If NNC classifier is used then the *initial* classification/recognition precision equals $P(w_1 = 1, w_2 = 1) = 3/5 = 60\%$. The patterns in the classes 4 and 5 were classified incorrectly. After the weighting is completed the *gallery* set is marked with "x" symbols and the *probe* set is marked with "*". In this case the classification precision becomes $P(w_1 = 1.67, w_2 = 5.51) = 5/5 = 100\%$. The feature vectors in classes 1,2 and 3 are now more scattered, while the opposite statement is true for classes 4 and 5. The gain in the performance of the classification process is obvious: $P(\text{NNC}) = 60\%$ and $P(\text{WNNC}) = 100\%$.

5.4 Face recognition: performance evaluation and experimental setup

Evaluation of the proposed face recognition methodology is performed on a **color FERET** database [1]. The standard subsets **fa** and **fb** (frontal face images) are selected from the color FERET database. Usually two separate datasets, namely *gallery set* and *probe set* are needed for performance evaluation. The *gallery set* contains the data of known individuals. An image of an unknown face presented to the algorithm is called a *probe*, and the collection of probes is called the *probe set*. The subset **fa** is used as a *gallery set* and **fb** as a *probe set* [93]. The total number of classes / persons in the database is $M = 993$, and individuals were asked for a different facial expressions in **fa** and **fb** sets.

The basic approaches for evaluating the performance of an algorithm are the *closed* and *open universes*. In an *open universe*, some probes are not in the gallery. The open universe model is used to evaluate verification applications.

In our case a **closed universe model** is selected for the evaluation of the algorithm performance [93]. In a closed universe identification every probe image has a corresponding matching template in the database. This assumption allows to determine the ability of the algorithm to identify a probe image. The results are usually represented in the form of **Cumulative Match Characteristics** (CMC), where the horizontal axis is **rank** and the vertical is the **probability of correct identification** P_I. The probability of correct identification at rank n means that the

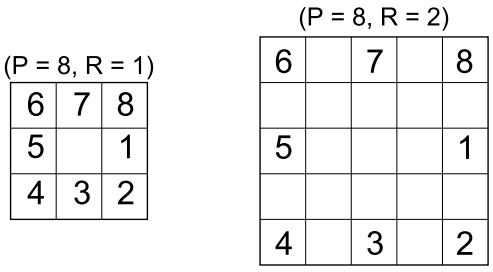


Figure 5.6: LBP operators with $P = 8$ and $R = (1, 2)$

correct match is somewhere in the top n similarity scores with corresponding probability. The probability at rank one $P_I(r = 1)$ is the parameter, which is often used in the literature to compare the performance of different algorithms. Such approach answers not only the question "is the top match correct?" but also "is the correct answer in the top n matches?". This lets one know how many images have to be examined to get a desired level of performance. The size of the gallery set and the number of probes should be stated in this approach. In our case the sizes of probe and gallery sets are similar and are equal to M.

5.5 Simulation results

Evaluation of the proposed face recognition algorithms is performed on a color FERET database. Introduced algorithms are based on the LBP and MSLBP transformations. The quasi-optimal values of the parameters of LBP and MSLBP operators are evaluated in Sections 5.5.1 and 5.5.4. The aspects of WNNC optimization are covered in Sections 5.5.2 - 5.5.3 for LBP - based approach and in Section 5.5.5 for MSLBP setup. PCA-based data compression is described in Section 5.5.6.

5.5.1 Evaluation of parameters for LBP-based face recognition

Many parameters of the proposed face recognition methodology requires an optimization in order to get high performance system. The first step of the estimation process evaluates the best R value for the LBP operator with parameters $(P = 8, R)$, see Figure 5.6.

The introduced methodology for the evaluation of R is based on the following equation:

$$F(R) = \frac{(M-1)\sum_{i=1}^{M} d(x(R)_i^{(1)}, x(R)_i^{(2)})}{\sum_{i=1}^{M}\sum_{j=1,j\neq i}^{M} d(x(R)_i^{(1)}, x(R)_j^{(2)})}, \tag{5.12}$$

where the notation $d(x(R)_i^{(1)}, x(R)_i^{(2)})$ stands for the Squared Euclidean distance between two intra-class face histograms and $d(x(R)_i^{(1)}, x(R)_j^{(2)})$ is the Squared Euclidean distance for inter-class histograms. In this case LBP histograms $x(R)$ are calculated **without** regioning (see Figure 2.2) and depend only on parameter R. In general $F(R)$ represents the ratio between average intra

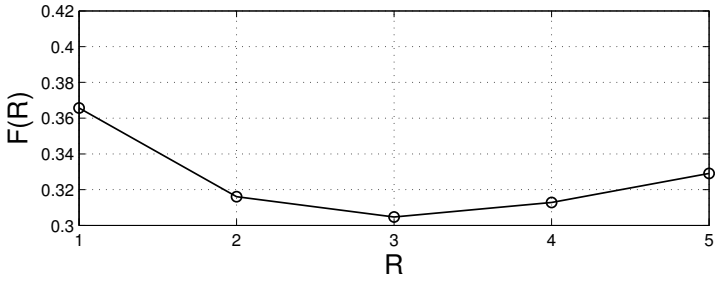


Figure 5.7: $F(R)$ for the subsets **fa** and **fb** of color FERET

and inter class face histograms.

The dependence $F(R)$ for the subsets **fa** and **fb** of a color FERET database is displayed in Figure 5.7. It has an explicit minimum at $R = 3$, which is the best choice according to the criteria stated in the (5.12).

The next factor to be optimized is the regioning process, see Figure 2.2 for details. Regioning helps to save the spatial information about the object, but very small partitioning of the image results in the loss of statistical information about the region. In order to determine the most appropriate compromise of this stage the probability of correct identification at rank 1 is calculated for the different number of regions:

$$\mathbf{m} = \{K^2 | k = 2, 3, \ldots, 10\}.$$

Only the subsets **fa** and **fb** are used for the calculation of $P_I(r = 1)$. For the pattern classification histogram intersection approach is used. The upright bilinear rotation of face images is performed before regioning if the deviation of the eye line from the horizontal exceeds the threshold $\alpha_{eyeline} \geq 3^\circ$.

The highest classification performance is achieved with the number of regions equal to $m = 8^2 = 64$: $P_I(k = 8) = 0.958$.

The parameters $P = 8, R = 3$ and $m = 64$ are used for the calculation of face feature histograms in the next sections. These parameters result in the feature vector length $N = 2^P m = 16384$.

5.5.2 Feature-level weighting for LBP-based face recognition

For training purposes the data set $\boldsymbol{X}^{(1)}$ is formed from the **fa** subset of a FERET database and $\boldsymbol{X}^{(2)}$ from the **fb** subset: $\boldsymbol{X}^{(1),(2)} \in \mathbb{R}^{993 \times 16384}$.

Elements of the vector $\boldsymbol{w} = (w_1, w_2, \ldots, w_{16384})$ are the weights to be adjusted during the optimization process in the feature level. An important aspect of the learning algorithm is an initialization of parameters to be iteratively updated. One of possible approaches is to set all weights to 1: $w_{i,k=0} = 1$, where k is the number of iterations. This initial assumption is accept-

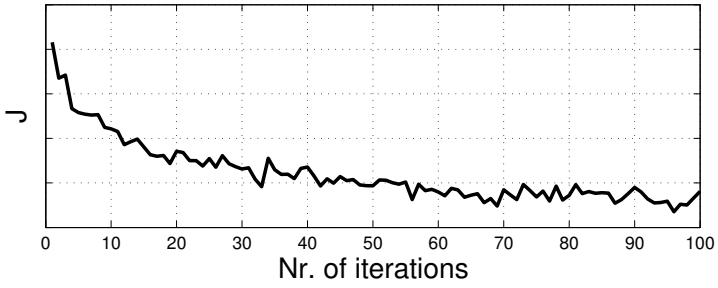


Figure 5.8: An example of the cost function J dependence from the number of iterations with random learning data ($\alpha = 4$, $\eta = 100$, **fa** and **fb** subsets of color FERET)

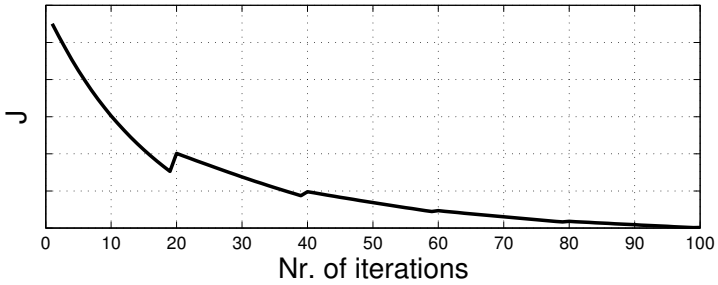


Figure 5.9: An example of the cost function J dependence from the number of iterations with optimal learning data ($\alpha = 4$, $\eta = 10$, $\delta_{iter} = 20$ **fa** and **fb** subsets of color FERET)

able, however correct initialization accelerates the convergence of the algorithm. An initialization technique is developed for better convergence of the algorithm.

$$w_{j,k=0} = 1 - \frac{(m-1)\sum_{i=1}^{M}(x_{i,j}^{(1)} - x_{i,j}^{(2)})^2}{m\sum_{i=1}^{M-1}(x_{i,j}^{(1)} - x_{i+1,j}^{(1)})^2}$$

$$w_{j,k=0} = \begin{cases} w_{j,k=0}, & \text{if } w_{j,k=0} \geq 0, \\ 0, & \text{if } w_{j,k=0} < 0. \end{cases} \tag{5.13}$$

An intuitive explanation of the (5.13) - set high initial weights for the features with good intra class stability and high inter class variance. With feature weighting according to the Equation (5.13) the probability of correct identification at rank $r = 1$ for the subsets **fa** and **fb** of the FERET database is $P_I(r = 1) = 0.962$.

The optimization according to the methodology described in subsection 5.3.1 is performed next, with both *random* and *optimal* learning data selection algorithms. An example of the cost function for 100 iterations (5.5) for *random* learning data selection principle is displayed in the Figure 5.8 and for *optimal* learning data selection principle is displayed in the Figure 5.9. The value of the cost function J is not important, therefore not displayed in the figures.

The cost function with random learning data (Figure 5.8) is decreasing in general, but there are small fluctuations over a small number of iterations, which is the result of random principle.

This aspect makes the learning algorithm slightly unstable and prone to local minimum. Additionally, the random learning data selection results in slightly different solutions every time the algorithm is executed.

The cost function with optimal learning data (Figure 5.9) has small steps after each $\delta_{iter} = 20$ iterations, this fact could be explained with the learning data selection procedure, which is performed after each δ_{iter} iterations. In this case the learning process is more stable, can be replicated, and comes up with a better solution, which is tested empirically.

The value of the probability of correct identification at rank one for the subsets **fa** and **fb** after 500 iterations of the feature-level weighting algorithm with learning parameters $\alpha = 0.5$, $\eta = 10$:

- with initialization of weights according to the Equation (5.13), random learning data selection principle: $P_I = 0.971$
- with initialization of weights according to the Equation (5.13), optimal learning data selection principle: $P_I = 0.976$

5.5.3 Block-level weighting for LBP-based face recognition

Same datasets $\boldsymbol{X}^{(1)}$ and $\boldsymbol{X}^{(2)}$ from the subsets **fa** and **fb** of a color FERET database are used in the learning process. The weights to be adjusted are the elements of a vector $\boldsymbol{w} = (w_1, w_2, \ldots, w_{64})$. For the initial state all weights are assumed to be equal to 1: $w_{i,k=0} = 1$, where k is the number of iterations. No special initialization technique is performed on the block weights, however prior feature-level weighting according to the Equation (5.13) is implemented for maximal performance.

The value of the probability of correct identification at rank one for the subsets **fa** and **fb** after 500 iterations of the block-level weighting algorithm with learning parameters $\alpha = 0.5$, $\eta = 10$:

- with initialization of feature-level weights according to the Equation (5.13), random learning data selection principle: $P_I = 0.980$
- with initialization of feature-level weights according to the Equation (5.13), optimal learning data selection principle: $P_I = 0.989$

5.5.4 Evaluation of the parameters for MSLBP-based face recognition algorithm

Proposed MSLBP-based face recognition algorithm has a lot of variables to be optimized. The MSLBP operator is described with the following parameters: $(P, R = (R_1, R_2, \ldots, R_{n_R}))$, Section 2.1.2. In order to reduce the considered space of the parameters the following variables are

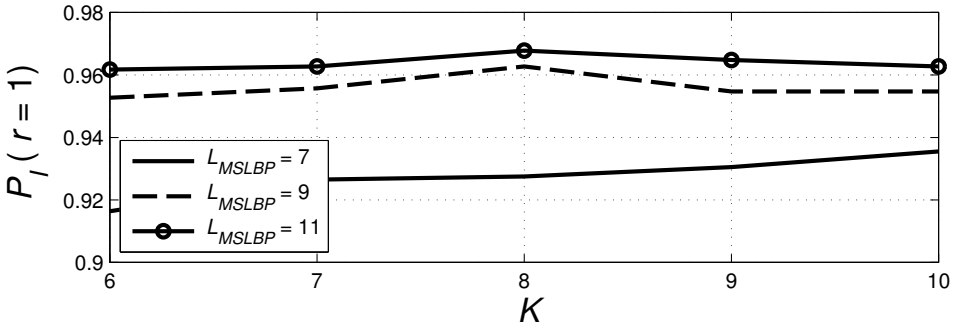


Figure 5.10: Probabilities of correct identification at rank one for different L_{MSLBP} and K values (**fa** and **fb** subsets of a color FERET)

set to constant values: $P = 8$, $n_R = 3$. Only $n_R = 3$ radii are utilized in the process of MSLBP calculation, which is a good choice both for the computational and recognition performances. The set of the radii is selected as follows:

$$R = \left\{ \frac{L_{MSLBP} - 1}{2} - 2, \frac{L_{MSLBP} - 1}{2} - 1, \frac{L_{MSLBP} - 1}{2} \right\},$$

where L_{MSLBP} is the size of the MSLBP neighborhood (Example, Figure 2.4: $L_{MSLBP} = 7$).

Another parameter of the algorithm to be optimized is the regioning factor K. Regioning implements the spatial information about the object into the feature vector, but very detailed partitioning of the object results in the statistical insufficiency of the region data.

The probabilities of correct identification at rank one are calculated for the subsets **fa** and **fb** of a color FERET database with different parameters $\boldsymbol{L_{MSLBP}} = (7, 9, 11)$, $\boldsymbol{k} = (6, 7, 8, 9, 10)$ in order to select the optimal values. For the classification of the patterns the histogram intersection methodology is selected. The upright rotation of the face images is used if the deviation of the eye line from the horizontal exceeds the value $\alpha_{eyeline} \geq 3°$. The size of the face is selected empirically according to the methodology described in Section 5.2.1. The resulting curves are displayed in Figure 5.10. An explicit maximum $P_I(r = 1) = 96.8\%$ is observed for the parameters $L_{MSLBP} = 11$ and $K = 8$, which are utilized in the subsequent optimizations.

Once the L_{MSLBP} and K parameters are selected the **mean filtering** of the size 3×3 is added to the preprocessing of the input image. This stage reduces the negative influence of the texture scale on the recognition result. The value of P_I with $L_{MSLBP} = 11$, $K = 8$ and mean filtering: $P_I = 97.8\%$.

5.5.5 Feature and block level weighting for MSLBP-based face recognition

For learning purposes the data sets $\boldsymbol{X}^{(1)}$ and $\boldsymbol{X}^{(2)}$ are formed from the subsets **fa** and **fb**. The dimensionality of $\boldsymbol{X}^{(1)}$ and $\boldsymbol{X}^{(2)}$ is $\mathbb{R}^{993 \times 16384}$.

The value of $P_I(r = 1)$ for the subsets **fa** and **fb** of a color FERET with empirical feature-

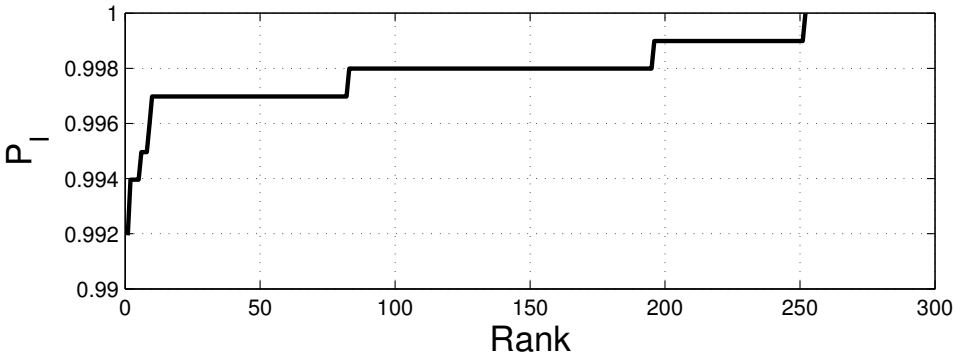


Figure 5.11: Cumulative Match Characteristics after the combination of MSLBP, mean filtering, bar and block level weighting(**fa** and **fb** subsets of a color FERET)

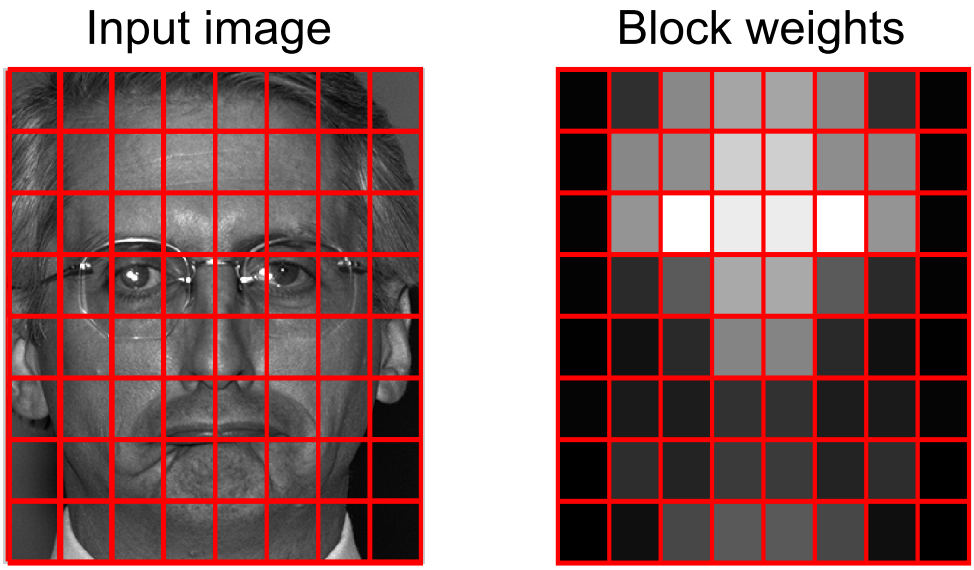


Figure 5.12: Visualization of block weights for each region of the face in MSLBP-based face recognition with block-level weighting principle

level weighting according to the Equation (5.13) is $P_I = 98.1\%$. The weighted histogram intersection is utilized in this case.

The weighting techniques with optimal learning data always provide better results, than the one with random learning data selection principle. Therefore only the results which are based on the optimal learning data selection procedure are introduced in this section.

The value of the probability of correct identification at rank one for the subsets **fa** and **fb** after 500 iterations of the weighting algorithms with learning parameters $\alpha = 0.5$, $\eta = 10$:

- with initialization of feature-level weights according to the Equation (5.13), optimal learning data selection principle, feature-level weighting is performed: $P_I = 0.989$,
- with initialization of feature-level weights according to the Equation (5.13), optimal learning data selection principle, block-level weighting is performed: $P_I = 0.992$. The corresponding CMC is displayed in Figure 5.11.

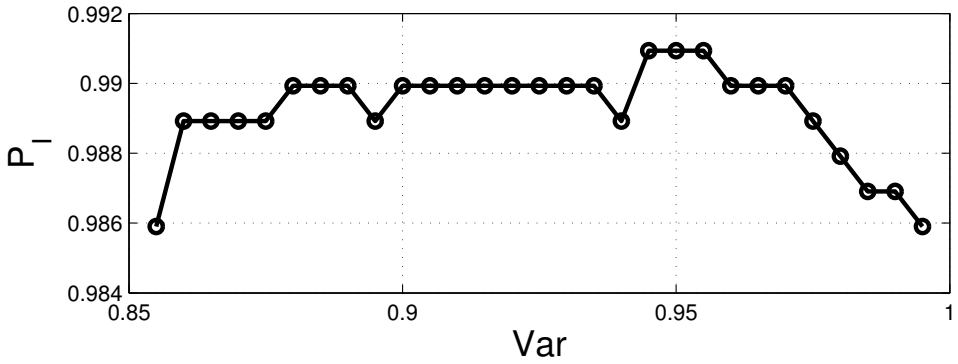


Figure 5.13: The dependence of $P_I(r = 1)$ from the value of the data variance Var retained in the blocks of the MSLBP histogram(**fa** and **fb** subsets of a color FERET)

The visualization of block weights for each region of the face is displayed in the Figure 5.12. The importance of each block is determined by gray-scale intensity: white - important / high value of the weight; black - unimportant / low value of the weight. The range of the weights in Figure 5.12 is $[0, 2.42]$. As seen in the Figure 5.12 the region of eyebrows is the most important in the recognition process. Authors in [102] review 19 important results regarding face recognition by humans. One of the facts was: *"of the different facial features, eyebrows are among the most important for recognition"* [102]. Thus, the result of the proposed weights adjustment algorithm is in some cases similar to the manner human beings recognize the face.

5.5.6 Reduction of the feature vector dimensionality with PCA

The length of the feature vector with previously determined parameters is $N(K = 8, P = 8) = K^2 \cdot 2^P = 16384$. The PCA-based methodology, Section 2.2.2, is utilized next to reduce the dimensionality of the feature space. The variance Var of the data in the blocks of the MSLBP histogram varies in the range $[0.850, 0.995]$ with the step 0.005. The dependence of $P_I(r = 1)$ from the value of the variance retained (**fa** and **fb** subsets, color FERET) is displayed in Figure 5.13. The L_1 distance is utilized to compare the feature vectors after the PCA-based compression.

The highest identification probability $P_I(r = 1) = 99.1\%$ is achieved for the value of the variance $Var = 0.945$. The length of the feature vector in this case is $N_{reduce} = 731$ and the data compression ratio CR:

$$CR = N_{reduce}/N = 731/16384 = 0.045.$$

5.5.7 Summary of the simulation results

The results of the LBP and MSLBP based face recognition are summarized in this subsection. Only the results achieved with optimal learning data are reported here, because this approach always outperforms random principle. The influence of the proposed processing steps onto the

value of $P_I(r = 1)$ for the **fa** and **fb** subsets of a color FERET is summarized here:

- LBP ($K = 8$, $P = 8$, $R = 3$): 95.8%,
- LBP + Empirical Feature-level Weighting (EFW): 96.2%,
- LBP + EFW + Iterative Feature-level Weighting (IFW): 97.6%,
- LBP + EFW + Iterative Block-level Weighting (IBW): 98.9%,
- MSLBP ($L_{MSLBP} = 11$, $K = 8$, $P = 8$, $n_R = 3$): 96.8%,
- MSLBP + Mean Filter (MF): 97.8%,
- MSLBP + MF + EFW: 98.1%,
- MSLBP + MF + EFW + IFW: 98.9%,
- MSLBP + MF + EFW + IBW: **99.2**%.
- MSLBP + MF + EFW + IBW + PCA ($N = 731$): 99.1%.

5.5.8 Comparison of face recognition algorithms

In this section a comparison of the proposed face recognition algorithms with other popular methodologies is introduced. The color FERET database [1] is utilized in the evaluation process. The proposed algorithms are based on the extensions of LBP face recognition approaches [89], however some classical non-LBP principles are also conducted in the comparative study. The identification rates at rank one for the **fa** and **fb** subsets of a color FERET are summarized in the Table 5.1. The identification performances of LBP [9] and MSLBP [23] based face recognition algorithms in original papers have been tested on FERET (not color FERET) database. Results from these papers are also included in the Table 5.1 with remark (**FERET**). However in order to make the comparison correct we have implemented the recognition principles from papers [9] and [23] and then tested them on the color FERET dataset. Next, the proposed preprocessing and weighting-based optimizations are incorporated for further gain in the identification performance, Table 5.1.

5.5.9 Conclusions

This chapter addresses the problem of frontal face recognition in digital images which are captured in semi-controlled lighting conditions. Significant papers in the field of face recognition, including LBP-based techniques, are observed first. A novel extension of LBP-based face recognition approach is introduced next. It is based on the combination of various preprocessing steps, modified Multi-Scale Local Binary Pattern histograms [24] and Weighted Nearest

Table 5.1:
Comparison of face recognition algorithms (**fa** and **fb** subsets of a color FERET)

Method:	Parameters:	$P_I(r=1)(\%)$
MSLBP + MF + EFW + IBW + PCA	$N = 731$ Histogram intersection	99.1
MSLBP + MF + EFW + IBW	$N = 16384$ Histogram intersection	**99.2**
MSLBP, our implementation of [23]	$L_{MSLBP} = 11$, $K = 8$, $P = 8$, $n_R = 3$, histogram intersection	96.8
MSLBP + LDA [23] (tested on **FERET**)		98.9
MSLBP [23] (tested on **FERET**)	Histogram intersection	95.6
LBP + EFW + IBW	Histogram intersection	98.9
LBP, our implementation of [9]	$K = 8, P = 8, R = 3$ Histogram intersection	95.8
LBP + EBW [9] (tested on **FERET**)		97.0
LBP [9] (tested on **FERET**)		93.0
PCA [32]	L1-metric	82.3
ICA [32]	L2-metric	81.5
LDA [32]	L2-metric	82.8

Neighbor Classifier. In general face recognition algorithms can be divided in two stages: feature extraction and classification, see Figure 1.1 for details. The novel contributions are made in both stages.

The introduced combination of Multi-Scale Local Binary Patterns with mean filtering has a better stability of the descriptor for different scales of the object / face in our case. This feature extraction approach is based on the observation, that the texture of the material varies for different magnification factors. LBP operator was originally introduced, as a texture descriptor and the result of the LBP transformation clearly depends on the scale of the object. Introduced features partially resolve this challenging aspect.

Significant attention is also given to the classification stage. Identification approaches are usually based on various Nearest Neighbor Classifiers, which suffer from the lack of statistical information about the problem. The Discriminative Feature Weighting (DFW) algorithm is developed in this research in order to compensate the statistical incompleteness of Nearest Neighbor Classifier by utilizing the information from all classes. The information obtained in the process of weights learning is incorporated in the recognition process by the use of Weighted Nearest Neighbor Classifier (WNNC). The DFW principles are extended in two levels: block-level and feature-level weighting [84] (Nikisins et al.). The advantage of the algorithms is the need of only *two* training examples per class. The algorithm also incorporates special procedure of learning data selection which makes the iterative process stable, predictable and provides better recognition results. Another positive aspect is the presence of mini-batch principle, which makes the proposed training methodology comparatively fast. The speed up of the learning process is important in the cases of massive training data sets and highly dimensional feature

vectors, which are usually true for biometric applications. The introduced approach is general and can be applied in any multi-class classification tasks. Both mathematical and visual interpretations of the proposed weighting algorithm are presented in this chapter.

The comparative study of the introduced face identification methodology has shown an equivalent or even improved performance compared to state-of-the-art recognition techniques, see Table 5.1 for details.

Chapter 6

IMPLEMENTATION OF AUTOMATIC FACE RECOGNITION ALGORITHM IN DIGITAL SIGNAL PROCESSOR

Face recognition systems is a significant part of todays video surveillance and biometrics markets [49]. The transition of biometric algorithms from research laboratories to real world products places new demands on the system: power consumption and cost become critical issues. An embedded implementations become attractive. One of possible approaches is to design the DSP-based system. The goal of this research is to evaluate the performance of face recognition system on the TMS320C6416 platform and to determine the feasibility of DSP-based implementation. The results show that LBP-based automatic face recognition algorithm is potentially a good choice for the design of embedded system.

6.1 Related work

A lot of research is done in the field of implementation of automatic face recognition algorithms. Various approaches are discussed in the literature: DSP-based implementations, FPGA-based systems, mobile phones that can also be considered as heterogeneous embedded systems.

An example of DSP-based face recognition system is introduced in [12]. The proposed system consists of a face detection block, an eye localization block, a face normalization block, and two face classification blocks. For face detection the probabilistic visual learning approach was used [81]. After the location of the face is found, eye localization is performed, where the eyes are detected inside the face at multiple scales and locations. Eye localization is based on the same technique [81]. Next, the face image is rotated to make the eyes horizontal, cropped to exclude the background, and resized to the dimensions of 128x128. These steps are called face normalization. Two classical face recognition algorithms are implemented by the authors: *eigenfaces* [112] and *segmented linear subspaces* [13].

Another example of DSP-based implementation is introduced in [120], where the MagicARM2410 development board is selected for the design. The system is based on Linux operating system. The algorithmic part consists of two blocks: face detection and face recognition. The face detection stage utilize the Haar features [117] and the recognition employs Gabor features and PCA transformation.

The FPGA-based face recognition system is developed in [76]. Authors stated that the system operates in real time. Algorithm consists of face detection, a recognition and a downsampling module. The setup receives video input from a camera, detects the locations of the face(s) using the Viola-Jones algorithm [117], subsequently recognizes each face using the Eigenface algorithm, and outputs the results to a display. An excellent performance of 45 frames per second is achieved on a Virtex-5 FPGA.

Face recognition technologies are gaining increasing attention in the mobile phones due to high popularity of digital cameras in these devices. Authors in [41] considered the task of face detection and authentication in mobile phones and experimentally analyzed a face authentication scheme using Haar-like features [117] with AdaBoost [118] for face and eye detection, and Local Binary Pattern approach for face authentication. Authors stated, that the obtained results are very promising and assess the feasibility of face authentication in mobile phones.

There are also a number of embedded face recognition systems available in the market. One of leading companies in this field is "L-1 Identity Solutions". The company offers for sale 3D face recognition setup and systems that are based on other biometric parameters. However the algorithmic base of the system is the proprietary information.

In the next sessions the proposed DSP-based face recognition system is briefly described both from software and hardware sides.

6.2 Implemented automatic face recognition algorithm

The proposed embedded face recognition system is based on algorithms that are introduced in [85] (Nikisins et al.). An algorithm, that was developed, considers the constraints specific to embedded systems and is based on mathematical operations with a short execution time: addition, multiplication, comparison, bitwise operations and division. The automatic face recognition process is divided into three main stages: face detection, face alignment and face recognition, that are covered in the next subsections.

6.2.1 Face detection stage

The introduced face detection principle is very similar to the one described in section 3.2.1 and belongs to the class of NNC-based detectors. The major differences of this algorithm are the absence of regioning of the sliding window and in the utilized parameters of the LBP operator. The LBP settings are fixed as: $P = 8$ and $R = 1$. The details of the algorithm are discussed

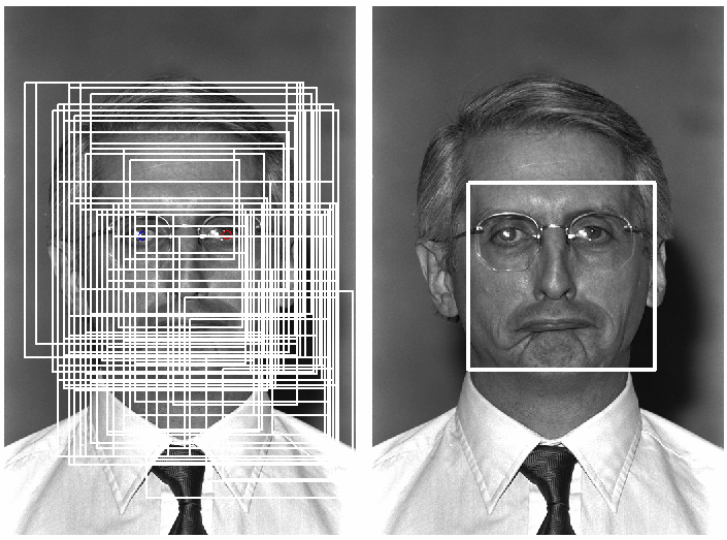


Figure 6.1: An example of multiple detections ($s = 13, k = 5$) and their merging

below.

The LBP transformation of the input image is scanned with windows of different scales s in order to detect the face. At each scanning position (i, j) the distances between the histogram $\boldsymbol{h}_{i,j}$ of the corresponding LBP image region and the histograms of face models $\boldsymbol{f}_k$ are calculated: $d_{i,j}(\boldsymbol{h}_{i,j}, \boldsymbol{f}_k)$.

Some of possible approaches for similarity / dissimilarity $d_{i,j}$ calculation are Histogram intersection, Chi square statistics and Squared Euclidean distance.

To determine the face model histograms $\boldsymbol{f}_k$ we adopted a popular face database, the **color FERET**. Face regions are extracted from all frontal images (**fa** and **fb** sets, number of face images is 2722) of the database. The normalized histograms of the LBP – transformed face images are next calculated for all samples. The method of cluster analysis, namely, **k-means** is next used to partition all face histograms into k clusters with corresponding centroids. We assume these centroids to be normalized face model histograms.

The top left corner coordinates of the face in the input image at a particular scale s for face model k are determined by the location of the maximum $D_{s,k} = \max(\boldsymbol{D})$ in the similarity matrix $\boldsymbol{D}$: $\{x_{s,k}, y_{s,k}\} = find(\boldsymbol{D} = D_{s,k})$.

The proposed face classifier is rather insensitive to detection offsets, which results in multiple detections of the face regions at different locations for different scales and face models, as illustrated in Figure 6.1. The total number of detected regions is equal to $s \cdot k$. The combination of multiple cluster detections is called **merging**. The simplest merging approach selects the detection with the highest similarity score. However, the most precise result in our case was achieved by modified weighted results averaging.

The first step of the proposed merging procedure is the normalization of $D_{s,k}$ scores to $\tilde{D}_{s,k}$ from zero to one. To amplify the influence of significant scores final weights are calculated by raising $\tilde{D}_{s,k}$ to the m power: $W_{s,k} = \tilde{D}^m_{s,k}$. These values are used for weighted averaging of face

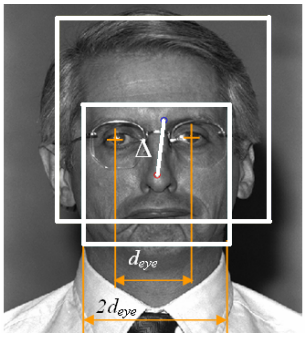


Figure 6.2: Parameters to measure the performance of face detection procedure

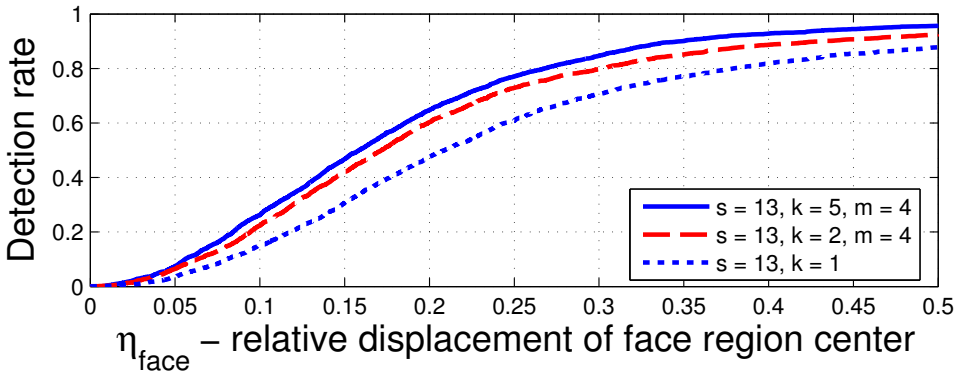


Figure 6.3: ECDF for relative displacement of face regions

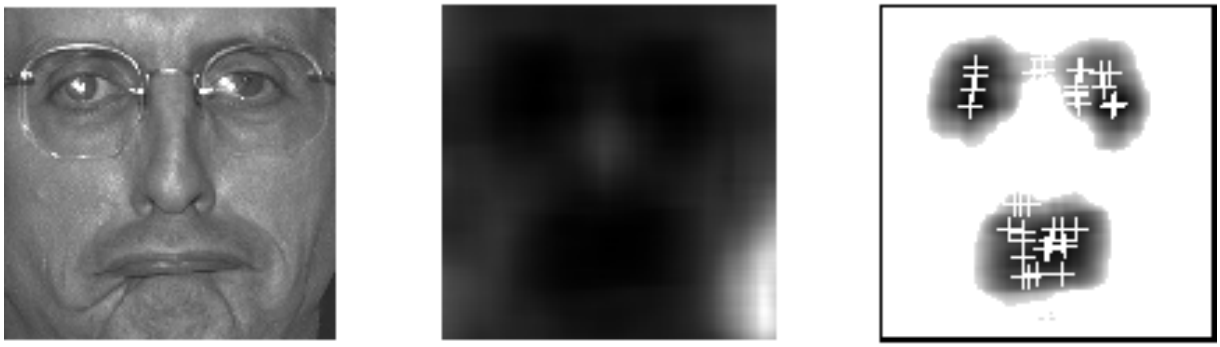
coordinates and sizes.

Evaluation of the proposed face detection algorithm is performed on all frontal face images of a color FERET database. The following criteria are employed in [85] (Nikisins et al.) to measure the effectiveness of the proposed methodology: $\eta_{face} = \Delta/(2 \cdot d_{eye})$, where η_{face} – is the relative displacement of the face region center; Δ stands for the value of an absolute displacement (Figure 6.2); d_{eye} represents inter ocular distance. The detection results are represented in the form of empirical cumulative distribution function (ECDF) Figure 6.3.

The best detection results are achieved when the input image is subsequently scanned with $k = 5$ face models and the results are merged according to $W_{s,k} = \tilde{D}^m_{s,k}$ with $m = 4$. Further augmentation of the amount of clusters increases the computation time, but does not improve the results significantly.

6.2.2 Eye localization - based face alignment

LBP-based face detection is rather insensitive to small detection offsets. That is why the eye localization technique is needed to achieve the expected performance from the system. Once face detection is performed an LBP transformed image of facial region is scanned with a window of size equal to the expected eye dimensions. The parameters of LBP transformation are the same as in the face detection stage: $P = 8$ and $R = 1$. At each position of the window the squared Euclidean distance between the histogram $\boldsymbol{h}_{i,j}$ of the corresponding region and the histogram



(a) (b) (c)

Figure 6.4: (a) - input image; (b) - the result of proposed scanning methodology; (c) - adaptive thresholding of (b) and local minimums '+'

of eye model $\boldsymbol{h}_{eye}$ is calculated: $d_{i,j}(\boldsymbol{h}_{i,j}, \boldsymbol{h}_{eye})$. The denormalization of normalized eye model histogram $\tilde{\boldsymbol{h}}_{eye}$ according to region area is needed before the histograms comparison: $\boldsymbol{h}_{eye} = \tilde{\boldsymbol{h}}_{eye} \cdot s_{eye}^2$. We assume that the eye image has a square shape with dimensions: $s_{eye} = 0.6 \cdot d_{eye}$. The histogram $\tilde{\boldsymbol{h}}_{eye}$ is equal to an average of normalized eye histograms calculated from the color FERET database. Both eyes from **fa** and **fb** sets were taken into consideration. The result of the proposed scanning methodology is displayed in the Figure 6.4, (b), where dark regions correspond to the maximum similarity with the model. However, further processing of the result is still a challenging task.

The adaptive thresholding is applied to $d_{i,j}$ elements of $\boldsymbol{D}_{face}$ matrix:

$$\boldsymbol{D}_{face}\{\boldsymbol{D}_{face} < \overline{\boldsymbol{D}}_{face}/2\} = \max(\boldsymbol{D}_{face}), \tag{6.1}$$

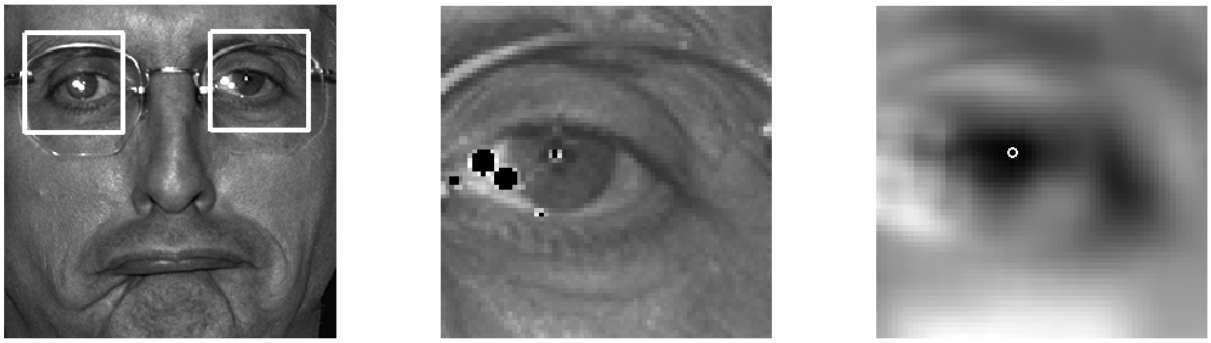
where $\overline{\boldsymbol{D}}_{face}$ is a mean value of $\boldsymbol{D}_{face}$ elements. The result of adaptive thresholding is displayed in the Figure 6.4, (c). The next step of the algorithm is to determine the local minimums $\{\boldsymbol{X}_{min}, \boldsymbol{Y}_{min}\}$ of $\boldsymbol{D}_{face}$ matrix. All these minimums $\{\boldsymbol{X}_{min}, \boldsymbol{Y}_{min}\}$ are considered as potential eye coordinates, but we need to select only two points that are most likely to be referring to eye positions.

A simplified way of eye coordinates selection is to find two points with corresponding minimal values of $\boldsymbol{D}_{face}$ matrix. However, we have developed a more stable algorithm to resolve this issue. Our approach takes into consideration both empirical and quantitative information: interocular distance, the angle between the eyes and the corresponding d value. The following matrices describe the stated parameters:

$$T_{i,j} = [|(y_i - y_j)/(x_i - x_j)| < t], \tag{6.2}$$

$$Dx_{i,j} = |x_i - x_j|, \tag{6.3}$$

$$V_{i,j} = \boldsymbol{D}_{face}(x_i, y_i) + \boldsymbol{D}_{face}(x_j, y_j), \tag{6.4}$$



(a) (b) (c)

Figure 6.5: An example of eye region detection result (a); the compensation of bright regions for the left eye (b); (c) – the result of Gaussian lowpass filtering of the input image (b)

$$i = 0, \ldots, m-1, j = 0, \ldots, m-1,$$

where [] is the notation that denotes Iverson bracket and the condition in square brackets is satisfied if the slope of a line that is bounded by i and j points is less than t. The elements of $\boldsymbol{Dx}$ represents the horizontal component of interocular distance and $\boldsymbol{V}_{i,j}$ is the sum of Squared Euclidean distances for i and j ponts; m is the total number of local minimums. The coordinates of maximum in the reference matrix $\boldsymbol{Ref}$ indicate the serial numbers of points $\{p_1, p_2\}$ that are most likely referring to eye positions:

$$\boldsymbol{Ref}_{i,j} = \boldsymbol{T}_{i,j} \cdot \boldsymbol{Dx}_{i,j} / \boldsymbol{V}_{i,j} \tag{6.5}$$

$$\{p_1, p_2\} = find(\boldsymbol{Ref} = \max(\boldsymbol{Ref})) \tag{6.6}$$

Detected eye regions Figure 6.5, (a) are still not precise enough for localization purpose. A few more steps are needed to achieve the desired precision. First, bright spots in the eye image are set to black Figure 6.5, (b), which is needed to reduce the effect of light - striking. The resulting image is blurred with a Gaussian low-pass filter Figure 6.5, (c) and the intensity minimum is detected next (white "**o**" dot Figure 6.5, (c)). Coordinates of the minimum define the position of the eye pupil center, which is the final step of the localization procedure.

Evaluation of the proposed eye detection algorithm is performed on all frontal face images of a color FERET database. The following criteria are selected for performance evaluation of eye localization approach: $\eta_{eye} = \Delta_{eye}/d_{eye}$, where Δ_{eye} is the absolute displacement of eye pupil center. The localization statistics are displayed in the form of ECDF for both eyes Figure 6.6.

For proper operation of the face recognition algorithm the detection rate for both eyes should not exceed the fixed value of relative eye displacement. The acceptable value for the displacement is: $\eta_{eye} \leq 0.1$. For assessment the correct localization rates P_L are observed for different η_{eye} values: $\eta_{eye} = (0.05, 0.1, 0.15)$, then corresponding $P_L = (40.9\%, 74.3\%, 77.8\%)$.

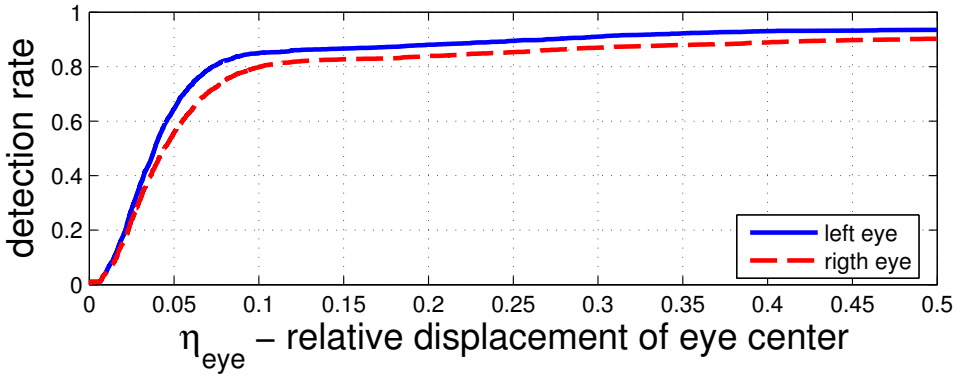


Figure 6.6: ECDF for relative displacement of eye centers

Figure 6.7: An example of a facial image divided into 10 windows: 4 regions per eye, 1 nose and 1 mouth region

6.2.3 Face recognition stage

The spatially enhanced LBP histograms effectively represents both local and regional description of the face. Extensive experiments with a region selection procedure clearly show the importance of this stage in face recognition applications.

Face division into $m = 10$ regions is selected in this approach, as illustrated in Figure 6.7. This selection is made according to the importance of facial regions in the recognition process, which is determined in Section 5.5.5, Figure 5.12. Region dimensions and positions of the top-left corners are determined by the coordinates of the right and left eyes.

To perform the comparison of faces with different scales and parameters the histogram of each region must be normalized according to region parameters (w_j – width and h_j – height of region j) to get a coherent description:

$$\tilde{\boldsymbol{h}}_{i,j} = \boldsymbol{h}_{i,j}/(w_j h_j), j = 0, \ldots, m-1. \tag{6.7}$$

For the pattern classification a nearest neighbor classifier is used. Among the most popular approaches for similarity / dissimilarity measures are Histogram intersection, Chi square statistics and Squared Euclidean distance [9].

A closed-set model is selected for algorithm evaluation. It is assumed, that in the closed–set

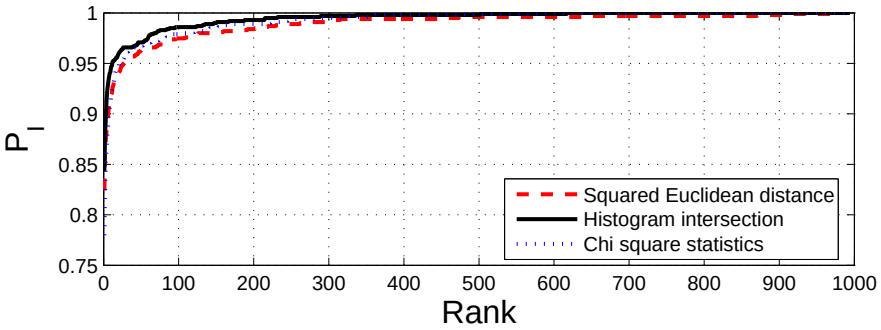


Figure 6.8: Cumulative Match Characteristic for LBP face recognition algorithm ($P = 8, R = 1, m = 10$); **Color FERET** database

identification every probe image has a corresponding match in the database. The closed–set model allows the determining of how good an algorithm is at identifying a probe image. The subset **fa** of a color FERET is used as a gallery set (contains frontal images of 993 persons) and **fb** is used as a probe set (993 images; the persons were asked for a different facial expressions than in **fa** set).

The performance statistics for this model are reported as Cumulative Match Characteristics in the Figure 6.8. The horizontal axis of the figure is rank and the vertical axis is the probability of identification (P_I). The probability of correct identification at rank N means that the correct match is somewhere in the top N similarity scores with corresponding probability.

6.2.4 Experiments with EDI face database

The FERET database is very popular among researchers in the field of face recognition. However for real life experiments it is important to have a database of actual users that is collected with experimental setup. For this purpose a database of the staff of Institute of Electronics and Computer Science was collected. The abbreviated name of this facial dataset is **EDI**. The database is utilized in the experiments with DSP-based face recognition system, it was collected for local use only and is not publicly available.

The database contains 168 frontal face images of 51 individuals. The number of images per person varies from 2 till 5. The individuals were asked for a different facial expressions in each image. The lighting conditions were semi-controlled.

Next, the above proposed algorithms (subsections 6.2.1 - 6.2.3) are tested on EDI face dataset. The following results are obtained for each stage of the automatic face recognition process:

- Face detection stage: 98.2 % of the faces are detected correctly. Here the term "correct detection" means that the eye regions are located inside of the face bounding box. Thus, the next stage of the algorithm can possibly be completed to a valid solution.

- Eye localization stage: $P_L(\eta_{face} \leq 0.1) = 86.3\%$. The absolute value of errors is 23 of

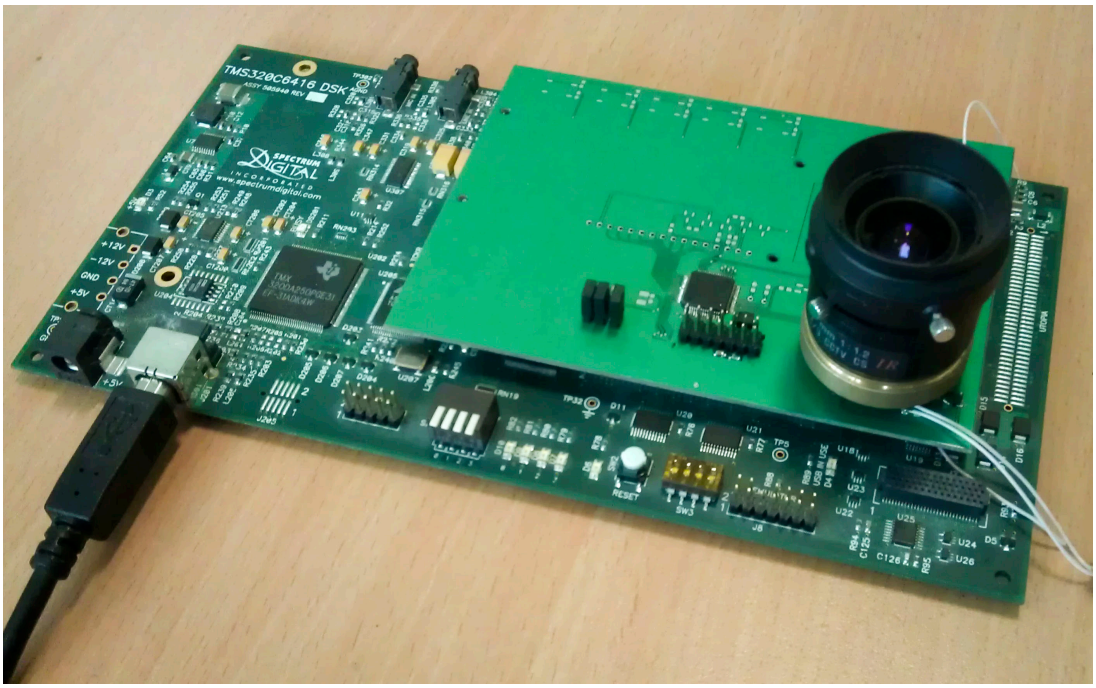


Figure 6.9: TMS320C6416 DSK development board

which 15 are observed for images with glasses.

- Face recognition stage: the value of EER is 6.5 %. The number of images per individual is variable therefore the Equal Error Rate (EER) [5] is selected as a performance measure.

The precision of algorithms that are presented in Chapters 3 - 5 of this research is higher. However, the implementation of above algorithms can serve as a good example of feasibility of face recognition system in DSP, and is a good starting point for future embedded solutions.

6.3 DSP-based automatic face recognition system

A fully automatic face recognition algorithm is implemented on TMS320C6416 DSK development board (Figure 6.9) that contains a TMS320C6416 fixed-point digital signal processor operating at 600 MHz and an external non-volatile Flash memory of size 512 Kbytes. The algorithmic base of the system is exactly the same as the one described in section 6.2, but a few simplifications are introduced in order to speedup the system:

- The expected size of the face is 150×150 pixels. Thus, the scanning of the input image is performed only once with a fixed size of the sliding window. This assumption is reasonable if the face is located at a fixed distance from the camera. In our setup the acceptable distances are in the range of 70 to 90 centimeters.

- The number of available face models is *one*, so only one similarity matrix is obtained after the scanning is completed.

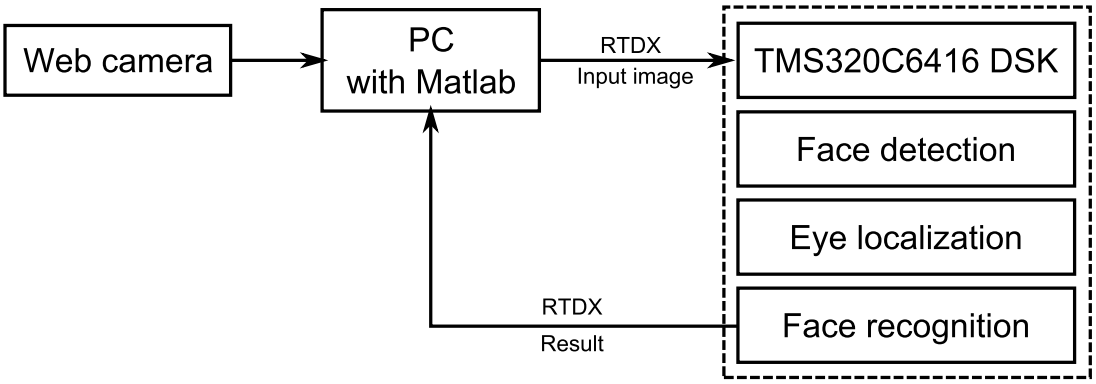


Figure 6.10: A block-scheme of the DSP based automatic face recognition setup

- The expected size of the eye image is 40×40 pixels and the number of available eye models is also *one*.

Introduced implementation follows the the block diagram shown in the Figure 6.10. The experimental setup consists of three main blocks: *web camera*, *PC with installed Matlab software* and *TMS320C6416 DSK*. The functionality of these blocks is as follows:

Web camera - is a capturing device. A still input image of the face is obtained from this device by a computer. The Prestigion 2.0 Mega-pixels camera PWC2 is selected for experiments.

PC with Matlab software. One of the primary functions of the PC is a user-computer interface. It displays video from the web camera on the screen as a reference information for the user. When the face is in the desired position (70 to 90 centimeters from the camera, frontal view) a frame is captured by a single button press on the keyboard.

The image is then preprocessed to a gray-scale format of the resolution 460×614 pixels and is transfered to the DSP via RTDX (real-time data exchange) interface. The TI C6000 DSP toolbox is needed for Matlab in order to use this interface.

Once the processing of the input image is completed on the DSP the coordinates of eye pupils and the recognition result are transfered to the PC via RTDX. The information about eye pupils is utilized as a reference data that guarantees the correct operation of face detection and eye localization blocks. The recognition result is represented in the form of image number in the EDI database that is most similar to the input face image. An image of the identified individual from EDI dataset is then displayed on the screen.

TMS320C6416 DSK is the main processing unit in the system. All stages of automatic face recognition process are implemented in the DSP: face detection, eye detection and face recognition. Since TMS320C6416 signal-processor is a fixed-point device the C code is optimized to operate with integer data. This optimization has both positive and negative impact on the performance of the system. Obviously, the calculations are executed faster in the DSP, however the gain in the computation time is not known due to the absence of floating-point C implementation of the algorithms. While the precision of face and eye detection stages is not affected much by fixed-point simplifications, the EER of face recognition stage for EDI database increased form

Table 6.1:
Performance profile of DSP based automatic face recognition algorithm

	CPU cycles ($\times 10^6$)	Computation time (CPU at 600 MHz)
LBP transformation	658	1.10 seconds
Face detection	314	0.52 seconds
Eye localization	1137	1.90 seconds
Face recognition	167	0.28 seconds

6.5% till 8.2%.

Next, the structure of the code is briefly discussed. The database is preloaded in the external non-volatile Flash memory of the board. The size of the Flash is 512 Kbytes, which is sufficient to store the dataset of 100 LBP histograms of the length $N = 2560$ with 2 byte encoding for each bin. Therefore only 100 patterns from EDI database are stored in the Flash. After the initialization of the DSP board is completed the database is loaded from Flash memory to the external SDRAM. 16 Mbytes of SDRAM are available on the board. SDRAM is connected to TMS320C6416 via external memory interface namely EMIFA. After the database related processes are completed the input image is transfered to the SDRAM from PC via RTDX. Size of the input image is more than 280 Kbytes. The DSP has two-block memory architecture: L1 and L2. The largest one is L2 which has 1024 Kbytes, but it is still not enough to store all the data in the internal memory, therefore the input image is transfered to external SDRAM. The histograms of face and eye models and Gaussian mask of the size 15×15 are copied next to internal DSP memory from PC. These vectors are often addressed during the execution of the program, thus it is a good practice to store them internally. These are the last initialization steps and the main body of the code (detection and recognition stages) is executed next.

First, the LBP transformation of the input image is performed. The size of the image is equal to 460×614 pixels and the parameters of LBP operator are $(P = 8, R = 1)$. Next, face detection, eye detection and face recognition functions are consequentially executed. The performance profile of automatic face recognition algorithm is described in the table 6.1. An image of the identified individual from EDI dataset is also displayed on the screen.

Visualization of the recognition result in the Matlab environment is displayed in the Figure 6.11. Red circles are displayed over eye pupils in the input as a reference data that guarantees correct operation of face detection and eye localization blocks. An image of the identified individual from EDI dataset is also displayed on the screen. This information is a good feedback for the user which is intended to detect possible issues in the system.

6.3.1 Conclusions

This chapter covers the process of DSP-based automatic face recognition system design. The relevance of embedded biometric systems in today's market and similar embedded solutions are observed first. Next, the implementation of fully automatic face recognition algorithm on

Figure 6.11: Visualization of the recognition result in the Matlab environment. Automatic face recognition algorithm is executed in the DSP

TMS320C6416 DSK development board is described. The board contains a TMS320C6416 fixed-point digital signal processor operating at 600 MHz and an external non-volatile Flash memory of size 512 Kbytes. The algorithmic part of the system is proposed first, since it has some empirical and mathematical simplifications in contrast to the principles in Chapters 3 - 5. Even with simplified algorithmic basis of the system the simulation results has shown a reasonable performance on real-life data. Thus, the system is suitable for stand-alone operation or in multi-modal biometric solutions. The resulting timing analysis confirms the feasibility of the proposed LBP and NNC based automatic face recognition algorithm in embedded systems. The recognition process requires less than 2.3×10^9 CPU cycles to process a single 0.3 Mpixel image. Furthermore, significant improvements in computational time are possible with methods which are described in Section 3.6.6. Perspective on the theme of this chapter a spacial attention can be given to *software and hardware level optimizations*. In particular the optimizations of the source code and of the compilation process can affect the execution time significantly. The proposed detection methodologies are histogram-based, thus the effective parallelization of the process is possible. Therefore, modern FPGA and multi-core DSP are potentially a better choice for embedded system design.

Chapter 7

CONCLUSION

In this thesis a fully automatic face recognition system is presented: both algorithmic basis and hardware implementation are covered. The automatic face recognition algorithm is composed of three stages: face detection, eye localization and face recognition. All stages are conceptually merged by utilized feature modules, which are based on Local Binary Patterns. First two stages are needed for precise alignment of facial region in the input image which is a critical aspect for correct operation of the final setup. The enclosing step, namely face recognition, compares the extracted face image with available database in order to identify the person or to reject the identification attempt. Since all blocks are executed sequentially the resulting performance is highly dependent on the accuracy of each module. Therefore significant attention is given to various optimizations of parameters of the algorithm in order to rise the accuracy of each step. A unified evaluation methodologies, which are accepted by many researchers in the corresponding fields, are utilized during optimizations, performance analysis and comparative studies. This approach makes the research results accessible to the research community. Besides the contributions in the feature extraction processes, the thesis also introduce a novel classification approaches, which are based on the iterative adjustment of the weights in the WNNC classifier. Both face and eye detection stages are two-class classification problems, while the recognition step is a multi-class classification task. Face and eye detectors are based on a well-known ANN and SVM classifiers, however the adjustment of the parameters of the classifier is a challenging task which is described in details in the corresponding sections. A novel classification technique is developed for the recognition stage which is a multi-class problem. The methodology is based on the iterative adjustment of feature/block weights in the Weighted Nearest Neighbor Classifier so as to introduce the measure of importance of each parameter in the recognition process. The proposed LBP-based face recognition algorithm is also implemented in the DSP-based platform in order to demonstrate the feasibility of the introduced automatic face recognition algorithms in the embedded solutions.

7.1 Face detection

The main research in the field of face detection has focused on the combination of Local Binary Patterns with Artificial Neural Network or Support Vector Machines so as to design a flexible and robust object detectors. As the result a novel cluster of LBP-based face detection algorithms is proposed. The advantage of this methodology is the flexibility of the algorithm, which allows to adjust the trade-off between the dimensionality of the feature space and the complexity of the classifier. Another positive moment is the absence of the down-sampling stage, which is often incorporated in the detection algorithms in order to localize object of various scales. The proposed methods are tested in terms of localization performance. The precision which is comparable to state-of-the-art algorithms is obtained in low-dimensional feature space (several hundreds of features) and with simple classifier (Artificial Neural Network with 10 Neurons in the hidden layer or Support Vector Machine (SVM) with 100-200 Support Vectors). While the localization precision is high, the computational time is still a challenging issue for the proposed approach. The detection time is measured in the range of seconds or even tens of seconds, depending on the parameters of the system. This fact can be explained by the absence of special techniques for the reduction of the number of scanning positions. Some of possible improvements of scanning process are described below:

- *Adjustment of scanning parameters*. Such factors as the step of the sliding window or the number of expected scales of the detectable object significantly impacts the computational time, however it is also affects the localization accuracy. The empirical knowledge about the task and physical setup of the system can help to find the desired trade-off.

- *Reduction of the search region*. Fast preprocessing methods can effectively discard subwindows before computationally expensive processing by the classifier. These methods are usually based on color, variance or texture analysis.

- *Adaptive adjustment of scanning parameters*. The adjustment of scanning parameters is possible based on the confidence scores in the previous scanning positions.

- *Task specific optimizations*. The biometric systems are often designed to operate in localization mode, thus only one face is present in the input image. Additionally, the user is usually interested to cooperate with the system. Based on this information the optimal selection of the starting scanning position and of the scale of the sliding window is possible.

- *Software and hardware level optimizations*. The source code and compilation optimizations also affect the execution time. The proposed detection methodologies are histogram-based, thus the effective parallelization of the process is possible. However parallelization requires special hardware solutions, such as multi-core DSP, FPGA or graphical cards.

Above methodologies can significantly speed up the introduced detectors, however these aspects are out of the scope of this research and can be considered as a future work in this field.

7.2 Eye localization

A novel LBP-based eye localization approach is developed in this thesis. The introduced localization algorithms consists of two main stages: localization of eye regions and detection of eye pupils. The first stage is an extension of proposed face detection methods to another cluster of detectable objects. The second stage is needed for further gain in the localization precision. The experimental section of the corresponding chapter clearly show that the proposed method outperforms observed state-of-the-art eye localization approaches. High localization accuracy is obtained in low-dimensional feature space (144 LBP features) and with simple classifier (Artificial Neural Network with 10 Neurons in the hidden layer or Support Vector Machine (SVM) with 100-200 Support Vectors). Similar to face detection, the scope of the experiments in this research is limited to the task of eye localization in frontal face images taken under semi-controlled lighting conditions. Partial occlusions are presented in the test images in the form of glasses, which is the main reason of erroneous detections.

The computational time for eye localization is measured in the range of seconds (for images in FERET dataset) and depends on the parameters of the system. This fact can be explained by the absence of special techniques for the reduction of the number of scanning positions. Some of possible improvements of scanning process are described above in Section 7.1.

Further improvements in localization accuracy are also possible and potential supplements to the algorithm are briefly summarized here:

- *Reduction of the search space*. The eye localization stage is the second step in automatic face recognition process and the location of the face is already estimated by the face detector. This information can help to restrict the area of possible eye locations in the input face image.

- *Utilize information about other facial features*. The analyzed input image is known to be face, thus the knowledge of relative positions of the facial features (eyes, nose, moth and others) can help to improve accuracy.

- *Shape information*. The second stage of eye localization algorithm, namely detection of eye pupils, is based on the intensity information only. The shape of eye pupils is explicitly circular which can also benefit as an additional data in more complicated localization approaches.

Above mentioned methodologies are based on the knowledge about the analyzed input image, which in our case is a face. The first principle can both improve the accuracy and speed up

the detector. However these aspects are out of the scope of this research and can be considered as a future work in this field.

7.3 Face recognition

One of significant contributions of this thesis is achieved in the face recognition stage. A novel extension of LBP-based face recognition approach is proposed. It is based on the combination of various preprocessing steps, modified Multi-Scale Local Binary Pattern histograms [24] and Weighted Nearest Neighbor Classifier. In general face recognition algorithms can be divided in two stages: feature extraction and classification, see Figure 1.1 for details. The novel contributions are made in both stages.

The introduced combination of Multi-Scale Local Binary Patterns with mean filtering has a better stability of the descriptor for different scales of the object / face in our case. This feature extraction approach is based on the observation, that the texture of the material varies for different magnification factors. LBP operator was originally introduced, as a texture descriptor and the result of the LBP transformation clearly depends on the scale of the object. Introduced features partially resolve this challenging aspect.

Significant attention is also given to the classification stage. Identification approaches are usually based on various Nearest Neighbor Classifiers, which suffer from the lack of statistical information about the problem. The Discriminative Feature Weighting (DFW) algorithm is developed in this research in order to compensate the statistical incompleteness of Nearest Neighbor Classifier by utilizing the information from all classes. The information obtained in the process of weights learning is incorporated in the recognition process by the use of Weighted Nearest Neighbor Classifier (WNNC). The DFW principles are extended in two levels: block-level and feature-level weighting [84] (Nikisins et al.). The advantage of the algorithms is the need of only *two* training examples per class. The algorithm also incorporates special procedure of learning data selection which makes the iterative process stable, predictable and provides better recognition results. Another positive aspect is the presence of mini-batch principle, which makes the proposed training methodology comparatively fast. The speed up of the learning process is important in the cases of massive training data sets and highly dimensional feature vectors, which are usually true for biometric applications. The introduced approach is *general* and can be applied in any multi-class classification tasks. Both mathematical and visual interpretations of the proposed weighting algorithm are presented in the corresponding chapter.

The comparative study of the introduced face identification methodology has shown an equivalent or even improved performance compared to state-of-the-art recognition techniques. However only the task of frontal face recognition in images captured under semi-controlled lighting conditions is observed in this research. The extensions of the proposed methods to uncontrolled lighting environment and off-plane rotations of the faces is a challenging task which can be considered as a valuable future work.

7.4 DSP-based implementation

The DSP-based demonstrator of automatic face recognition system is developed in this thesis. The system is based on TMS320C6416 DSK development board, which contains a TMS320C6416 fixed-point digital signal processor operating at 600 MHz and an external non-volatile Flash memory of size 512 Kbytes. The algorithmic part of the system is also covered in the corresponding chapter, since it has some empirical and mathematical simplifications in contrast to the principles in Chapters 3 - 5. Even with simplified algorithmic basis of the system the simulation results has shown a reasonable performance on real-life data. Thus, the system is suitable for stand-alone operation or in multi-modal biometric solutions. The resulting timing analysis confirms the feasibility of the proposed LBP and NNC based automatic face recognition algorithm in embedded systems. The recognition process requires less than 2.3×10^9 CPU cycles to process a single 0.3 Mpixel image. Furthermore, significant improvements in computational time are possible by incorporating methods from Section 3.6.6. In future research a spacial attention can be given to *software and hardware level optimizations*. The proposed detection methodologies are histogram-based, thus the effective parallelization of the process is possible. Therefore, modern FPGA and multi-core DSP are potentially a better choice for embedded system design.

Bibliography

[1] *The Color FERET Database.* http://www.nist.gov/itl/iad/ig/colorferet.cfm/.

[2] *The ORL Database of Faces.* http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html.

[3] *Tutorial: Unsupervised Feature Learning and Deep Learning. Stanford University.* http://deeplearning.stanford.edu/wiki/index.php/.

[4] *2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001), with CD-ROM, 8-14 December 2001, Kauai, HI, USA*. IEEE Computer Society, 2001.

[5] Biometrics testing and statistics. Technical report, National Science and Technology Council. Subcommittee on Biometrics, 2006.

[6] *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008), 24-26 June 2008, Anchorage, Alaska, USA*. IEEE Computer Society, 2008.

[7] *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*. IEEE, 2009.

[8] Yael Adini, Yael Moses, and Shimon Ullman. Face recognition: the problem of compensating for changes in illumination direction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:721--732, 1997.

[9] T. Ahonen, A. Hadid, and M. Pietikainen. Face recognition with local binary patterns. *Computer Vision, ECCV 2004 Proceedings, Lecture Notes in Computer Science 3021, Springer*, pages 469--481, 2004.

[10] M. Aizerman, E. Braverman, and L. Rozonoer. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control*, 25:821--837, 1964.

[11] E. Alpaydın. *Introduction to Machine Learning. Second Edition*. The MIT Press, 2010.

[12] A. U. Batur and B. E. Flinchbaugh. Performance analysis of face recognition algorithms on tms320c64x. Technical report, Texas Instrument DSP Solutions R & D Center.

[13] Aziz Umit Batur and Monson H. Hayes III. Linear subspaces for illumination robust face recognition. In *2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001), with CD-ROM, 8-14 December 2001, Kauai, HI, USA*, pages 296--301. IEEE Computer Society, 2001.

[14] D. P. Berrar, W. Dubitzky, and M. Granzow. *A Practical Approach to Microarray Data Analysis*. Kluwer Academic Publishers, 2003.

[15] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *SIGGRAPH*, pages 187--194, 1999.

[16] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces, 1999.

[17] David S. Bolme, Bruce A. Draper, and J. Ross Beveridge. Average of synthetic exact filters. In *CVPR* [7], pages 2105--2112.

[18] B. E. Boser, I. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers. In D. Haussler, editor, *COLT*, pages 144--152. ACM, 1992.

[19] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[20] D. Brain and G. Webb. On the effect of data set size on bias and variance in classification learning. *in D. Richards, G. Beydoum, A. Hoffman, P. Compton (eds), 4th Australian Knowledge Acquisition Workshop*, pages 117--128, 1999.

[21] C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121--167, 1998.

[22] P. Campadelli, R. Lanzarotti, and G. Lipori. Eye localization: a survey. In *The Fundamentals of Verbal and Non-verbal Communication and the Biometrical Issue*. NATO Science Series, 2007.

[23] C. H. Chan. *Multi-scale Local Binary Pattern Histogram for Face Recognition*. PhD thesis, Centre for Vision, Speech and Signal Processing School of Electronics and Physical Sciences University of Surrey, 2008.

[24] Chi-Ho Chan, Josef Kittler, and Kieron Messer. Multi-scale local binary pattern histograms for face recognition. In Lee and Li [64], pages 809--818.

[25] C. C. Chang and C. J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1--27:27, 2011.

[26] Chih-Chung Chang and Chih-Jen Lin. Libsvm: A library for support vector machines. *ACM TIST*, 2(3):27, 2011.

[27] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(6):681--685, 2001.

[28] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models - their training and application. *Comput. Vis. Image Underst.*, 61(1):38--59, January 1995.

[29] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273--297, 1995.

[30] Ingemar Cox, Joumana Ghosn, and Peter N. Yianilos. Feature-based face recognition using mixture-distance. pages 209--216. IEEE Press, 1996.

[31] Ángel de la Torre, Antonio M. Peinado, Antonio J. Rubio, José C. Segura, and M. Carmen Benítez. Discriminative feature weighting for hmm-based continuous speech recognizers. *Speech Communication*, 38(3-4):267--286, 2002.

[32] Kresimir Delac, Mislav Grgic, and Sonja Grgic. Independent comparative study of pca, ica, and lda on the feret data set, 2004.

[33] Oscar Déniz, Modesto Castrillón Santana, and Mario Hernández. Face recognition using independent component analysis and support vector machines. *Pattern Recognition Letters*, 24(13):2153--2157, 2003.

[34] Shifei Ding, Hong Zhu, Weikuan Jia, and Chunyang Su. A survey on feature extraction for pattern recognition. *Artif. Intell. Rev.*, 37(3):169--180, March 2012.

[35] Tiziana D'Orazio, Marco Leo, Grazia Cicirelli, and Arcangelo Distante. An algorithm for real time eye detection in face images. In *ICPR (3)*, pages 278--281, 2004.

[36] C. Elkan. Nearest neighbor classification. Technical report, University of California, San Diego, January 2011.

[37] Mark Everingham and Andrew Zisserman. Regression and classification approaches to eye localization in face images. In *FG*, pages 441--448. IEEE Computer Society, 2006.

[38] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Pictorial structures for object recognition. *International Journal of Computer Vision*, 61(1):55--79, 2005.

[39] M. R. Gupta and W. Mortensen. Weighted nearest-neighbor learning and first-order error. *Invited Paper to the Proc. Intl. Conf. on Frontiers of Interface Between Statistics and Science*, 2009.

[40] A. Hadid, M. Pietikäinen, and T. Ahonen. A discriminative feature space for detecting and recognizing faces. In *CVPR (2)*, pages 797--804, 2004.

[41] Abdenour Hadid, Jarkko Y. Heikkilä, Olli Silvén, and Matti Pietikäinen. Face and eye detection for person authentication in mobile phones. In *ICDSC*, pages 101--108. IEEE, 2007.

[42] Ferdinand Hahmann, Heike Ruppertshofen, Gordon Böer, and Hauke Schramm. Model interpolation for eye localization using the discriminative generalized hough transform. In Arslan Brömme and Christoph Busch, editors, *BIOSIG*, pages 1--12. IEEE, 2012.

[43] Antonio Haro, Myron Flickner, and Irfan A. Essa. Detecting and tracking eyes by using their physiological properties, dynamics, and appearance. In *CVPR*, pages 1163--1168. IEEE Computer Society, 2000.

[44] Jeff Hawkins and Sandra Blakeslee. *On Intelligence*. St. Martin's Griffin; First Edition, 2005.

[45] B. Heisele, P. Ho, J. Wu, and T. Poggio. Face recognition: component-based versus global approaches. *Computer Vision and Image Understanding*, 91(1-2):6--21, 2003.

[46] B. Heisele, T. Serre, and T. Poggio. A component-based framework for face detection and identification. *International Journal of Computer Vision*, 74(2):167--181, 2007.

[47] Bernd Heisele, Thomas Serre, Massimiliano Pontil, and Tomaso Poggio. Component-based face detection. In *CVPR (1)* [4], pages 657--662.

[48] Jennifer Huang, Bernd Heisele, and Volker Blanz. Component-based face recognition with 3d morphable models. In Josef Kittler and Mark S. Nixon, editors, *AVBPA*, volume 2688 of *Lecture Notes in Computer Science*, pages 27--34. Springer, 2003.

[49] Rabia Jafri and Hamid R. Arabnia. A survey of face recognition techniques. *JIPS*, 5(2):41--68, 2009.

[50] A.K. Jain, J. Mao, and K.M. Mohiuddin. Artificial neural networks: A tutorial. *Computer*, pages 31--44, 1996.

[51] Anil K. Jain. Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, 31(8):651--666, 2010.

[52] Oliver Jesorsky, Klaus J. Kirchberg, and Robert Frischholz. Robust face detection using the hausdorff distance. In Josef Bigün and Fabrizio Smeraldi, editors, *AVBPA*, volume 2091 of *Lecture Notes in Computer Science*, pages 90--95. Springer, 2001.

[53] Qiang Ji, Harry Wechsler, Andrew T. Duchowski, and Myron Flickner. Special issue: eye detection and tracking. *Computer Vision and Image Understanding*, 98(1):1--3, 2005.

[54] Takeo Kanade. Picture processing system by computer complex and recognition of human faces. In *doctoral dissertation, Kyoto University*. November 1973.

[55] A. M. Kibriya and E. Frank. An empirical comparison of exact nearest neighbour algorithms. *Proceedings of the 11th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD 2007)*, 4702 of Lecture Notes in Computer Science:140--151, September 2007.

[56] M. Kirby and L. Sirovich. Application of the karhunen-loeve procedure for the characterization of human faces. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12(1):103--108, January 1990.

[57] T. Kohonen, M. R. Schroeder, and T. S. Huang, editors. *Self-Organizing Maps*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 3rd edition, 2001.

[58] S. Kulkarni and G. Harman. *Statistical Learning Theory: A Tutorial*. Wiley Interdisciplinary Reviews: Computational Statistics. DOI: 10.1002/wics.179., 2011.

[59] B. V. K. Vijaya Kumar, Abhijit Mahalanobis, and Alex Takessian. Optimal tradeoff circular harmonic function correlation filter methods providing controlled in-plane rotation response. *IEEE Transactions on Image Processing*, 9(6):1025--1034, 2000.

[60] D. Kumar, C. S. Rai, and S. Kumar. Principal component analysis for data compression and face recognition. Technical report, Institute of Science and Technology, Klawad, 2008.

[61] A. Lanitis, C. J. Taylor, and T. F. Cootes. Automatic face identification system using flexible appearance models. *IMAGE AND VISION COMPUTING*, pages 97--12, 1995.

[62] Steve Lawrence, C. Lee Giles, and Ah Chung Tsoi. What size neural network gives optimal generalization? convergence properties of backpropagation. Technical report, 1996.

[63] Steve Lawrence, C. Lee Giles, Ah Chung Tsoi, and Andrew D. Back. Face recognition: A convolutional neural network approach. *IEEE Transactions on Neural Networks*, 8:98--113, 1997.

[64] Seong-Whan Lee and Stan Z. Li, editors. *Advances in Biometrics, International Conference, ICB 2007, Seoul, Korea, August 27-29, 2007, Proceedings*, volume 4642 of *Lecture Notes in Computer Science*. Springer, 2007.

[65] Bicheng Li and Hujun Yin. Face recognition using rbf neural networks and wavelet transform. In Jun Wang, Xiaofeng Liao, and Zhang Yi, editors, *ISNN (2)*, volume 3497 of *Lecture Notes in Computer Science*, pages 105--111. Springer, 2005.

[66] Huaqing Li, Shaoyu Wang, and Feihu Qi. Automatic face recognition by support vector machines. In Reinhard Klette and Jovisa D. Zunic, editors, *IWCIA*, volume 3322 of *Lecture Notes in Computer Science*, pages 716--725. Springer, 2004.

[67] Stan Z. Li, Jian-Huang Lai, Tieniu Tan, Guo-Can Feng, and Yangsheng Wang, editors. *Advances in Biometric Person Authentication, 5th Chinese Conference on Biometric Recognition, SINOBIOMETRICS 2004, Guangzhou, China, December 13-14, 2004, Proceedings*, volume 3338 of *Lecture Notes in Computer Science*. Springer, 2004.

[68] ShengCai Liao, XiangXin Zhu, Zhen Lei, Lun Zhang, and Stan Z. Li. Learning multi-scale block local binary patterns for face recognition. In Lee and Li [64], pages 828--837.

[69] Huchuan Lu, Wei Zhang, and Deli Yang. Eye detection based on rectangle features and pixel-pattern-based texture features. *Proceedings of 2007 International Symposium on Intelligent Signal Processing and Communication Systems*, pages 265--268, 2007.

[70] Yong Ma, Xiaoqing Ding, Zhenger Wang, and Ning Wang. Robust precise eye location under probabilistic framework. In *FGR*, pages 339--344. IEEE Computer Society, 2004.

[71] A. Mahalanobis, B. V. K. Vijaya Kumar, and S. R. F. Sims. Distance-classifier correlation filters for multiclass target recognition. *Appl. Opt.*, 35(17):3127--3133, Jun 1996.

[72] Abhijit Mahalanobis, B. V. K. Vijaya Kumar, and David Casasent. Minimum average correlation energy filters. *Applied Optics*, 26(17):3633--3640, 1987.

[73] Sébastien Marcel, Jean Keomany, and Yann Rodriguez. Robust-to-illumination face localisation using active shape models and local binary patterns. Idiap-RR Idiap-RR-47-2006, IDIAP, 0 2006. Submitted for publication.

[74] Aleix M. Martínez. Recognizing imprecisely localized, partially occluded, and expression variant faces from a single sample per class. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(6):748--763, 2002.

[75] Aleix M. Martínez and Avinash C. Kak. Pca versus lda. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(2):228--233, 2001.

[76] Janarbek Matai, Ali Irturk, and Ryan Kastner. Design and implementation of an fpga-based real-time face recognition system. In Paul Chow and Michael J. Wirthlin, editors, *FCCM*, pages 97--100. IEEE Computer Society, 2011.

[77] Iain Matthews, Jing Xiao, and Simon Baker. 2d vs. 3d deformable face models: Representational power, construction, and real-time fitting. *International Journal of Computer Vision*, 75(1):93--113, 2007.

[78] W. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5:115--133, 1943.

[79] K Messer, J Matas, J Kittler, J Luettin, and G Maitre. Xm2vtsdb: The extended m2vts database. In *Second International Conference on Audio and Video-based Biometric Person Authentication*, March 1999.

[80] B. Moghaddam, C. Nastar, and A. Pentland. A bayesian similarity measure for direct image matching. In *Proceedings of the 13th International Conference on Pattern Recognition - Volume 2*, ICPR '96, pages 350--, Washington, DC, USA, 1996. IEEE Computer Society.

[81] Baback Moghaddam and Alex Pentland. Probabilistic visual learning for object representation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(7):696--710, July 1997.

[82] O. Nikisins, R. Fuksis, M. Greitans, and M. Pudzs. Infrared imaging system for analysis of blood vessel structure. *Elektronika ir Elektrotechnika*, (1):45--48, 2010.

[83] O. Nikisins and M. Greitans. Local binary patterns and neural network based technique for robust face detection and localization. *Proceedings of the Special Interest Group on Biometrics and Electronic Signatures (BIOSIG 2012)*, pages 147--158, September 2012.

[84] O. Nikisins and M. Greitans. A mini-batch discriminative feature weighting algorithm for lbp - based face recognition. *Peoceedings of IEEE International Conference on Imaging Systems and Techniques (IST 2012)*, pages 170--175, July 2012.

[85] O. Nikisins and M. Greitans. Reduced complexity automatic face recognition algorithm based on local binary patterns. *Peoceedings of 19th International Conference on Systems, Signals and Image Processing (IWSSIP 2012)*, pages 447--450, April 2012.

[86] Olegs Nikisins, Modris Greitans, Rihards Fuksis, Mihails Pudzs, and Zanda Serzane. Increasing the reliability of biometric verification by using 3d face information and palm vein patterns. In Arslan Brömme and Christoph Busch, editors, *BIOSIG*, volume 164 of *LNI*, pages 133--138. GI, 2010.

[87] Zhiheng Niu, Shiguang Shan, Shengye Yan, Xilin Chen, and Wen Gao. 2d cascaded adaboost for eye localization. In *ICPR (2)*, pages 1216--1219. IEEE Computer Society, 2006.

[88] T. Ojala, M. Pietikainen, and D. Harwood. A comparative study of texture measures with classification based on feature distributions. *Pattern Recognition 29*, pages 51--59, 1996.

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

[90] E. Osuna, R. Freund, and F. Girosi. Training support vector machines: an application to face detection. In *CVPR*, pages 130--136. IEEE Computer Society, 1997.

[91] Alex Pentland, Baback Moghaddam, and Thad Starner. View-based and modular eigenspaces for face recognition. In *IEEE INTERNATIONAL CONFERENCE ON COMPUTER VISION and PATTERN RECOGNITION*, 1994.

[92] P. Jonathon Phillips, Patrick J. Flynn, Todd Scruggs, Kevin W. Bowyer, Jin Chang, Kevin Hoffman, Joe Marques, Jaesik Min, and William Worek. Overview of the face recognition grand challenge. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01*, CVPR '05, pages 947--954, Washington, DC, USA, 2005. IEEE Computer Society.

[93] P. Jonathon Phillips, Hyeonjoon Moon, Syed A. Rizvi, and Patrick J. Rauss. The feret evaluation methodology for face-recognition algorithms. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(10):1090--1104, 2000.

[94] Matti Pietikainen, Abdenour Hadid, Guoying Zhao, and Timo Ahonen. *Computer Vision Using Local Binary Patterns*. Computational Imaging and Vision. Springer, Dordrecht, 2011.

[95] Y. Rodriguez. *Face Detection and Verification using Local Binary Patterns*. PhD thesis, Ecole Polytechnique Federale de Lausanne, 2006.

[96] Yann Rodriguez, Fabien Cardinaux, Samy Bengio, and Johnny Mariéthoz. Measuring the performance of face localization systems. *Image Vision Comput.*, 24(8):882--893, 2006.

[97] H. A. Rowley, S. Baluja, and T. Kanade. Rotation invariant neural network-based face detection. In *CVPR*, pages 38--44. IEEE Computer Society, 1998.

[98] D. Rumelhart and J. McClelland. *Parallel Distributed Processing*. MIT Press, Cambridge, 1986.

[99] Marios Savvides and B. V. K. Vijaya Kumar. Efficient design of advanced correlation filters for robust distortion-tolerant face recognition. In *AVSS*, pages 45--52. IEEE Computer Society, 2003.

[100] H. Schneiderman and T. Kanade. Object detection using the statistics of parts. *International Journal of Computer Vision*, 56(3):151--177, 2004.

[101] Shai Shalev-Shwartz and Nathan Srebro. Svm optimization: inverse dependence on training set size. In William W. Cohen, Andrew McCallum, and Sam T. Roweis, editors, *ICML*, volume 307 of *ACM International Conference Proceeding Series*, pages 928--935. ACM, 2008.

[102] P. Sinha, B. Balas, Y. Ostrovsky, and R. Russell. Face Recognition by Humans: Nineteen Results All Computer Vision Researchers Should Know About. *Proceedings of the IEEE*, 94(11):1948--1962, January 2006.

[103] J. A. Snyman. *Practical Mathematical Optimization: An Introduction to Basic Optimization Theory and Classical and New Gradient-Based Algorithms*. Springer Publishing, 2005.

[104] M. Sonka, V. Hlavac, and R. Boyle. *Image Processing, Analysis, and Machine Vision*. Cengage-Engineering, third edition, 2007.

[105] Vitomir Struc, Jerneja Zganec-Gros, and Nikola Pavesic. Principal directions of synthetic exact filters for robust real-time eye localization. In Claus Vielhauer, Jana Dittmann, Andrzej Drygajlo, Niels Christian Juul, and Michael C. Fairhurst, editors, *BIOID*, volume 6583 of *Lecture Notes in Computer Science*, pages 180--192. Springer, 2011.

[106] Ning Sun, Haixian Wang, Zhen hai Ji, Cairong Zou, and Li Zhao. An efficient algorithm for kernel two-dimensional principal component analysis. *Neural Computing and Applications*, 17(1):59--64, 2008.

[107] K. K. Sung and T. Poggio. Example-based learning for view-based human face detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20(1):39--51, 1998.

[108] Xiaoyang Tan, Songcan Chen, Zhi-Hua Zhou, and Fuyan Zhang. Face recognition from a single image per person: A survey. *Pattern Recognition*, 39(9):1725--1745, 2006.

[109] Xiaoyang Tan, Fengyi Song, Zhi-Hua Zhou, and Songcan Chen. Enhanced pictorial structures for precise eye localization under incontrolled conditions. In *CVPR* [7], pages 1621--1628.

[110] Ninad Thakoor, Sungyong Jung, and Jean Gao. Hidden markov model based weighted likelihood discriminant for minimum error shape classification. In *ICME*, pages 342--345. IEEE, 2005.

[111] A. Torralba, K. P. Murphy, and W. T. Freeman. Sharing visual features for multiclass and multiview object detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(5):854--869, May 2007.

[112] M. A. Turk and A. P. Pentland. Face recognition using eigenfaces. In *Proceedings. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 586--591. IEEE Comput. Sco. Press, 1991.

[113] Roberto Valenti and Theo Gevers. Accurate eye center location and tracking using isophote curvature. In *CVPR* [6].

[114] Roberto Valenti, Zeynep Yücel, and Theo Gevers. Robustifying eye center localization by head pose cues. In *CVPR* [7], pages 612--618.

[115] V. N. Vapnik. *Estimation of Dependences Based on Empirical Data (in Russian)*. Nauka. Moscow. (English translation Springer Verlag, New York, 1982), 1979.

[116] V. N. Vapnik. *Statistical Learning Theory*. J Wiley and Sons. New York, 1998.

[117] P. A. Viola and M. J. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137--154, 2004.

[118] Paul A. Viola and Michael J. Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR (1)* [4], pages 511--518.

[119] Peng Wang, Matthew B. Green, Qiang Ji, and James Wayman. Automatic eye detection and its validation. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Workshops - Volume 03*, CVPR '05, pages 164--, Washington, DC, USA, 2005. IEEE Computer Society.

[120] Shimin Wang and Jihua Ye. Research and implementation of embedded face recognition system based on arm9. 2010.

[121] Yichen Wei and Litian Tao. Efficient histogram-based sliding window. In *CVPR*, pages 3003--3010. IEEE, 2010.

[122] Laurenz Wiskott, Jean-Marc Fellous, Norbert Krüger, and Christoph Von Der Malsburg. Face recognition by elastic bunch graph matching. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, 19:775--779, 1997.

[123] S. Yan, S. Shan, X. Chen, and W. Gao. Locally assembled binary (lab) feature with feature-centric cascade for fast and accurate face detection. In *CVPR* [6].

[124] M. H. Yang, D. J. Kriegman, and N. Ahuja. Detecting faces in images: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:34--58, 2002.

[125] Xiang-Yan Zeng, Yen-Wei Chen, Zensho Nakao, and Hanqing Lu. Texture representation based on pattern map. *Signal Processing*, 84(3):589--599, 2004.

[126] C. Zhang and Z. Zhang. A survey of recent advances in face detection. Technical Report MSR-TR-2010-66, Microsoft Research, Microsoft Corporation. One Microsoft Way. Redmond, WA 98052, June 2010.

[127] Daoqiang Zhang, Zhi-Hua Zhou, and Songcan Chen. Diagonal principal component analysis for face recognition. *Pattern Recognition*, 39(1):140--142, 2006.

[128] G. P. Zhang. Neural networks for classification: a survey. *Trans. Sys. Man Cyber Part C*, 30(4):451--462, November 2000.

[129] Guangcheng Zhang, Xiangsheng Huang, Stan Z. Li, Yangsheng Wang, and Xihong Wu. Boosting local binary pattern (lbp)-based face recognition. In Li et al. [67], pages 179--186.

[130] H. Zhang, W. G., X. Chen, and D. Zhao. Object detection using spatial histogram features. *Image Vision Comput.*, 24(4):327--341, 2006.

[131] L. Zhang, R. Chu, S. Xiang, S. Liao, and S. Z. Li. Face detection based on multi-block lbp representation. In Lee and Li [64], pages 11--18.

[132] Wenchao Zhang, Shiguang Shan, Wen Gao, Yizheng Chang, and Bo Cao. Component-based cascade linear discriminant analysis for face recognition. In Li et al. [67], pages 288--295.

[133] Haitao Zhao and Pong Chi Yuen. Incremental linear discriminant analysis for face recognition. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 38(1):210--221, 2008.

[134] Wen-Yi Zhao, Rama Chellappa, P. Jonathon Phillips, and Azriel Rosenfeld. Face recognition: A literature survey. *ACM Comput. Surv.*, 35(4):399--458, 2003.

[135] Zhiwei Zhu, Qiang Ji, Kikuo Fujimura, and Kuangchih Lee. Combining kalman filtering and mean shift for real time eye tracking under active ir illumination. In *ICPR (4)*, pages 318--, 2002.