Real-time Embedded Age and Gender Classification in Unconstrained Video

by

Ramin Azarmehr

Thesis submitted to the
Faculty of Graduate and Postdoctoral Studies
In partial fulfillment of the requirements
For the M.Sc. degree in
Computer Science

School of Electrical Engineering and Computer Science
Faculty of Engineering
University of Ottawa

Abstract

Recently, automatic demographic classification has found its way into embedded applications such as targeted advertising in mobile devices, and in-car warning systems for elderly drivers. In this thesis, we present a complete framework for video-based gender classification and age estimation which can perform accurately on embedded systems in real-time and under unconstrained conditions. We propose a segmental dimensionality reduction technique utilizing Enhanced Discriminant Analysis (EDA) to minimize the memory and computational requirements, and enable the implementation of these classifiers for resource-limited embedded systems which otherwise is not achievable using existing resource-intensive approaches. On a multi-resolution feature vector we have achieved up to 99.5% compression ratio for training data storage, and a maximum performance of 20 frames per second on an embedded Android platform.

Also, we introduce several novel improvements such as face alignment using the nose, and an illumination normalization method for unconstrained environments using bilateral filtering. These improvements could help to suppress the textural noise, normalize the skin color, and rectify the face localization errors. A non-linear Support Vector Machine (SVM) classifier along with a discriminative demography-based classification strategy is exploited to improve both accuracy and performance of classification. We have performed several cross-database evaluations on different controlled and uncontrolled databases to assess the generalization capability of the classifiers. Our experiments demonstrated competitive accuracies compared to the resource-demanding state-of-the-art approaches.

Acknowledgements

I would like to take the opportunity to appreciate my supervisors, Prof. Robert Laganière, for his invaluable guidance, encouragement, and persistent support throughout the project and, Prof. Won-Sook Lee, for providing me with all the necessary facilities for research, and sharing expertise.

I express my warm thanks to Christina Xu and Ali Osman Ors for their support at Cognivue Corporation, and my colleague Mohammad Esmaeel Mousa Pasandi for sharing his knowledge at VIVA Lab.

I am also extremely grateful to Masoumeh Shaneshin for her continuous support and encouragement, and my parents for making this venture possible.

Table of Contents

List of Tables

List of Figures

Nomenclature

2DPCA	Two Dimensional Principal Component Analysis
AAM	Aactive Appearance Model
AM	Anthropometric Models
AMF	Age Manifold
ANN	Neural Networks
APM	Appearance Models
CLAHE	Contrast Limited Adaptive Histogram Equalization
DoG	Difference of Gaussians
DP	Dynamic Programming
DPM	Deformable Part Models
DyWT	Dyadic Wavelet Transform
ECRM	Electronic Customer Relationship Management
EDA	Enhanced Discriminant Analysis
FLD	Fisher's Discriminant Analysis
FPGA	Field Programmable Gate Array
FPS	Filtered Preprocessing Sequence
fps	Frames Per Second
GA	Genetic Algorithm
HE	Histogram Equalization
HyperBF	Hyper Basis Function
ICA	Independent Component Analysis
LBP	Local Binary Pattern
LBPH	Local Binary Pattern Histogram

LDA	Linear Discriminant Analysis
LGBP	Local Gabor Binary Pattern
LUT	Look-up Table
MAE	Mean Absolute Error
MI	Mutual Information
MSLBP	Multi-scale Local Binary Pattern
PCA	Principal Component Analysis
RAM	Random Access Memory
RBF	Radial Basis Function
RGB-D	Red-Green-Blue-Depth Sensors
SIFT	Scale-Invariant Feature Transform
SOM	Self Organizing Map
SVM	Support Vector Machine
SVMAC	Support Vector Machine with Automatic Confidence

Chapter 1

Introduction

The human face is a rich source of information about the attributes of a person such as identity, ethnicity, age, gender, attractiveness, and behavior. Thanks to their strong visual capabilities and intelligence, human beings are able to accurately categorize these traits from the facial appearance at a glance. In spite of the apparent simplicity of recognition tasks in human beings, a great deal of effort has been put into developing computerized systems which are capable of doing the same task with similar degree of simplicity and accuracy. Essentially, the majority of these automatic facial trait classification systems are based on computer vision, and can be employed in industrial applications such as surveillance monitoring, security control, and targeted marketing systems.

However, despite the advent of novel classification methodologies, such vision-based systems are still far from ideal compared to human abilities. Because, the recognition rates of these classifiers are significantly compromised by the geometrical misalignment of the face image, or the variations in environmental illumination. Notably, the illumination problem in human beings is rectified by the sophisticated visual sensory receptor cells of our eyes (*i.e.*, Retina), and the powerful visual processor of our brain (*i.e.*, Visual Cortex) which under varying illumination conditions ensure a constant perception of true colors (*i.e.*, Reflectance). On the other hand, another problem with existing computer vision based approaches for facial trait classification is the requirement for high performance computer systems which prohibits the implementation of these classifiers on resource-constrained and mobile platforms.

In this thesis, we aim to investigate these outstanding challenges and present practical solutions for implementing a video-based age and gender classifier on embedded systems that is able to perform accurately in unconstrained environments. We introduce several novel improvements for face alignment and illumination normalization, and propose an effective segmental dimensionality reduction technique for face image representation. Also, robust discriminative classifiers for gender classification and age estimation are presented which have very low computational and memory requirements. We have conducted a series of evaluations on an embedded system, and obtained promising results that acknowledge the accurate and real-time performance of our classifiers.

1.1 Motivation

The first academic articles on age and gender classification were published in the late 1990s, and until the early 2000s the research was merely limited to academia. However, in recent years there is a growing demand for automatic gender classification, and age estimation systems in emerging industrial applications. For instance, following the increasing security threats, the airports are considering security measures at the security checkpoints to collect the ethnicity and gender information of the passengers, automatically. Another example is the utilization of an automatic age estimation system to deny under-aged internet users to access the web pages with inappropriate contents.

The targeted advertisement is another fast-growing technology that facilitates the advertising of consumer products to specific group of age or gender. In Section 2.1, we provide a detailed list of potential applications for automatic demographics classification. It should be noted that, these systems shall not be intrusive or require any cooperation from the user. For example, using the voice or the fingerprint for recognizing the age and gender is feasible, but these approaches are subject to security issues. This fact emphasizes the importance of *vision-based* approaches for demographics recognition.

Recently, the emerging applications of automatic age and gender classification for *mobile* devices have attracted the interest of researchers to develop robust classifiers that can work under unconstrained illumination conditions in real-time with minimal resource requirements. These mobile applications can range from the extra safety of cars for the

elderly people to human-robot interaction (see Section 2.1). However, the existing approaches for age and gender classification not only are sensitive to varying illumination and misalignment, but also are memory and computation intensive such that the mobile and other resource-limited platforms cannot afford their resource requirements.

These outstanding difficulties motivated us to conduct extensive research, and propose viable improvements for normalizing the geometric and photometric characteristics of the face images in unconstrained environments, and also enable the design and implementation of an accurate real-time age and gender classifier for embedded systems utilizing an enhanced dimensionality reduction technique.

1.2 Objectives

In a nutshell, our main objective is to design and implement a video-based an accurate gender classification and age estimation framework which can perform on embedded systems in real-time and under unconstrained conditions. To put it differently, we break down this main objective and list the resulting sub-objectives as follows:

1. **Face Acquisition:** The objective is to employ a fast face detector that is specifically designed for video sequences, and is able to track the face without re-performing the face detection for each input frame of the video, until the tracked face is lost. A facial landmark detector shall be used to facilitate the face alignment using the position of the key features of the face.

2. **Face Image Normalization:** In unconstrained environments the face is prone to variations in illumination which can result in incorrect classification. The same problem occurs for different skin colors. Hence, the system shall normalize and standardize the photometric characteristics without distorting the aging sings and wrinkles on the face image. In fact, unlike the ethnicity recognition systems, the age and gender classifiers would not need face skin color information for classification. Therefore, the face image can be represented by a single-channel gray-scale format in order to save memory and computational costs on embedded system.

3. **Face Representation:** The changes in head pose and facial expression can lead to displacement of the key features of the face (*i.e.*, eyes, nose, and mouth) which we refer to it as "localization errors". These small errors disrupt the comparability of the query images against the template face images in the training set. To counter this problem, our objective is to exploit a multi-scale face representation strategy for normalizing the geometric characteristics as well as extracting the most descriptive features from the face image.

4. **Dimensionality Reduction:** The foremost goal of our work is to enable the implementation of an age and gender classifier on the resource-limited embedded systems. This goal cannot be achieved using a large input training set and a high-dimensional face representation. Therefore, a dimensionality reduction strategy shall be utilized to reduce the redundancy in face representation without discarding the useful texture information.

5. **Age and Gender Classification:** A supervised and discriminative classification approach shall be employed to identify the category of a new query image based on a previously categorized training set of labeled face images. The age classifier shall be able to classify four age groups: 0-19, 20-36, 37-65, and 66+.

6. **Video-based classification:** Unlike the regular still-image-based classification, the still-to-still classification in video sequences is an *ill-posed* problem (see Section 4.5.2). Even a small and transient change in head pose, facial expression, or illumination can cause misclassification in each frame of the video. Therefore, the objective in here is to stabilize the results across multiple frames of the video by keeping the best results until the tracked face is lost. In general, a real-time demographics classifier shall be able to process 15 to 25 frames per second (fps).

7. **Embedded System Considerations:** In addition to dimensionality reduction for compressing the data, a portable and self-contained binary file format shall be designed to store the compressed training set information and all parameters of the classifiers. No parameters shall be hard-coded in the system (Section 4.5.3).

In Chapter 5, the evaluation results of these objectives and a detailed analysis of improvements in terms of accuracy, computation, and memory requirements are presented.

1.3 Contributions

Essentially, the main contributions of this thesis to the methodology of age and gender classification are based on minimizing the memory and computational requirements to enable a real-time performance on the resource limited embedded systems while achieving a comparable recognition rate to the existing state-of-the-art but resource-intensive systems. Figure 1.3.1 illustrates the general flow of the classification process in our system, and Figure 4.0.1 shows the full pipeline of our age and gender classification approach. Herein, we list a summary of the contributions of this thesis as follows:

1. **Correcting face alignment errors using the nose:** Using the distance between the eyes is a common approach to determine the cropping area of the face image. However, this approach is sensitive to the head's yaw angle, causing localization errors and degradation of recognition rate. We correct the misalignment using two landmark positions on the nose to compensate for over-scaling problem (see Section 4.1).

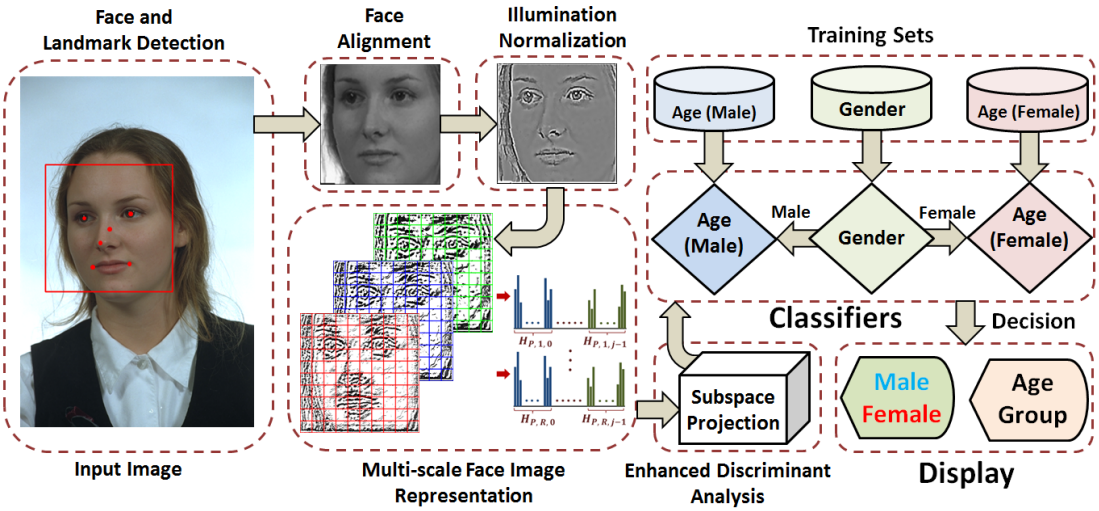


Figure 1.3.1: A general block diagram to illustrate the flow of the classification process

2. **Improving illumination normalization utilizing a sequence of filters:** A common gender misclassification problem is the false perception of gender from the androgynous faces illuminated by a light source at certain positions (Figure 4.2.1). This problem can misclassify a male as a female or vice versa. We propose to use the Preprocessing Sequence (PS) approach [119] to fix this problem as well as the variations of illumination in unconstrained environments. However, this method introduces textural noise in the face representation, and we present a practical solution to counter this problem using bilateral filtering (Section 4.3.1).

3. **Rectifying the localization errors in face representation:** The small and transient changes in facial expression or head pose can cause the displacement of key features of the face image. We propose to employ a multi-scale face representation to compensate for the localization errors (Section 4.3.1).

4. **Minimizing the resource requirements using segmental dimensionality reduction:** The redundancy and noise in feature vector can degrade the recognition rate. Also, classification based on a large input feature vector is deemed impractical on the resource-limited platforms. To conquer this limitation, we propose to utilize a *segmental* Enhanced Discriminant Analysis (EDA) technique which not only reduces the dimensionality, but also is able to retain only the most descriptive and discriminative features of the face. The segmental nature of this technique prevents the common problems of regular discriminant analysis methods such as overfitting and singularity (Section 4.4).

5. **Demography-based gender and age classification:** The Support Vector Machine (SVM) classifiers with non-linear RBF kernels are accurate, but are memory and computation-intensive. In contrast, the linear SVM classifiers are fast, but not accurate. However, our segmental dimensionality reduction technique allows the implementation of SVM with RBF kernel on the resource-constrained systems which otherwise is not possible using other approaches. Moreover, we introduce a generalization of demography-based gender and age classification which not only increases the accuracy, but also speeds up the classification process (Section 4.3.1).

It is worth noting that the contributions of our approach for age and gender classification have been accepted to be presented in the 11th IEEE Embedded Vision Workshop of Computer Vision and Pattern Recognition Conference (Boston, USA, 2015) [8].

1.4 Outline of the Thesis

We begin this thesis by reviewing the real-world applications, and the related works in the field of age and gender classification in Chapter 2. Due to differences in the core and evaluation methodologies, the age estimation and gender classification are reviewed separately. In Chapter 3, we introduce the theoretical perquisites of our approach which includes a general description of the common components of every facial trait classification system. This chapter provides the necessary information in order to prepare for Chapter 4 that presents the details of our contributions to the methodology of video-based age and gender classification for embedded systems.

In detail, Chapter 4 presents our improvements for the alignment and illumination normalization of the face image, and also our novel strategies for segmental dimensionality reduction and demography-based gender and age classification. In Chapter 5, we explain our embedded benchmarking setup used to evaluate our age and gender classifiers, and present a thorough analysis of the accuracy in comparison to the state-of-the-art approaches. Furthermore, we analyze the computational and memory requirements for the embedded systems. Finally, we conclude this thesis with a summary of presented material, and discuss the limitations as well as the future work and potential strategies to improve our age and gender classification system.

Chapter 2

Literature Review

Up to the present time, the outstanding challenges of automatic demographics classification are still attracting the interest of researchers. Particularly, the age and gender classification using computer vision has been given increased attention in recent years. Many researches have addressed the potential applications, and investigated the challenges that are associated with age and gender classification in real-world environments and videos. However, there are very few studies that have investigated the demographics classification problem for resource-constrained and embedded platforms. Regardless of the accuracy, most of the existing solutions have prohibitively large time and space complexities. Therefore, they are not able to perform in real-time on an embedded platform.

On the other hand, although age and gender classifiers have many components in common, they may be different in the core methodology of classification. In fact, there are certain methodologies that are suitable only for either age or gender recognition such as gait analysis for gender, or wrinkle analysis for age estimation. For this reason, we intend to survey the age and gender recognition methods in two separate sections in order to address the specific issues of each classifier in detail. In this chapter, we start by presenting an overview of the potential applications of automatic age and gender recognition in Section 2.1. Next, various robust approaches for gender classification and age estimation are surveyed in the Sections 2.2 and 2.3, respectively. Finally, we conclude this chapter in Section 2.4.

2.1 Real-world Applications

In recent years, automatic demographic classification has found its way into industrial applications such as surveillance monitoring, security control, video indexing, and targeted marketing systems. Many of such applications are based on computer vision and pattern recognition algorithms. However, in addition to vision-based systems, there are various other approaches such as gender and age recognition using iris [10, 110], fingerprint [125, 20], or audio [39]. But, the applications of these methods are limited since they require cooperation from the human. Also, they are intrusive and subject to privacy issues or security concerns [44]. Therefore, we mainly consider vision-based applications in this section.

As a matter of fact, implementing a demographic classifier on embedded platforms can extend its usefulness to even a wider variety of applications in mobile services. For instance, Feld *et al.*[39] applied automatic age recognition as a driver assistance system for elderly people. This system could provide additional safety features such as sustained in-car display of road traffic signs to compensate for decreased vision or reduced cognitive capacity. In another study [11], a priori gender categorization of a face during face recognition is used to speed up the comparison process between the perceptual input and the facial representation. This face recognition technique is convenient for computation-constrained embedded platforms.

Recently, the importance of demographics classification in surveillance monitoring and security control has become increasingly apparent. For instance, an age classifier can control the internet contents visited by under-aged user, and deny access to internet pages with unsuitable material [76]. Another example is the collection of demographic information such as ethnicity, gender, or age from the passengers at the security checkpoint to provide the security personnel with the statistics of passengers [115].

Electronic Customer Relationship Management (ECRM) [102, 44] is another fast-growing technology that facilitates marketing customized products and services based on customer's age or gender in an automatic and non-intrusive way. For example, an advertisement application on a mobile phone can recognize the gender and age via the embedded camera and display targeted ads for females (*e.g.*, lipstick), males (*e.g.*, wallet), or children (*e.g.*, toys).

Another common application for demographics classification is the content-based indexing which can be used for the retrieval of the face images from large databases [76]. Such automatic systems can index or annotate the demographic information of people in images or videos [90]. Therefore, based on gender or age categories, they can carry out content-based searching in the large image datasets, efficiently. The surveys in [90, 44] provide a detailed list of other potential applications for gender and age recognition such as human-robot interaction and biometrics.

2.2 Gender Classification

In here, we present a chronological survey of different gender classifiers. The majority of these classifiers require high performance computer systems; nonetheless, we will investigate the suitability of some classifiers for embedded systems. Generally, the vision-based gender classification can be grouped in two categories: (1) faced-based, (2) gait-based. In essence, several underlying components of these groups are different. For instance, the face-based approaches need a *face* detection stage to extract facial features, but gait-based methods require *human* detection algorithms to extract a binary silhouette of body for gender recognition. Different approaches based on these categories are surveyed in [85, 90]. However, we are mainly concerned with the face-based methods.

Generally speaking, the face-based approaches can be divided into *feature-based* and *appearance-based* methods [12]. Both of these categories can perform at global (holistic) or local (regional) level. Many of early face-based methods used an appearance-based model along with a multi-layer neural network method for classification. Perhaps, one of the pioneer studies in this field was conducted by Cottrell and Metcalfe [29] in 1990. They utilized a holistic representation of face called *holons*, as an input to a back propagation neural network to automatically classify the human face, emotion, and gender.

At the same time, Golomb *et al*. [52] introduced the "SEXNET" framework which was trained by a fully-connected back-propagation neural network. They evaluated this system on 90 exemplars, and reported an average error rate of 8.1%. In both of these studies, the faces were manually aligned. Later, a feature-based method in 1995 [99] proposed to extract a set of geometric features to feed a Hyper Basis Function (HyperBF) network. By

excluding the hair from the faces they achieved 79% success rate from the HyperBF classifier. In the same year, Abdi *et al*. [2] experimented with a Radial Basis Function (RBF) network that was preceded by an eigen-decomposition preprocessing step. They concluded that the recognition results using a pixel-based input can be comparable to measurement-based methods when the data are preprocessed.

Tamura *et al*. [118] employed neural networks to experiment on very low resolution face images of size 8×8 and achieved 93% classification rate. In [131], Gabor wavelets were placed on the nodes of a elastic bunch graph model that was manually aligned on the faces. The gender classification rate was 91.3% in this method. Lyons *et al*. [83] extended this method by automatically aligning the graph, and exploited Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) to classify gender. With 92% success rate, their classifier was slightly better than the previous work in [131].

In a groundbreaking study in 2001, Viola and Jones [127] proposed a robust cascaded face detector, which is by far an integral part of many face-based classifiers. The advent of Support Vector Machines (SVMs) [17] and Adaboost [43] classifiers were also a major break-through in pattern recognition research. In 2002, Shakhnarovich *et al*. [111] trained two classifiers for gender and ethnicity recognition by combining the threshold Adaboost and the cascaded face detector. Wu *et al*. [133] created a set of weak classifiers based on Look-Up Tables (LUT) and Adaboost. They claimed that LUT Adaboost can model the features that have multi-peak value distributions, unlike the threshold Adaboost in the previous method [111]. Later, Makinen and Raisamo [85] acknowledged this claim in their experiments.

One of the widely-cited approaches for gender recognition is proposed by Moghaddam and Yang [88], and it was considered state-of-the-art for several years. Evaluating on a public face image database and utilizing automatic face detection, face alignment, and image normalization method were the important differences of their method compared to others. They performed the evaluation using a Support Vector Machine (SVM) classifier with Radial Basis Function (RBF) kernel on 1755 face images from the public FERET database [96] and reported 96.6% gender classification rate.

Notably, their face detector was based on a maximum-likelihood estimation system which is 1000 times slower than the cascaded face detector of Viola and Jones [127]. Also,

Shakhnarovich *et al*. [111] pointed out that the same SVM classifier achieved 75.5% success rate on the collected images from internet. On the other hand, in terms of computation and memory requirements, the SVM+RBF classifier is known to be resource intensive and, therefore, not appropriate for embedded applications.

Another important factor to increase the recognition rate is *feature selection.* In a comparison study [117], a genetic algorithm (GA) was exploited to select a subset from a feature vector that was created using PCA. The subset was used to feed four different classifiers: Bayesian, neural network, SVM, and LDA. The results of comparisons demonstrated the superiority of SVM classifier with 95.3% gender recognition rate. Jian and Huang (2004) [67] employed independent component analysis (ICA) to extract facial features and used LDA to classify the gender. On manually cropped and normalized face images of FERET database [96], they claimed 99.3% recognition rate.

In the same year, Costen *et al*. [28] evaluated a sparse SVM classifier on Japanese face images, and achieved 94.42% classification rate. Sun *et al*. [116] proposed a novel feature-based representation of the face image using the Local Binary Patterns (LBPs) [91] for gender recognition. They experimented with both Self Organizing Map (SOM) and threshold Adaboost classifiers and reported 95.75% classification rate. Soon after, Lian and Lu [79] applied the SVM classifier on the same LBP feature vector and claimed 96.75% success rate.

The active appearance model (AAM) is another feature-selection strategy that was employed along with the SVM classifier by Saatci and Town [106]. Considering that facial expressions may affect the gender recognition results, they suggested to classify the facial expression first, and based on the detected expression perform the gender classification. However, this experiment decreased the success rate due to inadequate number of training images for different facial expressions.

Baluja and Rowly (2007) [9] defined a set of pixel comparison operators to create weak classifiers, and combined them into a single strong classifier using the Adaboost algorithm. They claimed that the classification accuracy was even better than the SVM classifiers that use the raw pixels for the input. The proposed pixel operators were fast to compute, and this method could outperform SVM classifiers with 50x faster classification. Therefore, it can be a good choice for real-time classification in resource-constrained and embedded

systems. Also, they concluded that the impressive results from Moghaddam and Yang [88] are biased due to evaluation on noise-free face images, and the existence of subjects with the same identity in different folds of FERET database.

To investigate the effects of face alignment on gender classification, Makinen and Raisamo (2008) [85] performed several experiments on 411 images from FERET database. They compared the results of classification using the aligned and unaligned faces, and the appearance-based and feature-based face representation models. They exploited different classifiers and achieved the best results using the SVM classifier followed by threshold Adaboost and Neural Network. In a novel research, Scalzo *et al*. [107] created a large feature vector by fusing the Gabor and Laplace filters in a hierarchy, and used a genetic algorithm for feature selection. They evaluate this classifier on 400 images and reported 3.8% error rate.

Inspired from the optimization of Fisher's discriminant ratio, Zafeiriou *et al*. [137] introduced a variant of SVM classifier with RBF kernel, and compared it to regular SVM classifiers. They achieved 2.8% overall error rate by evaluating the gender classifier on the XM2VTSDB [86] commercial database. Another variant of SVM classifier with automatic confidence (SVMAC) was proposed by Zheng and Lu [138], and was applied on a feature vector created from Local Gabor Binary Pattern (LGBP) [134]. They claimed that the SVMAC variant is 3% more accurate than the regular SVM classifiers.

Aghajanian *et al*. (2009) [3] proposed a Bayesian framework for gender and pedestrian pose classification by building a grid of non-overlapping patches of images. They evaluated the classifier on a custom face database of 500 females and 500 males, and achieved 89% correct recognition rate. In a fusion-based method [82], the classification results of three facial regions were combined to improve the robustness to facial expressions. To reduce the dimensionality of the feature vector, a two dimensional PCA (2DPCA) was used. Utilizing the SVM+RBF classifier, the recognition accuracy of this method was reported 95.33% on FERET database.

The scale invariant feature transform (SIFT) algorithm [81] is another widely-used feature extraction method in pattern recognition research. These features are invariant to rotation, translation, and scale of image. Demirkus *et al*. (2010) [34] applied a Bayesian classifier on a SIFT feature vector, and achieved 90% accuracy on an unconstrained video

sequence of 15 male and 15 female subjects. Soon after, Wang *et al*. [128] adopted Adaboost on a SIFT feature descriptor that was extracted at regular grid points, and fused it with the global shape contexts of the face image. They performed the evaluation using a 10-fold cross-validation on a mixture of images from different databases, and reported 97% accuracy. Alexandre *et al*. [5] extracted and combined the shape and LBP features of multiple image resolutions, and used linear SVM for gender classification. On a small subset of FERET database with 60 males and 47 females, they claimed up to 99% accuracy.

Bekios-Calfa *et al*. (2011) [12] experimented with SVM and Adaboost classifiers on LDA, ICA and PCA+LDA transformed features, and concluded that the accuracy of PCA+LDA transformation is better. Shan [113] employed an LBP feature selection strategy using the Adaboost algorithm and applied the SVM classifier with an RBF kernel on the boosted LBP features. The outcome of this method was 94.81% classification accuracy on the LFW [64] public database. Ullah *et al*. [123] tried LBP and Dyadic Wavelet Transform (DyWT) which is a multi-scale image transformation technique for gender recognition. They divided the image into non-overlapping blocks, extracted the DyWT+LBP face descriptor, and classified using SVM with 99% success rate on FERET.

Tapia *et al*. (2013) [120] used mutual information (MI) for feature selection with four different measures: (1) minimum redundancy and maximal relevance (mRMR) [35], (2) normalized mutual information feature selection (NMIFS) [37], (3) conditional mutual information feature selection (CMIFS), and (4) conditional mutual information maximization (CMIM) [25]. As a result, they achieved a real-time performance by reducing the dimension of feature vector. Fazl-Ersi *et al*. (2014) [38] integrated different feature descriptors such as LBP, SIFT and color histogram (CH), and employed a feature selection method [126] to extract the most informative features. The combination of these methods could achieve 91.59% classification rate on Ghallager [48] database.

Recently, deep learning algorithms have become popular in pattern recognition. Based on deep neural networks [63, 74], Eidinger *et al*. (2014) [36] proposed to combine a "dropout" technique with SVM classifier (dropout-SVM) in an effort to prevent overfitting due to scarcity of the available data. Also, they created the "Adience" face database, a very challenging database labeled for age and gender, by collecting 26,580 face images from 2,284 subjects in unconstrained environments. By training with Ghallager database

[48] and evaluating on the Adience database, they reported 77.8% success rate.

One of the few papers that have investigated the gender recognition on embedded systems is published by Irick *et al*. [66]. Training with 200,000 images, they implemented an *appearance-based* gender classifier on FPGA using Artificial Neural Networks (ANN). On a Xilinx Virtex-4 FPGA platform, they achieved a real-time performance with 83% accuracy by evaluating the classifier on a database of 3,826 images. Moreover, the SHORE object recognition engine from Fraunhofer[42] is a proprietary and embedded-friendly architecture that could achieve 94% gender recognition accuracy on BioID database [68]. The performance was 10 frames per second for gender recognition on Google Glass[129].

2.3 Age Estimation

In this section, we provide an overview of different age estimation approaches. For the sake of coherency, we comply with the same chronological format as presented in Section 2.2. To the best of our knowledge, there are no or very few studies for age estimation on embedded platforms. However, for some approaches we will investigate the resource requirements on embedded systems.

In general, the existing automatic age estimation methods are divided into two different groups: (1) age group classification (range of years), (2) actual age estimation (cumulative years lived). The age group classification has many similarities with gender classification, with the exception that it is a *multi-class* problem (*i.e*., no. classes > 2). In contrast, the actual age estimation is usually based on *regression* methods, or a hybrid of classification and regression to provide an estimated number for age.

Usually, the error measurement for the actual age estimation is reported using the Mean of Absolute Errors (MAE) between the estimated and the ground truth age labels [76]. That is to say $MAE = \sum_{k=1}^{N} |e_k - g_k|/N$, where e_k is the estimated age for the kth sample, g_k is the ground truth age, and N is the total number of images [44]. The surveys in [101, 44] have discussed different age estimation and error measurement approaches in detail.

Typically, the age estimation systems are consisted of two parts: (1) age image representation, (2) age estimation algorithm. Generally, there are five age image representation techniques which are briefly explained in below:

- **Anthropometric Models (AM):** It is based on the cranio-facial theory [6] which describes the growth of the head from infancy to adulthood. In other words, this is a mathematical model for the morphological changes in the human cranium as a result of growth [101]. Therefore, the age image can be represented by measuring the sizes and the relative proportions of the key features on the face (anthropometric features). Notably, in this method the estimation rates for young faces are higher.

- **Active Appearance Models (AAM):** A statistical method for coding the face model [26]. It utilizes the Principal Component Analysis (PCA) to learn a statistical shape and intensity model from a training set. In contrast to AM, the AAMs can represent all the ages, since they consider facial texture as well as the shape of the facial features.

- **Aging Pattern Subspace (AGES):** Geng *et al*. [50] proposed to build an aging pattern using a sequence of aging face images collected from *each* individual that is sorted in a temporal order. In the training stage, these sequences are projected into PCA subspace. Inevitably, there would be missing age images in each sequence for each person. Therefore, an EM-like iterative method is exploited to synthesize the missing images in the aging pattern subspace [49]. For age estimation, the aging pattern subspace is searched for the best match to the query face image that has the minimum reconstruction error. Then, the position of the matched image in the pattern is then reported as a number for the actual age [50].

- **Age Manifold (AMF):** The AGES method can be improved into a more flexible representation by building an aging pattern from *many* individuals at different ages [47]. Therefore, unlike the AGES method, the missing images at different ages can be obtained from other individuals. To learn a common aging pattern the manifold embedding technique [109] is used to learn a low-dimensional aging sequence from many images at each age.

- **Appearance Models (APM):** To represent a face image, the APM extracts facial features at global (holistic) or local (regional) level [59]. These features can be based on the shape of facial features (geometry), or the facial texture (wrinkles) [60]. The face

representation in this method is similar to the appearance-based gender classification method that described in Section 2.2.

Essentially, the early approaches for age estimation were based on age group classification. Perhaps, the earliest study in this area was published by Kwon and Lobo (1994) [75], and investigated the age group classification using anthropometric models (AM). They used six ratios computed from the distances of different facial features to classify the infants and adults (*e.g.*, eye to eye/eye to nose, eye to eye/eye to mouth). In addition to AM, they also incorporated an appearance model (APM) to characterize the density of facial wrinkles using the *snakelets [72].* This APM method was utilized to distinguish the young adults from the senior adults. Kanno *et al.* (2001) [70] employed Artificial Neural Networks (ANN) along with an APM representation of the face. They achieved 80% accuracy for classifying the four age groups of 110 male face images that were selected from FG-NET [1] public database.

A Support Vector Machine (SVM) classifier was exploited by Iga *et al.* (2003) [65], and applied on a feature vector consisted of geometric features, texture, and luminosity patterns. The reported accuracy was 67.4% for a five age groups classification that was evaluated on 300 subjects of 15 to 64 years old. In a novel approach, Lanitis *et al.* (2004) [76] adopted the Active Appearance Model (AAM) to estimate the actual age. They combined the shape and the intensity models, and extracted the principal components from the corresponding eigenspace to represent the face image. For age estimation, they utilized regression functions, age-based distribution functions, and neural networks. The Mean of Absolute Errors (MAE) was reported 3.82 to 5.58 years for estimating the age of 400 subjects from 0 to 35 years old.

Ueki *et al.* (2006) [122] formulated an age group classifier with 11 Gaussian models for each age group in a 2D-LDA+LDA feature subspace using an expectation-maximization (EM) algorithm. Basically, the classifier fits the query image to each Gaussian model and compares the likelihoods. They considered the age range of 3 to 85 years old, and achieved 50% age classification accuracy for males and 43% for females. The interesting idea of Aging Pattern Subspace (AGES) method was first published by Geng *et al.* [50]. Unlike the age estimator of Lanitis *et al.* [76] that used 50 AAMs, Geng *et al.* used 200 AAMs to encode the face images. Evaluating on FG-NET database, they reported a MAE of 6.77

years. However, in AGES method the problem is that, it assumes a face image exits in the pattern's subspace of the training database that is similar to the face of the query image at different ages. As a result, it is difficult to collect a large database containing the face images of the same subject at many different ages [44].

Later, Fu *et al*. (2007) [47] rectified the problems of AGES method using the Age Manifold (AMF) technique. The age manifold learning projects the images into low-dimensional manifold embedding space [109] by capturing the geometric structure and intrinsic data distribution. Instead of PCA, it employs the Orthogonal Locality Preserving Projections (OLPP) to project the image data and preserve the essential manifold structures. Using a quadratic regression approach they achieved a MAE of 8 years on 4000 test images. Fu and Huang [46], used Conformal Embedding Analysis (CAE) along with quadratic regression to improve the MAE down to 6 years. In a similar research, Guo *et al*. [54] proposed the Locally Adjusted Robust Regression (LARR) and applied it on OLPP subspace, achieving 5.07 years for the MAE on the FG-NET database.

Also, the Biologically Inspired Feature (BIF) [104] was demonstrated by Guo *et al*. (2009) [57] to be a successful age image representation strategy to further improve the accuracy of age estimation. This method is based on a feed-forward model of the primate visual object recognition pathway, namely, the "HMAX" model [44]. It is consisted of alternating layers of cell units called Simple (S) and Complex (C). The complexity of these layers increases from the primary Visual cortex (V1) to the Inferior Temporal (IT) cortex [57]. The first layer S1 is created by Gabor filtering and the second layer C1 from a "MAX" operation on S1. The BIF feature can effectively capture the aging patterns, and is invariant to small rotations, translations, and scale changes [44]. Adopting SVM classifier, Guo *et al*. [56] proposed a framework for age and gender estimation using the BIF and Age Manifold (AMF) features. They reported MAE of 2.61 years for females and 2.58 years for males on YGA database [46], and demonstrated the superior performance of BIF for age image representation.

The introduction of Gallagher [48] real-life face database in 2009 was a turning point for the age estimation researchers. The challenging face images of this large database were collected from Flickr, and labeled with 7 age groups: 0-2, 3-7, 8-12, 13-19, 20-36, 37-65, and 66+. The majority of recent age estimation methods have adopted normalization

strategies to deal with the severe distortion, head pose, and illumination problems of this database. For instance, Shan (2010) [112] used illumination-invariant appearance features such as Local Binary Patterns (LBP) [91] and Gabor to represent the real-life faces. In addition, Adaboost was adopted as a feature selection approach to learn the discriminative local features. By applying the SVM classifier with an RBF kernel on the boosted features, he reported 55.9% exact classification accuracy and 87.7% classification accuracy when the error of one age category is allowed. In a similar effort, Ylioinas *et al*. [136] improved the accuracy by creating regional histograms from the LBP features, and achieved 88.7% age classification accuracy.

Chang *et al*. (2011) [23, 22] proposed a ranker for ordinal hyperplanes that could separate all the facial images into two groups according to their relative order. In other words, they used a conventional binary classifier (*e.g*., SVM) to carry out piece-wise classification among k classes to find the rank (*i.e*., age) of the query face image. In order for this to work, a cost-sensitive strategy was employed to find better hyperplanes based on the classification costs. They achieved a MAE of 4.48 years on FG-NET database, and 6.07 years on MORPH database [103]. Alnajar *et al*. (2012) [7] proposed a soft assignment approach for encoding the face images by extracting and learning multiple codebooks [19] for individual face patches (*i.e*., local regions). They formulated a weighting scheme that softly assigns each pixel to multiple candidate codes. To build the feature vector, they computed the orientation histogram of the local gradients in each neighborhood. Compared to the results of Shan [112], the accuracy of this method was 3.6% better on Ghallager database [48].

One of the major problems in age estimation is the imbalanced training data due to lack of sufficient samples in some age groups (*e.g*., senior adults) compared to other classes (*e.g*., young adults). Chen *et al*. (2013) [24] solved this problem by extracting low-level visual features from sparse and imbalanced image samples, and projecting them into a cumulative attribute space [45] to learn a regression model. For k age groups, they considered $k-1$ binary attributes that each of them separates facial images above a certain age from all those below. Also, each attribute conditions all the other attributes, cumulatively. The MAEs for this regression model was reported 4.67 years for FG-NET database, and 5.88 years for MORPH database [103].

Inspired by deep neural networks [63, 74], Eidinger *et al*. (2014) [36] devised the dropout-SVM classifier and applied it on a feature vector built from Four Patch LBP codes (FPLBP) [132]. They claimed that this classifier is robust to overfitting and the problems with imbalanced training data. The classifier was evaluated on the Adience [36] and Gallagher [48] databases, and achieved 45% and 66% age group classification rates, respectively. Soon after, Fazl-Ersi *et al*. [38] proposed to build an appearance-based model by fusing the features from Local Binary Patterns (LBP) [91], SIFT [81] and a color histogram (CH). In addition, they employed the feature selection method in [126] to extract the most informative features. Performing a 5-fold evaluation on Ghallager [48] dataset, they reported a maximum age recognition rate of 63.01% on Gallagher database.

To the best of our knowledge, the only existing embedded approach for age estimation is the commercial object recognition engine (SHORE) from Fraunhofer[42]. On a Google Glass[129] device it processes 10 frames per second, and its age estimation accuracy is 6.85 years of MAE on FG-NET database.

2.4 Conclusion

This chapter has provided a chronological overview of the robust and state-of-the-art approaches in the realm of gender classification and age estimation, and their potential applications. Also, the strengths and weaknesses of some well-known methods for face image representation and classification have been discussed. Although in this dissertation our main concerns are the resource-constrained systems and embedded platforms, there exist no or very few correlated publications in this area. This fact highlights the importance of our efforts to investigate the requirements and viable solutions for age and gender recognition on embedded systems. In the next chapters, we present a thorough analysis of the required components and methodologies to address the classification issues for embedded platforms.

Chapter 3

Generic Facial Trait Classification

Generally speaking, many of the face-based trait classification approaches have a number of common components as the integral part of their classification pipeline. Figure 3.0.1 shows a block diagram of an automatic facial trait classification pipeline. Needless to say, all of the major components of this pipeline have been integrated into our embedded age estimation and gender classification system. Hence, before proceeding to describe the contributions and methodologies in our work, it is necessary to expound the prerequisites and fundamental theories behind each of these modules. As a matter of fact, there exist a plethora of algorithms and methods for each module, but their time and space complexities may be the key obstacles for adopting them. Therefore, in here we attempt to focus on the complications of implementing a real-time age and gender recognition system for resource constrained and embedded platforms.

As shown in Figure 3.0.1, three major parts of this pipeline are: (1) face image acquisition, (2) face image representation, (3) face-based classification. To acquire the input images, usually a 2D image sensor device is used that performs in the visible light spectrum,

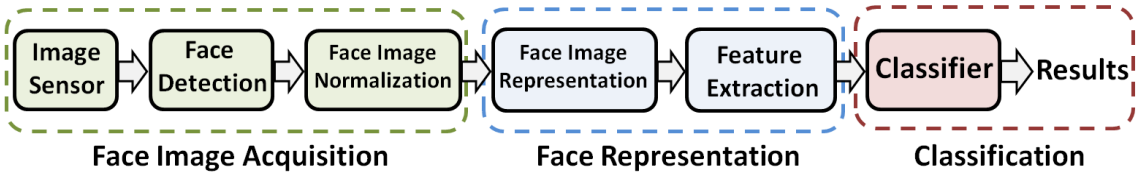


Figure 3.0.1: Block diagram of a generic facial trait classification system

and is highly affected by the illumination conditions in environment. In fact, there exist several image acquisition techniques that are invariant to illumination such as near-infrared and thermal sensors [77], or 3D face acquisition using RGB-D sensors [78]. However, due to limitations with the availability of such devices on embedded platforms, we use a regular visible light spectrum image sensor along with a robust image normalization approach to overcome the illumination problems.

Generally, the facial appearance representation methods can be categorized as global (holistic) and local (component-based) [61]. The holistic approaches are easier to implement, because the whole face is represented by a *single* feature vector. Normally, this single large feature vector is meant to feed the classifier's input, but there are several prohibitive problems associated with the size of such structures, known as the *curse of dimensionality*. Firstly, the bulky nature of such vector is at odds with the limited capabilities of embedded platforms. Secondly, the high degree of redundancy and presence of textural noise can drastically degrade the accuracy of classification.

In here, we refer to redundancy as the features that add no useful information to the feature vector. A common strategy to deal with redundancy in image data is to reduce the dimensionality of data by compressing the feature vector, and only extract the most discriminative features. In addition, the holistic methods are highly sensitive to changes in illumination, scaling, rotation, and translation of the face image.

In contrast, the component-based approaches aim to collect local facial features in order to compensate for the face localization errors and misalignment. This technique is proven [4] to amplify the robustness of classification against the changes in face pose and illumination by allowing a flexible geometric arrangement among the features of the face image. A widely-used scheme in component-based systems is to partition the face into overlapping or non-overlapping regions [4] and extract the regional information as the components. Typically, the fusion of these components constitutes a feature vector that feeds the classifier.

In this chapter, we present a generic description of the fundamental theories that we have used for each module of our age and gender recognition system. Also, we investigate the problems associated with the complexities of some approaches on resource-constrained systems, and will propose viable solutions for them in the next chapter. We start this chapter by describing the face detection module in Section 3.1, and image normalization techniques

in Section 3.2. Various image transformation and representation methods are explored in Section 3.3, and two dimensionality reduction approaches are presented in Section 3.4. Next, we review some classifiers and their suitability for embedded systems in Section 3.5. Finally, conclude this chapter in Section 3.6.

3.1 Face Detection

Face detection is the first step in face-based classification systems, and its accuracy affects the performance of classification, significantly. This task is far from trivial in a complicated scene that contains a variety of objects with different shapes. Nowadays, there exist various techniques for face detection that are surveyed in [135]. In general, face detection approaches can be categorized as feature-based and appearance-based methods. The feature-based methods extract certain features such as skin-color, edges, and geometric information from the face image while in appearance-based methods the whole face is used as an input to the face detector [85].

Up to the present time, perhaps the most commonly used face detector is the *cascaded* face detector proposed by Viola and Jones (2001) [127]. This technique utilizes a sweeping window to scan the image from top-left corner to bottom-right corner to find a face (Figure 3.1.1). This iterative process is repeated several times with different dimensions for the sweeping window to locate a face. In each iteration the content of the window is passed

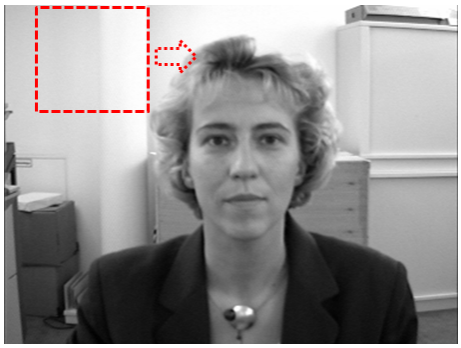


Figure 3.1.1: Sweeping window scans the image [68] from the top-left corner to the bottom-right corner to find a face

to a series of cascaded layers, each of which can reject the non-faces by comparing the extracted features of the face with a predefined face pattern. If the extracted features of the window are not rejected in a layer then they are passed to the next layer.

In case that the window is passed through all the layers successfully, then the window contains a face. Typically, the cascaded face detector is a fast algorithm, and is able to reject most of the non-faces in the early stages of detection. Also, there exist several memory-efficient implementations of this algorithm that can perform in real-time on embedded platforms [16].

3.2 Face Normalization

Essentially, the face-based classification systems are sensitive to geometrical misalignment, uneven illumination, and textural noise on the face image. Therefore, the face image should be normalized by aligning the face followed by photometric corrections and filtering operations. In fact, the photometric correction methods can standardize the representation of the face images that are acquired from the environments with different illumination conditions. For aligning a face image, a common strategy is to locate the position of key facial features (landmarks) and use them as references for geometrical normalization of the face image. In this section, first we describe the process of facial landmark detection which is a necessary module for the face alignment technique that is described in Section 4.1. Next, we introduce different techniques of photometric correction.

3.2.1 Facial Landmark Detection

In order to obtain the landmark positions from the face image, Uricar *et al.* (2012) [124] developed a memory-efficient and real-time facial landmark detector library called "flandmark". Figure 3.2.1 shows 8 landmark positions $\{\epsilon_0, \ldots, \epsilon_7\}$ detected by flandmark. This method exploits the concept of Deformable Part Models (DPM) [30] to create a structured output SVM classifier, and train the classifier using the annotated examples on the face. Given an input face image I and a set of quality scores for M facial landmark positions

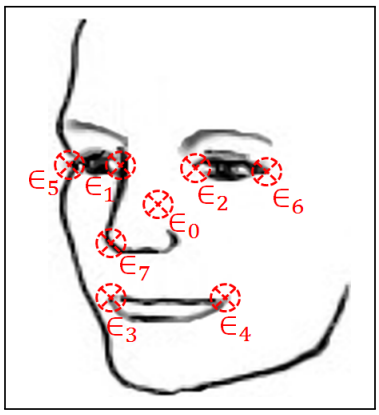


Figure 3.2.1: The position of eight facial landmarks detected by flandmark library [124]

$s = (s_0, \ldots, s_{M-1})$, they defined an optimization problem that maximizes the scoring function f, which is the sum of the appearance fit q, and the deformation costs Γ (see Equation 3.2.3):

$$f(I,s) = \sum_{i=0}^{M-1} q_i(I,s_i) + \sum_{i=1}^{M-3} \Gamma_i(s_0,s_i) + \Gamma_5(s_1,s_5) + \Gamma_6(s_2,s_6) + \Gamma_7(s_0,s_7) \tag{3.2.1}$$

$$q_i(I,s_i) = \left(w_i^q, \Psi_i^q(I,s_i)\right) \tag{3.2.2}$$

$$\Gamma_{ij}\left(s_i,s_j\right) = \left(w_{ij}^{\Gamma}, \Psi_{ij}^q\left(s_i,s_j\right)\right) \tag{3.2.3}$$

where Ψ_i^q and Ψ_{ij}^{Γ} are predefined maps, and w_i^q and w_{ij}^{Γ} are parameter vectors that will be learned from examples. To put it differently, this method models the landmark scores as a directed graph and localizes the nodes s_i of this graph on the facial features by fitting the graph on the appearance of the face. This fitting problem is solved by Dynamic Programming (DP) and a set of graph constraints. Section 4.1 provides a detailed procedure for face alignment using these detected facial landmarks.

3.2.2 Photometric Correction

As a matter of fact, the uneven illumination conditions in unconstrained environments can degrade the accuracy of classification, regardless of the robustness of the classifier. There are a variety of photometric normalization techniques that can neutralize the effect of shadows or over-illumination on certain regions of the face image. In this section, we present

and compare four commonly-used photometric normalization techniques. In Section 4.2, we present our approach for illumination normalization. Also, we compare the effectiveness of these methods on our classifier's accuracy in Chapter 5. The reader can refer to the survey in [62] to obtain detailed information of illumination normalization techniques.

- **Histogram Equalization (HE):** A fast method to enhance the global contrast of the image. Considering G gray levels per pixel, it transforms the distribution of N pixels with intensity values $v_{k\in[0,G-1]}$, into a uniform distribution using the transformation function T [62] (see Equation 3.2.4). On the negative side, in addition to global contrast, this method enhances the noise as well. Also, it is greatly influenced by the background noise. As shown in Figure 3.2.2(b), it is not effective to remove the shadows caused by a directed light source.

$$T(v_k) = (G-1)\sum_{i=0}^{k}\frac{n_i}{N} \tag{3.2.4}$$

- **Contrast Limited Adaptive Histogram Equalization (CLAHE):** Proposed by Pizer *et al*. **[98]** to improve the performance of regular histogram equalization by creating several locally equalized histograms using the transformation functions that are adapted for each local neighborhood. Moreover, it limits the contrast enhancement in each local neighborhood by clipping the upper parts of the local histograms that exceed a predefined threshold. As a result, the over-amplification of noise can be prevented by limiting the contrast enhancement. It should be noted that the clipped part of each histogram is not discarded and, instead, it is redistributed among the bins that their values do not exceed the clipping threshold. Figure 3.2.2(c) shows the effect of CLAHE method on the face images.

- **Retinex:** Inspired from the Human Visual System (HVS); particularly, the *retina* that is a preprocessing step to condition the visual data for facilitated high level analysis, and *V1 cortex area* which is a low-level visual information describer [15]. The two well-known channels of the retina's output are Parvocellular channel (Parvo) that is dedicated to detail extraction and Magnocellular (Magno) for motion information extraction. Nowadays, the bio-inspired models of the retina are widely-used in modern image processing applications to enhance the dynamic range compression and

color independence from the spectral distribution of the scene illumination [100]. In here, we are interested in Retinex (from the words retina and cortex) models, and the characteristics of Parvo channel for illumination normalization. Generally speaking, Retinex aims to estimation the reflectance component R from the luminance component L and the input image I, as follows:

$$R(x,y) = \frac{I(x,y)}{L(x,y)}$$

The advantage is that the reflectance, unlike luminance, is invariant to illumination and is resulted by the attenuation of the reflection from the surface of an object. Therefore, reflectance can serve as a means to derive an illumination invariant channel from the input image. Given the input image I, the luminance L can be estimated using a reflectance perception grid model proposed by Gross *et al.* [53] that minimizes the cost function:

$$J(L) = \underbrace{\iint \rho(x,y)(L-I)^2 dxdy}_{\text{perception gain model}} + \underbrace{\lambda \iint ({L_x}^2 + {L_y}^2) dxdy}_{\text{smoothness constraint}} \tag{3.2.5}$$

where λ is a parameter to control the relative importance of the two terms (see Equation 3.2.5), and $\rho(x,y)$ controls the anisotropic nature of the smoothness constraint. This calculus can be modeled by an Euler-Lagrange equation that is discretized on a rectangular lattice [53]:

$$\begin{aligned} I_{i,j} = &\frac{\lambda}{h\rho_{i,j-\frac{1}{2}}}\left(L_{i,j} - L_{i,j-1}\right) + \frac{\lambda}{h\rho_{i,j+\frac{1}{2}}}\left(L_{i,j} - L_{i,j+1}\right) + \\ &\frac{\lambda}{h\rho_{i-\frac{1}{2},j}}\left(L_{i,j} - L_{i-1,j}\right) + \frac{\lambda}{h\rho_{i+\frac{1}{2},j}}\left(L_{i,j} - L_{i+1,j}\right) + L_{i,j} \end{aligned} \tag{3.2.6}$$

where h is the pixel grid size, and $I_{i,j}$ is the intensity of a pixel at position (i,j). The weight ρ penalizes the smoothness at every edge of the lattice, and is formulated as:

$$\rho_{\frac{a+b}{2}} = \frac{|I_a - I_b|}{\min(I_a, I_b)}$$

where $\rho_{\frac{a+b}{2}}$ is a weight between two neighboring pixels with intensity values I_a and I_b [53]. Figure 3.2.2(d) shows the effect of this method on three face images with different illumination conditions.

- **Preprocessing Sequence (PS):** Introduced by Tan and Triggs [119] to counter the effects of illumination variations, local shadows and highlights without losing essential textural information for facial classification. This method starts by applying a gamma correction which is able to enhance the local dynamic range of the pixel intensity values in shadowed regions of the face, and at the same time, suppresses the bright regions. In order for this to work, it performs a non-linear transformation to replace the gray level pixel intensity values I of the input image with I^{γ}, where constant $\gamma \in [0,1]$. Next, it removes the intensity gradient and shading effects of the gamma corrected image I by convolving it with a *band-pass* filter like the Difference of Gaussians (DoG) filter Ψ in the following equation:

$$\Psi_{\sigma_{1,}\sigma_2}(x,y) = I * \left(\underbrace{\frac{1}{2\pi{\sigma_1}^2} e^{\frac{-(x^2+y^2)}{2{\sigma_1}^2}}}_{\text{Gaussian \#1}} - \underbrace{\frac{1}{2\pi{\sigma_2}^2} e^{\frac{-(x^2+y^2)}{2{\sigma_2}^2}}}_{\text{Gaussian \#2}} \right)$$

where σ_1 and σ_2 are two constants that determine the width of the two Gaussian kernels. This band-pass filter can effectively suppress the high frequencies caused by noise and aliasing artifacts, as well as the low frequencies caused by the illumination gradients. The novelty of this approach is a two stage contrast equalization strategy that re-normalizes the pixel value intensities and standardizes the global contrast. These two stages are formulated in the Equations 3.2.7 and 3.2.8.

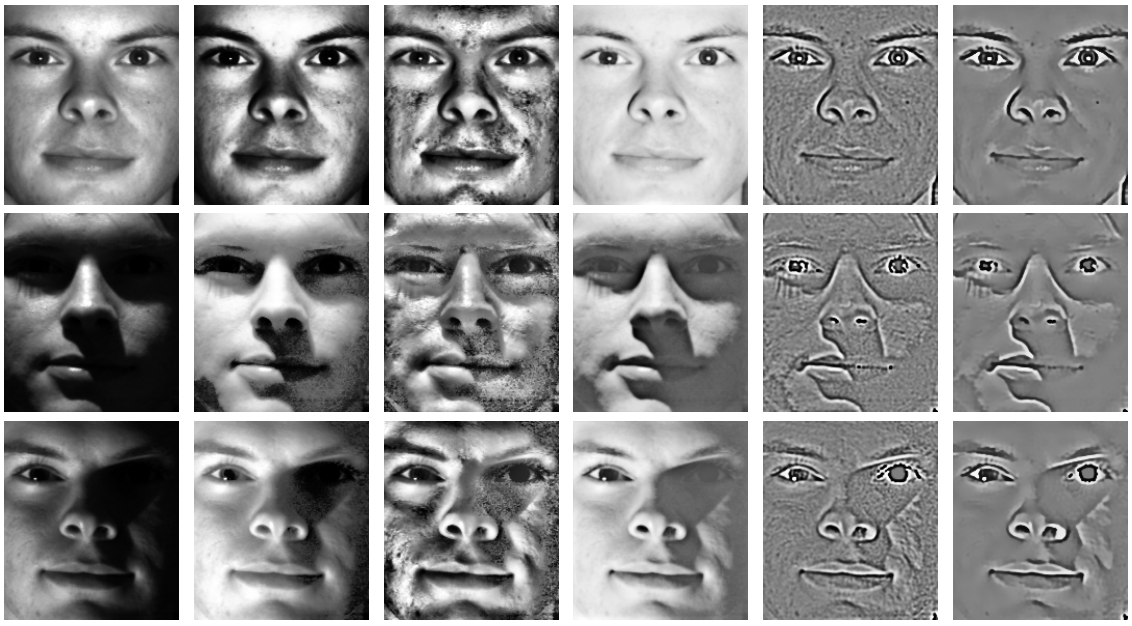
$$\Gamma(x,y) = \frac{\Psi(x,y)}{(\text{mean}(|\Psi(x,y)|^a))^{\frac{1}{a}}} \tag{3.2.7}$$

$$\phi(x,y) = \frac{\Gamma(x,y)}{(\text{mean}(\min(\tau, |\Gamma(x,y)|)^a))^{\frac{1}{a}}} \tag{3.2.8}$$

In these equations, a is used to reduce the influence of large values, and τ is a threshold that truncates the large values after the first stage of the normalization. Finally, a hyperbolic tangent is applied as a squashing function to normalize the extreme values within the range of $(-\tau,\tau)$.

$$\widehat{I}(x,y) = \tau \tanh\left(\frac{\phi(x,y)}{\tau}\right)$$

The output of the Preprocessing Sequence approach is shown in Figure 3.2.2(e). Notably, the default values of the constants used in this method are: $\gamma = 0.2$, $\sigma_1 = 1$, $\sigma_2 = 2$, $a = 0.1$, and $\tau = 10$.



(a) Original (b) H.E (c) CLAHE (d) Retinex (e) P.S (f) F.P.S

Figure 3.2.2: Effects of the different illumination normalization methods on three images [51]. The Filtered PS (F.P.S) is our normalization approach that is described in Section 4.3.

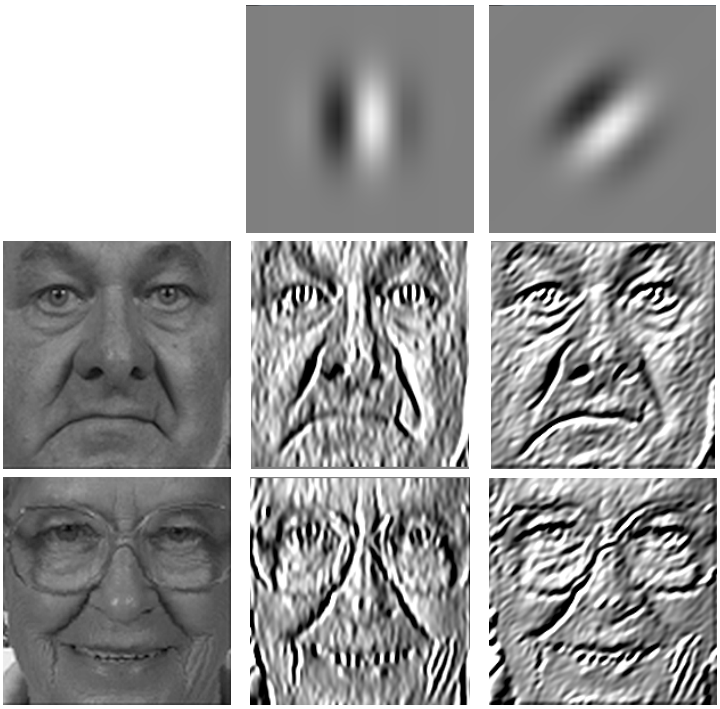
3.3 Face Representation

As demonstrated in Section 3.2, the illumination normalization methods help to enhance the contrast and improve the photometric characteristics of the face image. However, the remaining major problem is the negative effects of the geometrical displacements on the face image which are caused by the variations in facial expression or facial pose. By merely using the pixel intensity values for face representation, the classifier becomes highly sensitive to these variations, especially in unconstrained environments and video sequences. For this reason, a great deal of effort has been put into improving the robustness and stability

of face representation. In this section, we discuss the theories and the effectiveness of the variants of two well-known approaches, namely the Gabor features, and the Local Binary Patterns (LBP).

3.3.1 Gabor Wavelets

One of the earliest studies that considered the Gabor wavelet for computer vision applications was conducted by Daugman (1985)[33]. Later, Wiskott *et al*. [130] tailored the Gabor filters for face recognition. Similar to the Retinex method that we discussed in Section 3.2.2, the concept of Gabor wavelets are inspired from the human Retina. Utilizing Gabor wavelets, the face image can be represented by selective frequency and orientation features to enhance the key facial features like eyes, nose, mouth, and facial details like wrinkles and scars.



(a) Original (b) 0° (c) 45°

Figure 3.3.1: Examples of two Gabor kernels (orientations: 0°, 45°) applied on two images [96]

Essentially, the Gabor wavelet is a complex-valued function defined as a sinusoidal plane wave that is restricted by a Gaussian envelop with scale w and orientation v [83]:

$$\varphi\left(\mathrm{k}_{v,w},\mathrm{z}\right)=\frac{\|\mathrm{k}_{v,w}\|^2}{\sigma^2}\exp\left(\frac{-\|\mathrm{k}_{v,w}\|^2\|\vec{z}\|^2}{2\sigma^2}\right)\left[\exp\left(i\mathrm{k}_{v,w}.\mathrm{x}\right)-\exp\left(-\frac{\sigma^2}{2}\right)\right]$$

where $w\in[0,4]$ and $v\in[0,7]$. The subtraction of the term $\exp\left(-\frac{\sigma^2}{2}\right)$ makes the filter slightly invariant to global illumination in the face image. In the definition of the wave vector $\mathrm{k}_{v,w}$ in below, the parameter $\phi_u=\frac{\pi u}{8}$ controls the orientation, and $k_w=\frac{\pi}{2^{w+1}}$ controls the spaces between the kernels in the frequency domain:

$$\mathrm{k}_{v,w} \quad = \quad k_w\exp\left(i\phi_u\right)$$

In order to compensate for the localization errors in the face image, normally 5 scales for w and 8 orientations for v is used to build 40 Gabor filters. Finally, the Gabor image can be compute by the convolution of the Gabor filter and the face image $\mathrm{I}(z)$:

$$\mathrm{G}_{v,w}(\mathrm{z})=\mathrm{I}(z)*\varphi\left(\mathrm{k}_{v,w},\mathrm{z}\right)$$

Figure 3.3.1 shows the effects of two different Gabor filters on two face images. These terms can be convolved in Fourier domain to improve the computation time. In addition, PCA can be used to reduce the dimensionality of the Gabor images, and improve the memory requirements.

3.3.2 Local Binary Patterns

The advent of Local Binary Pattern (LBP) operator represented a major breakthrough in the field of object recognition. First time introduced by Pietikaeinen *et al*. [97] in 1994, the LBP operator has consistently demonstrated an excellent performance as a texture descriptor in various empirical studies for motion detection, remote sensing, visual analysis, and image retrieval. This operator is capable of capturing block-wise information with minimal computation and memory requirements while being invariant to the monotonic variations of illumination. Hence, the resource-constrained embedded systems can benefit from exploiting this efficient texture descriptor. This section provides a detailed explanation for several robust and popular variants of the LBP operator.

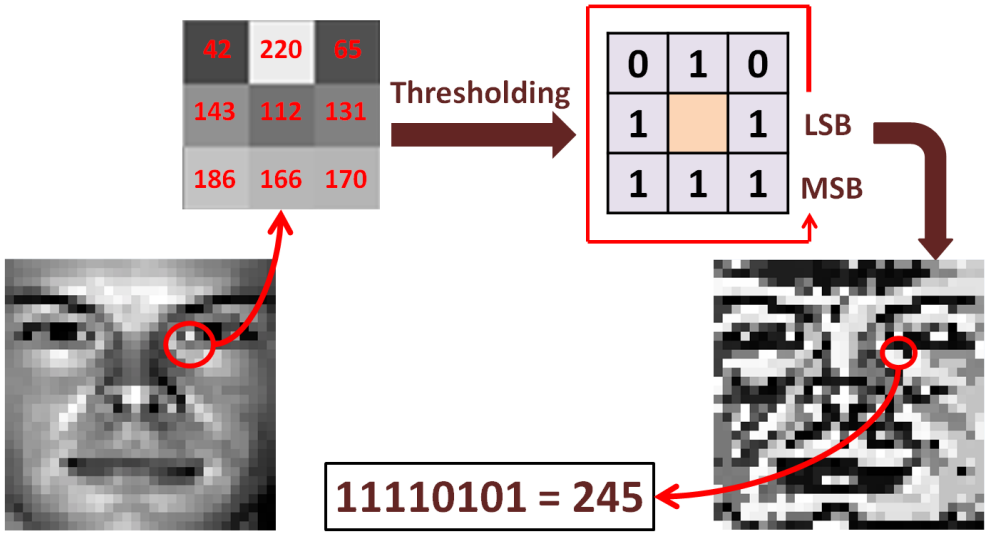


Figure 3.3.2: Illustration of the basic LBP operator on a 30 × 30 face image [51] (deliberately down-sampled to show the details).

- **Basic Local Binary Pattern:** For each pixel at a center position (x, y) of a circular neighborhood, the $LBP_{P,r}$ operator builds a binary sequence by applying the value of the center pixel as a threshold to P pixels in a circular neighborhood of radius r. Denoting the gray values of the center pixel as g_c and the surrounding pixels as g_p, the LBP can be defined as [91]:

$$LBP_{P,r}(x, y) = \sum_{p=0}^{P-1} 2^p \times s(g_p - g_c) \tag{3.3.1}$$

where $s(u)$ is 1 if $u \geq 0$ and 0 otherwise. Figure 3.3.2 illustrates the basic LBP operating on a face image that is deliberately down-sampled to show the details of the operation. The basic LBP generates 8-bits from a 3 × 3 block ($P = 8$). However, the circular neighborhood can be expanded to a wider radius including a higher number of pixels (*e.g.*, $P = 16$). A well-known strategy for LBP representation is to adopt the aggregate statistics such as LBP histograms (LBPH) [92]. As a result, the size of the texture descriptor can be further reduced from the image size to the number of histogram bins. Also, it can mitigate the effects of misalignment and affine transformations in the face image. Equation 3.3.2 shows the process of LBP histogram

creation.

$$H_i = \sum_{x,y} S(LBP_{P,r}(x,y) = i) \tag{3.3.2}$$

where $i \in \left[0, 2^P - 1\right]$, and $S(w)$ is 1 if w is true and 0 otherwise.

- **Uniform Local Binary Pattern:** The LBP features contain certain patterns, known as the *uniform* patterns, which occur frequently to represent the specific local structures such as the corners, line ends, edges, spots and flat areas. Conducting statistical analysis on different textures, Ojala *et al.* [92] concluded that the binary pattern of these structures contain at most two bit-wise transitions from 1 to 0, or 0 to 1. Therefore, they defined a uniformity measure to count the number of spatial transitions:

$$U\left(LBP_{P,r}\right) = |s\left(g_{p-1} - g_c\right) - s\left(g_0 - g_c\right)| + \sum_{p=1}^{P-1} |s\left(g_p - g_c\right) - s\left(g_{p-1} - g_c\right)| \tag{3.3.3}$$

  In Equation 3.3.3, if $U\left(LBP_{P,r}\right) \leq 2$ then the pattern $LBP_{P,r}$ is uniform. Considering this constraint, the total number of uniform patterns is $L = P(P-1) + 2$, and the number of histogram bins is $L + 1$, including an extra bin to accumulate the non-uniform patterns. Based on the uniformity measure $U\left(LBP_{P,r}\right)$, the uniform LBP operator is defined as:

$$LBP^{u2}_{P,r}(x,y) = \begin{cases} \sum_{p=0}^{P-1} 2^p \times s(g_p - g_c) & \text{if } U(LBP_{P,r}) \leq 2 \text{ (uniform)} \\ P(P-1)+2 & \text{otherwise (non-uniform)} \end{cases} \tag{3.3.4}$$

  Figure 3.3.3 shows the 58 possible uniform patterns for a circular neighborhood of 8 pixels (*i.e.*, $P = 8$) which are categorized by the representation of local structures, *i.e.*, lined ends, corners, edges, spot, and flat.

- **Rotation-Invariant Binary Local Pattern:** The in-plane rotation of the face image results in a different binary pattern, because the P pixels of each circular neighborhood are rotated around the center pixel as well. To rectify this problem, the bit-wise rotational right shift operator $\text{ROR}(w,i)$ is applied i times on a binary pattern $w = LBP_{P,r}$ until a minimal decimal value is found for w. The following equation illustrates this technique:

$$LBP^{ri}_{P,r}(x,y) = \min\left\{\text{ROR}\left(w,i\right) \mid i \in [0, P-1]\right\} \tag{3.3.5}$$

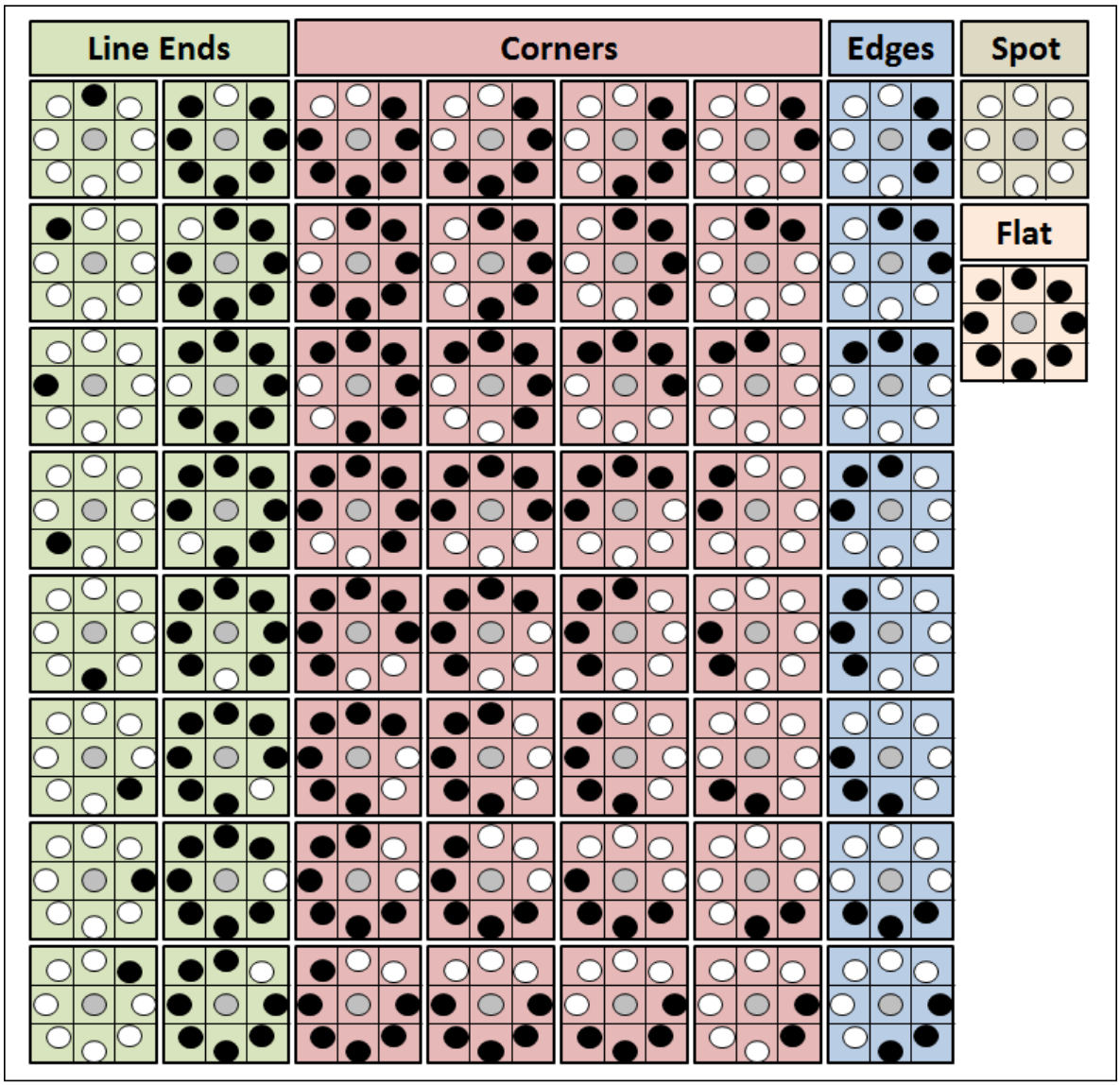


Figure 3.3.3: Example of all 58 uniform patterns for a circular neighborhood of 8 pixels categorized by the representation of line ends, corners, edges, spot, and flat structures (black circles represent the 1's of the binary sequence)

- **Local Ternary Patterns:** Essentially, the LBP operator performs robustly in the presence of monotonic intensity transformations. However, as can be seen in Figure 3.3.2, the thresholding process in LBP is highly sensitive to noise and non-monotonic transformations. To suppress the noise in LBP, Tan *et al.* [119] introduced the Local Ternary Patterns (LTP) operator that employs *hysteresis* thresholding. The dual threshold action in LTP creates a dead zone within a tolerance interval of $[g_c - t, g_c + t]$, around the gray value g_c of the center pixel, generating a ternary pattern in s. For LTP operator, we rewrite the $s(u)$ of Equation 3.3.1 as follows:

$$s(g_p, g_c, t) = \begin{cases} 1 & g_p \geq g_c + t \text{ (above positive threshold)} \\ 0 & |g_p - g_c| < t \text{ (within tolerance interval)} \\ -1 & g_p \leq g_c - t \text{ (below negative threshold)} \end{cases} \tag{3.3.6}$$

where t is a user-defined threshold that determines the width of the tolerance interval. In Equation 3.3.6, if the difference of the gray values for surrounding pixels g_p, and center pixel g_c exceed the upper threshold $g_c + t$, then $s(u)$ is 1, and if falls below the lower threshold $g_c - t$, it is -1, and 0 otherwise. To represent the LTP as a

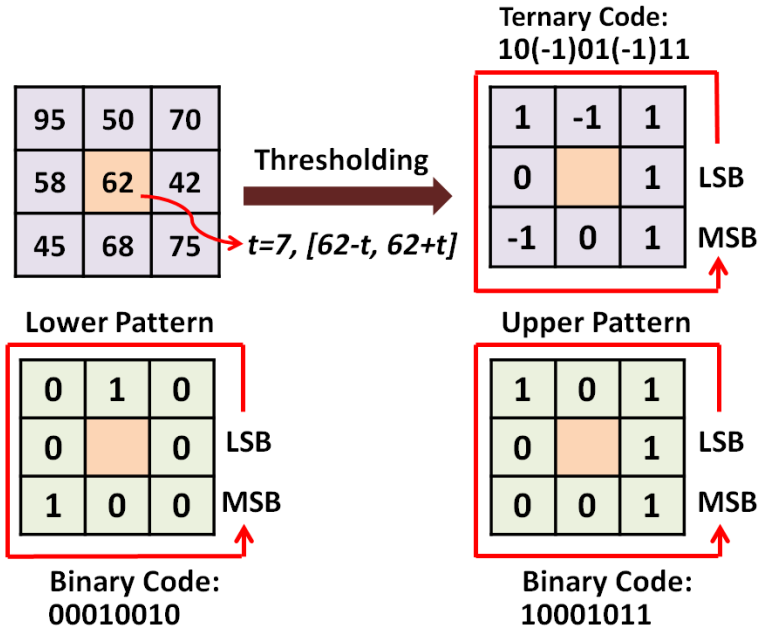


Figure 3.3.4: Illustration of the LTP operator

binary sequence, they suggested splitting the ternary pattern into upper and lower patterns, as illustrated in Figure 3.3.4. Therefore, it requires double the size of the LBP operator for storing the patterns.

3.4 Feature Extraction

Regardless of the use of image-based or feature-based representation of the vector data, there are several prohibitive problems associated with the dimension of these data structures, known as the *curse of dimensionality*. Specifically, most of the systems with limited resources cannot afford the large memory requirements of such texture representations. Therefore, we need a strategy to reduce the dimensionality of texture data, and extract its discriminative features. To this end, the generic approach is to transform a high-dimensional input data vector into a low-dimensional subspace in order to obtain the data vector. For this purpose, a generic subspace transformation can be defined as follows:

$$Y = \mathrm{XW} \tag{3.4.1}$$

where the input data vector $\mathrm{X} = [x_1, \ldots, x_n]^T$, the transformed vector $\mathrm{Y} = [y_1, \ldots, y_m]^T$, and the generic transformation matrix W:

$$\mathrm{W} = \begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,m} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n,1} & w_{n,2} & \cdots & w_{n,m} \end{bmatrix} \tag{3.4.2}$$

In this section, we discuss two classical approaches for dimensionality reduction, namely the Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA). In section 4.4, we will make use of these methods in our embedded classification system to find efficient transformation matrices that can reduce the dimensionality while preserving the discriminative features of the facial texture data.

3.4.1 Principal Component Analysis

First time proposed by Karl Pearson [95], the Principal Component Analysis (PCA) is a widely-used dimensionality reduction technique that projects a set of correlated variables into a set of linearly uncorrelated values called *principal components*. Karhunen [71] and Leove [80] further developed this method for signal processing, and modeled the PCA as an orthogonal linear transformation of a signal into eigenspace that yields a set of orthonormal basis vectors, namely the principal components. These vectors can optimally describe the underlying variance and internal structure of a dataset (*i.e.*, signal). The scatter-plot in Figure 3.4.1 shows two principal components of a two-dimensional dataset.

Notably, PCA transforms the data such that the first vector has the highest possible variance followed by the second largest vector which is orthogonal to the preceding vector, and so on for the rest of the succeeding vectors. In general, PCA can be computed using the eigen-decomposition of the data covariance matrix to derive its eigenvalues and eigenvectors. The eigenvectors represent the principal components and their associated eigenvalues represent the magnitude of the variance (*i.e.*, length of the corresponding eigenvector).

Considering a training set X that contains the representation of K face images, we define the k-th face image of this set as [121]:

$$x_k = \left[x_k^1, \ldots, x_k^N\right]$$

where N denotes the dimension of each face image that can be either the number of pixels for image-based representation, or the number of features for the feature-base approaches. The first step to compute the PCA is to normalize the samples of the training set by centering them using the mean of all samples μ:

$$\hat{x}_k = x_k - \mu, \text{ where } \mu = \frac{1}{K}\sum_{k=1}^{K} x_k \tag{3.4.3}$$

Accordingly, the mean-centered training set $\widehat{\mathrm{X}} \in \mathbb{R}^{N\times K}$ is created from all normalized samples:

$$\widehat{\mathrm{X}} = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,K} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,K} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N,1} & x_{N,2} & \cdots & x_{N,K} \end{bmatrix} \tag{3.4.4}$$

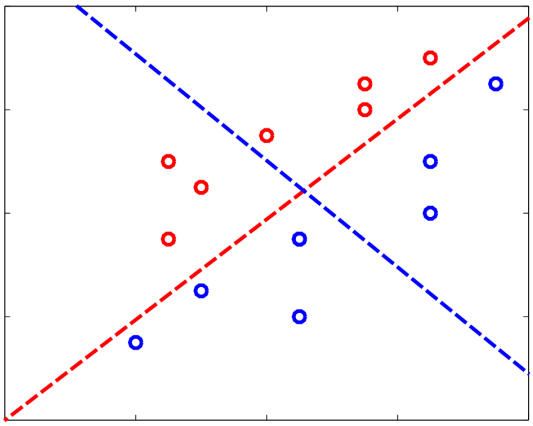


Figure 3.4.1: Two principal components of a 2D dataset. The red line represents the largest eigenvector (87% of the total variance)

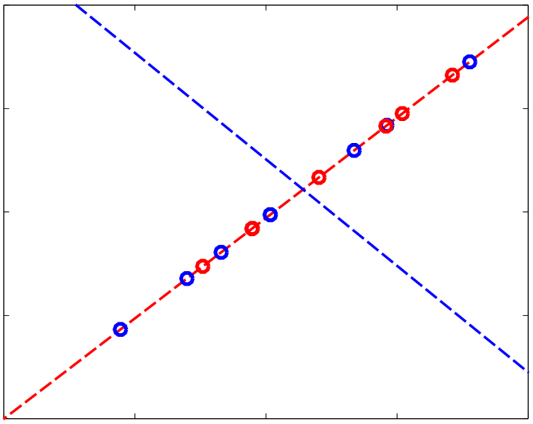


Figure 3.4.2: PCA projection on the first and largest principal component (red line) of a 2D dataset. The projected samples of the blue and red classes are one-dimensional (not linearly separable).

Now, the covariance matrix Γ is given as:

$$\Gamma = \frac{1}{K}\sum_{k=1}^{K} \hat{x}_k \hat{x}_k^T = \frac{1}{K}\widehat{\mathrm{X}}\widehat{\mathrm{X}}^T$$

where covariance matrix Γ can have a maximum of K eigenvectors associated with non-zero eigenvalues. Next, we feed the covariance matrix Γ to the eigen-decomposition stage in order to obtain the eigenvectors v_k and the corresponding eigenvalues λ_k:

$$\Gamma v_k = \widehat{\mathrm{X}}\widehat{\mathrm{X}}^T v_k = \lambda_k v_k$$

Considering that $\widehat{\mathrm{X}}\widehat{\mathrm{X}}^T$ is a huge matrix, we multiply both sides by $\widehat{\mathrm{X}}$ and use $\widehat{\mathrm{X}}^T\widehat{\mathrm{X}}$ to compute its eigenvalue decomposition as follows:

$$\widehat{\mathrm{X}}^T\widehat{\mathrm{X}}u_k = \lambda_k u_k \Rightarrow \widehat{\mathrm{X}}\widehat{\mathrm{X}}^T(\widehat{\mathrm{X}}u_k) = \lambda_k(\widehat{\mathrm{X}}u_k)$$

where u_k is the eigenvector for $\widehat{\mathrm{X}}^T\widehat{\mathrm{X}}$, and $v_k = \widehat{\mathrm{X}}u_k$ is the eigenvector for Γ. Finally, the eigenvectors are sorted in a descending order based on the magnitude of the associated eigenvalues. As a result, the principal components with the largest variations are concentrated in the lower-order portion of the eigenvectors.

As a matter of fact, we only preserve a portion of eigenvectors that represents the maximum amount of variation in data, and discard the eigenvectors with smaller eigenvalues that do not contribute to texture description. Thereby, the first advantage is that the dimensionality of input data is reduced, considerably. Consequently, the memory requirements and computation time is decreased. The second advantage is that the noise can be roughly eliminated from the texture representation thanks to the very small variations associated with the irregularly distributed noise data.

However, we need a proper strategy to preserve an *optimal* number of principal components without disposing useful information from the texture. There are various approaches for eigenvector selection, and we discuss some of them in here.

- **Standard eigenspace projection:** Retains all eigenvectors that are associated with the non-zero eigenvalues [73].

- **Preserve 60% of the eigenvectors:** As mentioned above, the eigenvectors are sorted in a descending order based on the magnitude of their eigenvalues. This method suggests to only keep 60% of the eigenvectors that have the largest eigenvalues [89].

- **Energy dimension:** This approach provides the flexibility to define a threshold for retaining a minimum number of eigenvectors that their cumulative energy function e_k exceeds the threshold [73]. Typically, the value for the user-defined threshold is greater than 0.9. This energy function is defined using the summation of the first k eigenvalues and the summation of all n eigenvalues:

$$e_k = \frac{\sum_{j=1}^{k} \lambda_j}{\sum_{j=1}^{n} \lambda_j} \tag{3.4.5}$$

- **Stretching dimension:** A common strategy to select eigenvectors is to compute the stretch s_k of the k-th eigenvector such that the ratio of the k-th eigenvalue λ_k over the maximum eigenvalue λ_m is greater than a threshold [73]:

$$s_k = \frac{\lambda_k}{\lambda_m}$$

- **Removing the three largest eigenvectors:** Unlike the methods mentioned in above, this approach discards the three eigenvectors with largest eigenvalues [89]. This is based on the assumption that the illumination variations contribute to the largest eigenvectors which can degrade the classification accuracy.

After eigenvector selection, similar to Equation 3.4.1 and 3.4.2, a mean-centered face representation X can be projected into eigenspace using the following transformation:

$$\mathrm{Y} = (\mathrm{W}^{PCA})^T \mathrm{X}, \text{ where } \mathrm{W}^{PCA} = u_k \tag{3.4.6}$$

The scatter-plot in Figure 3.4.2, shows the projection of the samples of Figure 3.4.1 on the first principal component, reducing the dimension of the 2D dataset to one dimension. As can be seen in Figure 3.4.2, although there are two classes with different set of samples (blue and red circles), the PCA projection could not yield a *linearly separable* representation in the eigenspace. This is a major problem for PCA-based classification methods. PCA can extract the most descriptive information, but is not able to discriminate the samples of different classes. In the next section, a practical solution is provided for this problem.

3.4.2 Linear Discriminant Analysis

With this fact in mind that PCA is an unsupervised approach which renders the classes of a projected dataset linearly inseparable, another strategy is required to provide the subspace projection process with the information of classes. To this end, Fisher [40] developed a *supervised* dimensionality reduction technique called Linear Discriminant Analysis (LDA) that aims to find a projection that linearly separates the distributions of two or more classes in the subspace.

The scatter-plot in Figure 3.4.3 shows the LDA component computed from a two-dimensional dataset. Also known as Fisher's Discriminant Analysis (FLD), Belhumeur *et al*. (1997) [13] employed FLD in face recognition for the first time (*i.e*., Fisher). In general, LDA attempts to maximize the ratio of between-class scatter over the within-class scatter. Similar to Equation 3.4.1 and 3.4.2, the objective is to find a transformation matrix W that projects the N-dimensional input data X with C classes onto LDA subspace data Y with $C-1$ dimensions such that:

$$\mathrm{Y}=\mathrm{W}^T\mathrm{X}$$

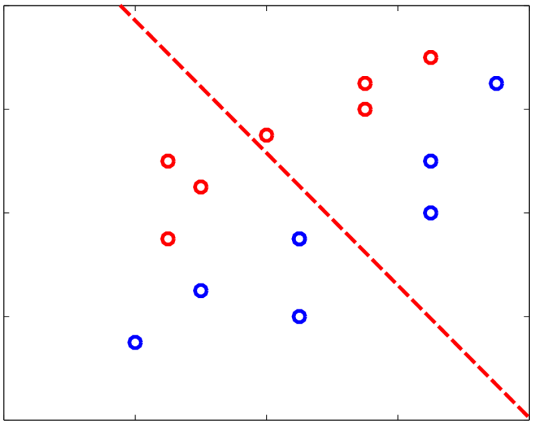


Figure 3.4.3: The component found by LDA on a 2D dataset. The red line represents the 1D LDA space.

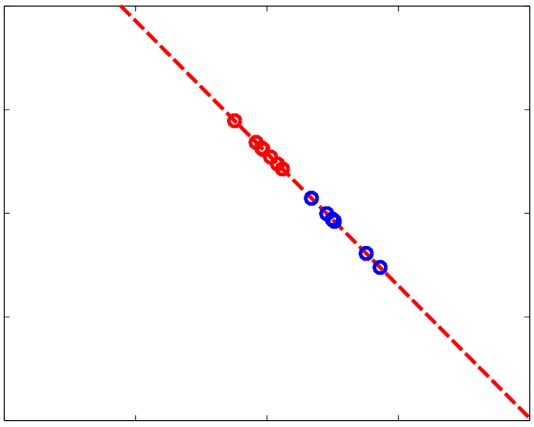


Figure 3.4.4: LDA projection of a 2D dataset on the 1D LDA subspace (red line). The projected samples of the blue and red classes are linearly separable.

The scatter-plot in Figure 3.4.4, shows the projection of the samples of Figure 3.4.3 on the LDA component, reducing the dimension of 2D dataset to one dimension. Clearly, it can be seen in Figure 3.4.4 that the two classes (red and blue circles) are separated and the within-class scatter is minimized.

Considering the same notations used in Section 3.4.1, we assume that the K samples of the N-dimensional input training set X is divided into C subsets $\mathrm{X}_i \in \{\mathrm{X}_1, \ldots, \mathrm{X}_C\}$ each of which represents a class that contains n_i samples. The *scatter* or the population variance σ^2 of the samples $x_j^i \in \mathrm{X}_i$ within each class is defined as:

$$\sigma_i^2 = \sum_{j=1}^{n_i} \left(x_j^i - \mu_i\right)^2 \text{,where } \mu_i = \frac{1}{n_i} \sum_{j=1}^{n_i} x_j^i \mid i \in [1, C], j \in [1, n_i] \tag{3.4.7}$$

$$\mu = \frac{1}{K} \sum_{j=1}^{K} x_j^i \mid i \in [1, C], j \in [1, K] \tag{3.4.8}$$

where μ_i is the mean of samples within each class X_i, and μ is the mean of all K samples in the input dataset X. Now, considering two classes X_1 and X_2, in order to maximize the ratio of between-class scatter S_B to that of within-class scatter S_W, Fisher proposed the

following criterion for maximization:

$$J(\omega) = \frac{\overbrace{|\mathrm{W}^T\mu_1 - \mathrm{W}^T\mu_2|^2}^{\text{between-class scatter}}}{\underbrace{{\sigma_1}^2 + {\sigma_2}^2}_{\text{within-class scatter}}} \tag{3.4.9}$$

where the between-class scatter can be rewritten as:

$$\begin{aligned}
|\mathrm{W}^T\mu_1 - \mathrm{W}^T\mu_2|^2 &= \left(\mathrm{W}^T\mu_1 - \mathrm{W}^T\mu_2\right)\left(\mathrm{W}^T\mu_1 - \mathrm{W}^T\mu_2\right)^T \\
&= \mathrm{W}^T\left(\mu_1 - \mu_2\right)\left(\mu_1 - \mu_2\right)^T\mathrm{W} \\
&= \mathrm{W}^T\mathrm{S}_B\mathrm{W}
\end{aligned}$$

And the within-class scatter can be rewritten as:

$$\begin{aligned}
{\sigma_1}^2 + {\sigma_2}^2 &= \sum_{j=1}^{n_1}\left(\mathrm{W}^T x_j^1 - \mathrm{W}^T\mu_1\right)^2 + \sum_{j=1}^{n_2}\left(\mathrm{W}^T x_j^2 - \mathrm{W}^T\mu_2\right)^2 \\
&= \sum_{j=1}^{n_1}\mathrm{W}^T\left(x_j^1 - \mu_1\right)\left(x_j^1 - \mu_1\right)^T\mathrm{W} + \sum_{j=1}^{n_2}\mathrm{W}^T\left(x_j^2 - \mu_2\right)\left(x_j^2 - \mu_2\right)^T\mathrm{W} \\
&= \mathrm{W}^T\mathrm{S}_1\mathrm{W} + \mathrm{W}^T\mathrm{S}_2\mathrm{W} = \mathrm{W}^T\left(\mathrm{S}_1 + \mathrm{S}_2\right)\mathrm{W} \\
&= \mathrm{W}^T\mathrm{S}_W\mathrm{W}
\end{aligned}$$

In Section 4.4, we will define the generalization of the scatter matrices S_B and S_W for all C classes, and will present a common approach to solve the Fisher's maximization problem. However, there are three important assumptions for LDA that should be taken into consideration:

1. The samples of all classes must be normally distributed which is not always possible.

2. The dimensionality of the input data must be less than $N - C$, otherwise the within-class scatter matrix S_W will be singular and the inverse of it cannot be computed.

3. The number of samples K in in the training set must be much higher than the dimension N of each sample (*i.e.*, $K \gg N$). Otherwise LDA computation will be subject to the same singularity problem for within-class scatter matrix S_W.

3.5 Classifier

In the field of vision-based pattern recognition, classification is considered as a supervised learning technique that identifies the category of a new query sample based on a previously categorized training set of well-defined and labeled samples. For the sake of comparison, a similarity measurement strategy is required to measure the degree of similarity between the query and each template image.

Some commonly-used classifiers in pattern recognition are: Fisher's Discriminant Analysis (FLD) (Section 3.4.2), boosting ensemble classifier, and the Support Vector Machine (SVM) classifier. In essence, these discriminative methods are *binary* classifiers (except FLD), but there are two common approaches to apply them on multi-class problems. The following strategies model a multi-class problem as multiple binary problems:

- **One-versus-one:** Classification is performed between every pair of classes and a max-wins voting scheme determines a category that gained the most votes.
- **One-versus-all:** The degree of similarity is reported from the classifiers and a winner-takes-all strategy is employed to determine the category that gained the highest degree of similarity.

In this section, we discuss the SVM and the boosting classifiers, in detail. The discussion is relevant to the description of our implementation in the next chapter. Because, the boosting classifier is used in our face detection module, and the SVM classifier is adopted by our embedded age and gender recognition system. Therefore, we present a detailed description for the classification algorithms.

3.5.1 Boosting Ensemble

Boosting refers to an effective and accurate prediction algorithm that combines a set of weak classifiers to form a single strong classifier. To put it differently, boosting learns from the ensemble of rough and moderately inaccurate prediction rules which their accuracies are only slightly better than random guessing. Adding each of these weak classifiers to the combination can *boost* the accuracy of the final classifier [108].

Algorithm 3.1 The Adaboost algorithm [43]

Initialize $w_1(i) = \frac{1}{m}$.

For $t = 1, \ldots, T$:

1. Train weak learner by finding a weak hypothesis $h_t : \mathrm{X} \to \{-1, +1\}$ that minimizes the error ϵ_t:

$$h_t = \underset{h_t \in \mathrm{H}}{\operatorname{argmin}}\, \epsilon_t, \text{ where } \epsilon_t = \sum_{i=1}^{m} w_t(i)\,[h_k(x_i) \neq y_i]$$

2. Terminate the loop if $\epsilon_t \geq \frac{1}{2}$.

3. Let $\alpha_t = \frac{1}{2}\ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$.

4. Update the weights:

$$\begin{aligned} w_{t+1}(i) &= \frac{w_t(i)}{Z_t} \times \begin{cases} e^{-\alpha_t} & \text{if } h_t(x_i) = y_i \\ e^{+\alpha_t} & \text{if } h_t(x_i) \neq y_i \end{cases} \\ &= \frac{w_t(i) e^{-\alpha_t y_t h_t(x_i)}}{Z_t} \end{aligned}$$

   where Z_t is a normalization factor in order to $\sum_{i=1}^{m} w_{t+1}(i) = 1$.

The final strong hypothesis is define as:

$$H(x) = \operatorname{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$$

Based on this hypothesis, Freund and Schapire [43] developed a robust boosting algorithm called Adaboost, which could solve the practical difficulties of the earlier boosting algorithms. Notably, the well-known *cascade* classifiers (Section 3.1) are based on such boosting algorithms [127].

Given an input training set $\mathrm{X} = \{x_1, \ldots, x_m\}$ consisted of samples $x_{i\in[1,m]}$ which are labeled by the corresponding labels $y_i \in \mathrm{Y} = \{y_1, \ldots, y_m\}$, the Adaboost algorithm repeats T iterations to produce a strong classifier H. It is assumed that $\mathrm{Y} = \{-1, +1\}$, and for each iteration $t \in \{1, ..., T\}$ there are a set of weights $w_t(i)$ where $i \in [1, m]$. The algorithm 3.1 illustrates the Adaboost algorithm. Typically, the boosting ensemble classifiers are fast algorithms that are able to reject most of the false patterns in the early stages of classification. Additionally, they require low amount of memory which makes them a good candidate for embedded systems with limited resources.

3.5.2 Support Vector Machine (SVM)

First time proposed and developed by Vapnik and Boser [17], the Support Vector Machine (SVM) and its variants are the most commonly-used supervised approaches for classification. In general, SVM is a discriminative binary classifier that attempts to find a separating hyperplane that has the widest margin to the closest training data sample of any class. This hyperplane acts as a *decision function* to predict the category to which a new query sample belongs.

Generally speaking, the wider the margin, the lower generalization error of the SVM classifier. Hence, SVM can be modeled as a maximization problem that finds a *maximum-margin hyperplane* with the largest possible distance to the nearest data points, or the so-called *support vectors* of each class. Figure 3.5.1 illustrates a linear SVM training process with two classes that produced a maximum-margin hyperplane and two *marginal hyperplanes*.

Given an input training set $\mathrm{X} = \{x_1, \ldots, x_m\}$ with two classes consisted of N-dimensional samples $x_{i\in[1,m]}$ which are labeled by the corresponding set of labels $\mathrm{Y} = \{y_1, \ldots, y_m\} \mid y_i \in [-1, 1]$, the goal of SVM is to find a maximum-margin hyperplane that separates the samples labeled as $y_i = -1$ from those of $y_i = 1$. This hyperplane can be defined using

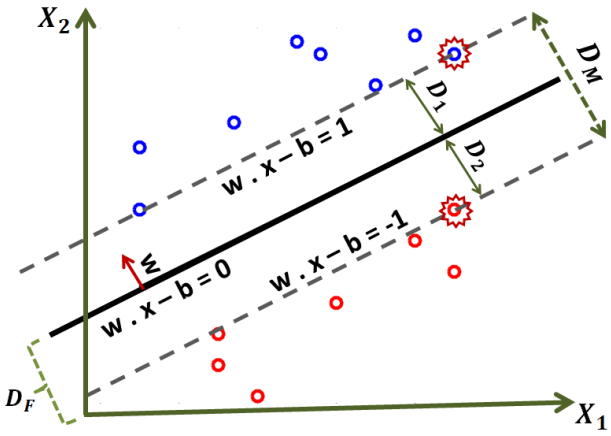


Figure 3.5.1: Illustration of the linear SVM training on the samples of two classes (blue and red circles). The starred samples on the marginal hyperplanes (dotted gray line) are support vectors, and the thick black line is the maximum-margin hyperplane

its normal vector w such that it satisfies $\mathrm{w.x} - b = 0$; where $\frac{b}{\|w\|}$ denoted as D_F in Figure 3.5.1, is the perpendicular distance between the hyperplane and the origin, and "." denotes the dot product operation.

Also, the maximum-margin distance $D_M = \frac{2}{\|\mathrm{W}\|}$ is the sum of $D_1 = \frac{1}{\|\mathrm{W}\|}$ and $D_2 = \frac{1}{\|\mathrm{W}\|}$ which are the distances from the two marginal hyperplanes to the maximum-margin hyperplane. The marginal hyperplanes are defined by the equations $\mathrm{w.x} - b = -1$ and $\mathrm{w.x} - b = 1$. Needless to say, there are no samples within the range of maximum-margin imposing the following constraints [31]:

$$\begin{cases} \mathrm{w.x}_i - b \geq 1 & \text{for } y_{i=+1} \\ \mathrm{w.x}_i - b \leq 1 & \text{for } y_{i=-1} \end{cases} \tag{3.5.1}$$

These constraints can be combined to formulate an optimization problem to maximize the margin that satisfies:

$$\underset{(\mathrm{W},b)}{\operatorname{argmin}} \|\mathrm{w}\| \text{ such that } y_i\,(\mathrm{w.x}_i - b) \geq 1, \text{ where } i \in [1, m] \tag{3.5.2}$$

By substituting the term $\|\mathrm{w}\|$ with $\frac{1}{2}\|\mathrm{w}\|^2$ and utilizing Lagrange multipliers α, we rewrite

the problem in primal form:

$$\arg \min_{(\mathrm{W},b)} \max_{\alpha_i \geq 0} \frac{1}{2}\|\mathrm{w}\|^2 - \alpha\left(y_i\left(w.\mathrm{x}_i - b\right) - 1\right)$$
$$\Rightarrow \arg \min_{(\mathrm{W},b)} \max_{\alpha_i \geq 0} \frac{1}{2}\|\mathrm{w}\|^2 - \sum_{i=1}^{m} \alpha_i y_i \left(w.\mathrm{x}_i - b\right) + \sum_{i=1}^{m} \alpha_i \tag{3.5.3}$$

Solving the optimization problem using quadratic programming techniques, we get the normal vector w as:

$$\mathrm{w} = \sum_{i=1}^{m} \alpha_i y_i \mathrm{x}_i \tag{3.5.4}$$

Considering that a few number of α_i will be greater than zero, the corresponding x_i samples will exactly represent N_s number of support vectors $\mathrm{x}_{s\in[1,N_s]}$ with labels y_s lying on the marginal hyperplanes. Hence, we can obtain the offset b from the support vectors:

$$b = \frac{1}{N_s} \sum_{s=1}^{N_s} \left(\mathrm{w}.\mathrm{x}_s - y_s\right) \tag{3.5.5}$$

A major problem with classification methods is the *misclassification* of samples due to inseparability of the distributions between the classes. As a result, the SVM trainer is not able to find a maximum-margin hyperplane that can clearly separate the classes. Particularly, this problem is very common in facial trait classification applications due to similarity of the subjects in the face images of different classes (see Section 4.5).

To counter this problem, Cortes and Vapnik [27] modified the SVM and adopted a *soft margin* technique that allows the misclassified samples with an associated *penalty cost* proportional to their misclassification error. Thus, they suggested to relax the constraints in Equation 3.5.1 by introducing a positive slack variable $\xi_{i\in[1,m]}$ into the optimization problem of Equation 3.5.3, and reformulate the Lagrangian as follows [27]:

$$\arg \min_{(\mathrm{W},\xi,b)} \max_{\alpha_i,\beta_i \geq 0} \frac{1}{2}\|\mathrm{w}\|^2 + C\sum_{i=1}^{m} \xi_i - \sum_{i=1}^{m} \alpha_i \left(y_i\left(\mathrm{w}.\mathrm{x}_i - b\right) - 1 + \xi_i\right) - \sum_{i=1}^{m} \beta_i \xi_i \tag{3.5.6}$$

where the constant parameter C controls the balance between the maximum-margin size and the penalty of slack variable.

On the other hand, in some classification problems the training samples of different classes are not *linearly* separable. An example could be the training samples of a class that

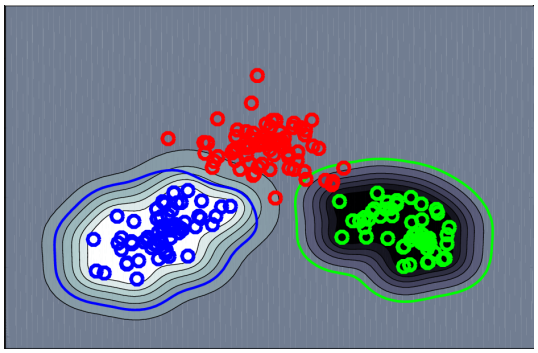


Figure 3.5.2: Example of SVM training for three classes with RBF kernel ($C = 1, \gamma = 10$). The maximum-margins are wide enough to minimize the generalization error. The red samples in the territory of the green class are penalized to compensate for misclassification.

are encircled by the training samples of another class. To solve this problem, Boser *et al.* [17] suggested applying the *kernel trick* to find the maximum-margin for non-linear problems. For this purpose, the kernel trick approach replaces every dot product by a non-linear kernel function k that transforms the data into a high dimensional feature space Φ, where a linear maximum-margin hyperplane can be found to separate the classes. Therefore, we can rewrite the Equation 3.5.4 as:

$$\mathrm{w} = \sum_{i=1}^{m} \alpha_i y_i k(\mathrm{x}_i, \mathrm{x}), \text{ where } k(\mathrm{x}_i, \mathrm{x}) = \Phi(\mathrm{x}_i).\Phi(\mathrm{x}) \tag{3.5.7}$$

The three common kernel types are:

- **Polynomial kernel:** $k(\mathrm{x}_i, \mathrm{x}_j) = (\mathrm{x}_i.\mathrm{x}_j + a)^d$, where d is the polynomial degree and $a$is a constant.
- **Sigmoidal kernel:** $k(\mathrm{x}_i, \mathrm{x}_j) = \tanh(a\mathrm{x}_i.\mathrm{x}_j - b)$, where a and b are the sigmoidal constants.
- **Radial Basis Function (RBF) kernel:** $k(\mathrm{x}_i, \mathrm{x}_j) = e^{-\gamma\left(\|\mathrm{x}_i - \mathrm{x}_j\|^2\right)}$

In fact, the Gaussian-like RBF kernel is the most popular and reliable kernel for non-linear classification. Figure 3.5.2 shows an example of a RBF kernel applied on a problem with three classes. As can be seen in this example, the maximum-margin Gaussians are found

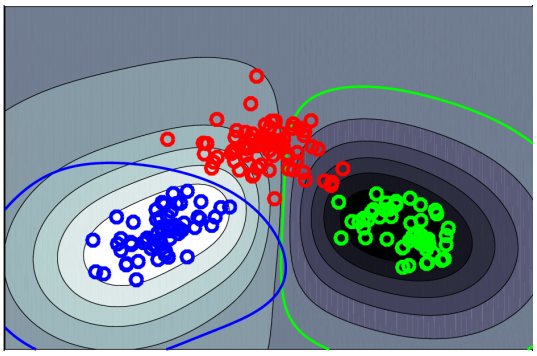


Figure 3.5.3: Example of **underfitting** in SVM training of three classes with RBF kernel ($C = 1, \gamma = 1$).

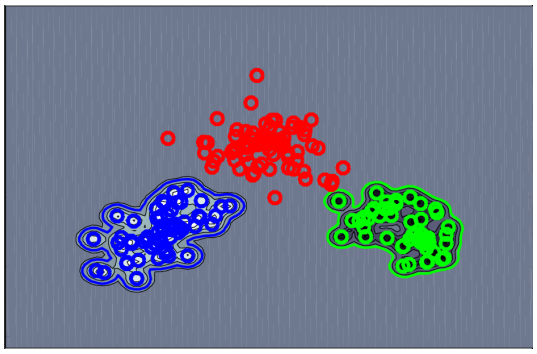


Figure 3.5.4: Example of **overfitting** in SVM training of three classes with RBF kernel ($C = 1, \gamma = 100$).

for the blue and green classes, and a query sample that does not lay into the territory of these Gaussians is categorized as the red class. Although this kernel is very accurate, there are two problems associated with it:

1. **Large training data size:** The decision boundaries that encircle each class are consisted of numerous Gaussians each of which is close to a support vector. Considering that the size of training data is proportional to the number of support vectors, in a large and high-dimensional training input the training data size may become so large that the host platform could not be able to afford the memory and computation requirements for classification. Specifically, this is a major problem for embedded systems with limited resources. In Section 4.5, we investigate and provide viable solutions for this problem.

2. **Optimal values for RBF parameters:** Another challenging problem is to find optimal value for the RBF parameters C and γ such that the generalization error of the classifier is minimized. As a matter of fact, assigning inappropriate values to these parameters can cause *underfitting* or *overfitting* phenomena both of which can increase the generalization error and compromise the classification accuracy, significantly. Figures 3.5.3 and 3.5.4 show the examples of underfitting and overfitting phenomena, respectively. The maximum-margins are widened in Figure 3.5.3 and shrunk in Figure 3.5.4, unreasonably. A typical solution to this problem is to exploit

grid search and *cross-validation* to examine different combinations of C and γ, and finally select an optimal combination that achieves the best results.

3.6 Conclusion

In fact, the described classification problem in this thesis, namely age and gender recognition, is a specific type of the facial trait classification systems which share many major components. With this in mind, it deems necessary to expound the prerequisites and fundamental theories behind each of these components to prepare for describing our embedded implementation in the next chapter. We have grouped the modules of the facial trait classification pipeline into face image acquisition, representation and classification.

In this chapter, we started with a brief description of cascade classifiers used for face detection, and presented a robust facial landmark detector to be used for geometrical face image alignment. Also, we described and compared different photometric and illumination normalization techniques such as Retinex and Preprocessing Sequence (PS). Next, two image representation methods, namely the Gabor Wavelets and the Local Binary Patterns (LBP) were explored.

In order to reduce the dimensionality of image representation, the Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) approaches were discussed. Finally, we reviewed the concepts of some robust classifiers such as Boosting Ensemble and Support Vector Machines (SVM), and investigated the problems associated with the non-linear SVM classifiers. In the next chapter, we will describe how these modules are tailored to our real-time and embedded age and gender recognition system.

Chapter 4

Video-based Age and Gender Classification on Embedded Systems

Referring to Chapter 2, the majority of the existing state-of-the-art approaches for age and gender recognition are resource-intensive and require high-performance computer systems. However, the emerging applications of video-based demographics classification in mobile services (see Section 2.1) demand a real-time system which is appropriate for resource-limited embedded platforms and mobile devices.

Notably, the few embedded approaches that focused on this problem either are not accurate enough [66], or they are unable to reproduce their performance in outdoor environments with difficult illumination conditions [42]. These facts emphasize the importance of our objectives to propose practical solutions for implementing a video-based age and gender classifier on embedded systems that is able to perform accurately in unconstrained environments.

With this intention, in this chapter we present our novel contributions to the methodology of age and gender recognition for resource-limited systems. As mentioned in Chapter 3, the age and gender recognition is a sub-problem of the facial trait classification problem, and they share several modules as the integral part of the classification pipeline.

However, there are several other modules that should be added to the generic pipeline of Figure 3.0.1 in order to meet the specific requirements of our goal to design an accurate and real-time demographics classifier for unconstrained video that demand minimal

resources to perform. To this end, we have designed the novel architecture of Figure 4.0.1, which integrates various robust mechanisms for face image normalization, dimensionality reduction, and discriminative age recognition based on gender. In general, this architecture is consisted of three parts:

- Training modules that require a high-performance computer system to perform.
- Classification modules that are optimized to perform in real-time on resource-limited embedded systems.
- The modules which are common for both training and testing stages.

This chapter describes the implementation details of the architecture shown in Figure 4.0.1, and explains the advantages associated with this embedded design. Each major component of this block diagram may contain several sub-components that we will address them in relevant sections of this chapter.

First, we start by proposing an improvement in face alignment using the nose in Section 4.1, and a robust illumination normalization strategy in Section 4.2. A review of local patterns and our further optimizations are presented in Section 4.3. Next, a segmental dimensionality reduction method for multi-resolution feature vectors is introduced in Section 4.4 which reduces the computation and memory requirements, remarkably.

We generalize a discriminative demographics classification approach in Section 4.5 to further improve the performance on embedded systems. Finally, the conclusion is presented in Section 4.6. It should be noted that the values of the constants used in our experimental setup are provided in Section 5.2.

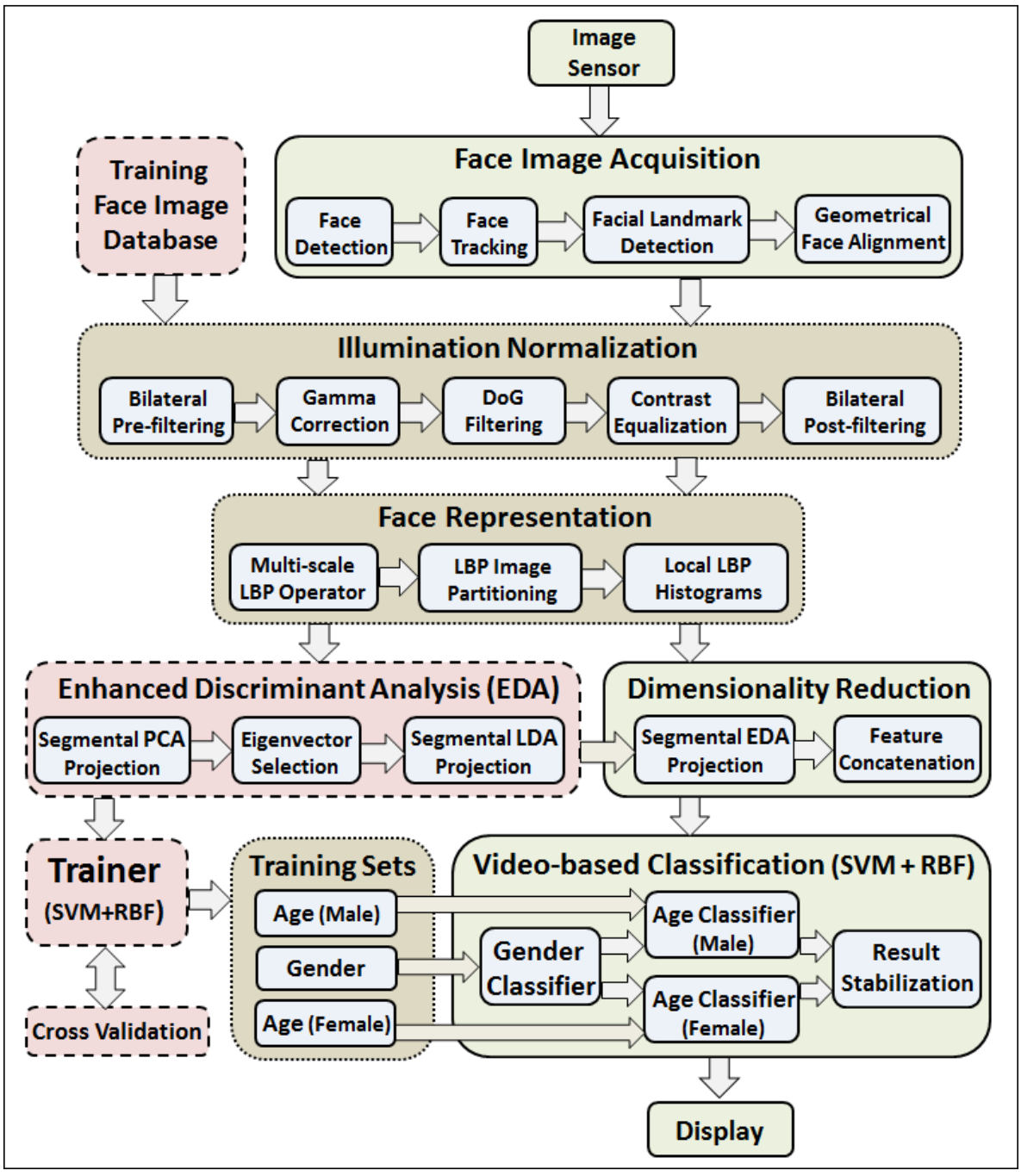


Figure 4.0.1: The block diagram showing the architecture of our video-based age and gender classification system. The training parts are shown in red boxes (dashed-frame), the classification parts in green (solid-frame), and the common modules for training and classification in brown (dotted-frame).

4.1 Face Image Acquisition

Essentially, the face image acquisition stage in Figure 4.0.1 is consisted of four parts: (1) Face Detection, (2) Face Tracking, (3) Facial Landmark Detection, and (4) Face Alignment. In this work, the standard cascaded face detector by Viola and Jones [127] is employed to locate the rectangular regions that contain faces (Figure 4.1.2). As discussed in Section 3.1.1, this method is roughly fast, however, for the large input images of a video-sequence the computation time is increased, proportionally.

To counter this problem, in our real-time system we utilize the "detection-based face tracker" from OpenCV [18]. This tracker detects the faces once, and for subsequent frames it limits the searching area within a neighborhood of the previously detected faces. It offers a timer that searches for the new faces in whole image after a predefined interval. As a result, it avoids searching the whole image for faces in each frame of the video sequence. On the other hand, the changes in head pose and facial expression can lead to displacement of the key features of the face (*i.e.*, eyes, nose, and mouth) which we refer to it as "localization error". Generally, the facial trait classification approaches are sensitive to geometrical misalignment, and the face image should be normalized by aligning the face in order to reduce the localization errors.

A common strategy is to locate the position of the key facial features (landmarks) and regard them as geometric references for performing affine transformations. In other words, the face is transformed into an upright canonical pose by rotating, translating, and scaling the face image. A proper geometric correction strategy can regulate the comparability of query images against the images of training set. A popular approach in face alignment is the positioning of the frontal face images into an upright canonical pose based on the position of eyes [85].

To locate the eyes, we use the open-source *flandmark* library [124] that we introduced in Section 3.2.1. Figure 4.1.1 illustrates some detected facial landmark points on the eyes and nose. The eyes can be aligned horizontally by an in-plane rotation of the face image into an upright pose using the angle θ of Equation 4.1.2. In here, the points $\left(P_{l,x}, P_{l,y}\right)$ and $\left(P_{r,x}, P_{r,y}\right)$ denote the center positions of the left and right eye. Typically, the distance between the eyes d_{eyes} (Equation 4.1.1) is utilized to compute the dimensions of the cropping

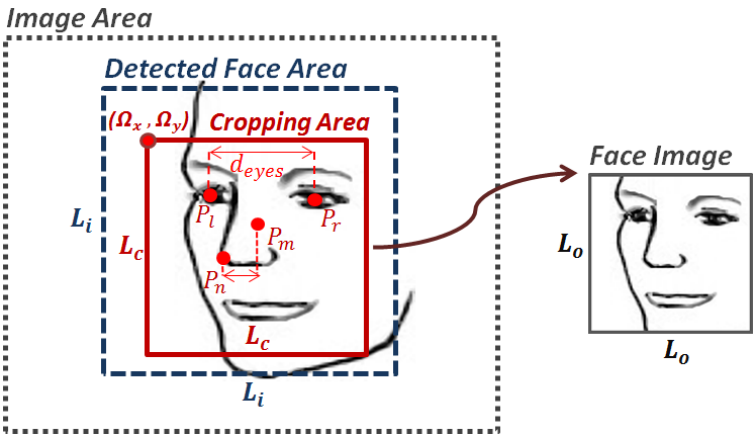


Figure 4.1.1: Facial alignment using the landmarks on nose and eyes. The horizontal distance between P_m and P_n is used to correct the over-scaling problem.

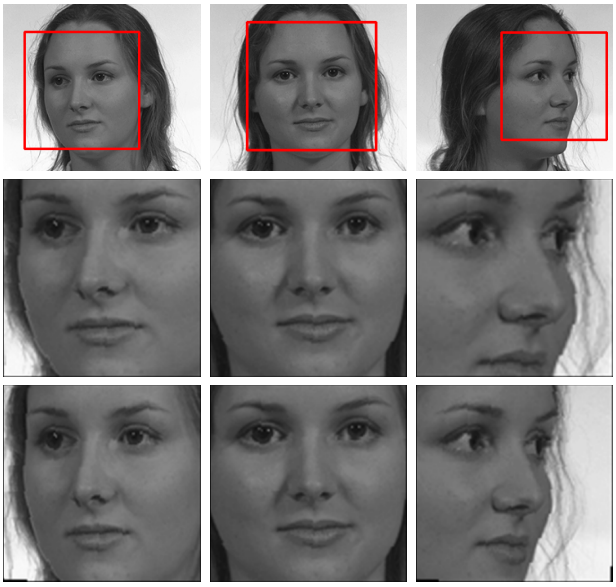
area.

$$d_{eyes} = \sqrt{\left(P_{r,x} - P_{l,x}\right)^2 + \left(P_{r,y} - P_{l,y}\right)^2} \tag{4.1.1}$$

$$\theta = \arctan\left(\frac{P_{r,y} - P_{l,y}}{P_{r,x} - P_{l,x}}\right) \tag{4.1.2}$$

However, in uncontrolled environments as the head's *yaw* angle increases, the eyes distance d_{eyes} shortens. As a result, the dimensions of the cropping area shrink, causing an over-scaling error proportional to the yaw angle and, consequently, the loss of information from the upper and lower parts of the face. Figure 4.1.2 illustrates this problem on three face images posing with different yaw angles. On the other hand, as shown in Figure 4.1.1, the horizontal distance between the points P_n and P_m on the nose increases when the eyes distance d_{eyes} shortens.

Therefore, we propose to use the horizontal positions of the upper nose $P_{m,x}$ and the lower nose $P_{n,x}$ to compensate for the over-scaling in face alignment. These points can be extracted using the facial landmark detector. In Equation 4.1.3, we apply the ratio of these points to find the scale factor S_0, which is used to calculate the offset and the size of cropping area. Indeed, the maximum dimension of the cropping area is limited as a sub-region of the detected face region in order to avoid under-scaling in the case of unreasonably large distance between the points P_n and P_m.



(a) $\angle yaw : -22.5°$ (b) $\angle yaw : 0°$ (c) $\angle yaw : +67.5°$

Figure 4.1.2: Top row: examples [96] of three detected faces posing in three yaw angles. Middle row uses regular face alignment prone to over-scaling error proportional to the head's yaw angle. Bottom row uses our approach to correct the over-scaling problem.

Figure 4.1.1 illustrates this process on a generic face model. In this work, the detected face region is an $L_i \times L_i$ square, and the resulting aligned and cropped face is an $L_o \times L_o$ square image on which the left eye is fixed at the top-left offset Ω_o.

$$S_0 = \left(\frac{d_{eyes}}{L_o - 2\Omega_o} \right) * \max \left(\frac{P_{m,x}}{P_{n,x}}, \frac{P_{n,x}}{P_{m,x}} \right) \tag{4.1.3}$$

From the scale factor S_0, we compute the dimensions $L_c \times L_c$ of the cropping area, its horizontal offset Ω_x, and its vertical offset Ω_y, as follows:

$$L_c = S_0 * L_o \tag{4.1.4}$$

$$\Omega_x = P_{l,x} - S_0 \Omega_o \tag{4.1.5}$$

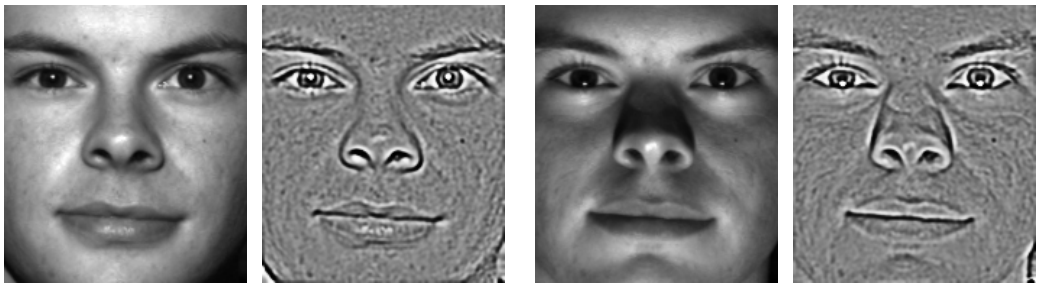
$$\Omega_y = P_{l,y} - S_0 \Omega_o \tag{4.1.6}$$

The description of our experimental setup in Section 5.2 provides the values that we have used in the face acquisition stage.

4.2 Illumination Normalization

In demographics recognition, many researchers have focused only on still face images in controlled environment. However, in real-life video analysis, the facial texture is prone to non-monotonic variations in illumination which impacts the demographics perception. Russell [105] demonstrated the Illusion of Sex on an androgynous face by only increasing the facial contrast, resulting in a feminine look on a male subject. Similarly, in our experiments we have observed the same effect on various *lighting* conditions. For instance, Figure 4.2.1 shows an androgynous male face from the Extended Yale-B database [51], illuminated under two different light source positions.

In Figure 4.2.1(b), the light source is 35° below the horizon inducing non-monotonic gray value transformations by which the observer perceives a feminine look from the male subject. In order to normalize the photometry and reduce the effects of local shadows and highlights, we propose to apply the Preprocessing Sequence (PS) approach [119] on the aligned face image (Section 3.2.2). The results of applying the PS are shown in Figures 4.2.1(a) and 4.2.1(b). In our work, we have used the default values mentioned in Section 3.2.2 for all PS parameters. Nevertheless, a large amount of textural noise is still present. We provide a practical solution to this issue in section 4.3.



(a) PS on Masculine Face $(5°, 10°)$ (b) PS on Feminine Face $(0°, -35°)$

Figure 4.2.1: The effect of illumination on gender perception of a male subject. Original images [51] illuminated from (azimuth, elevation). Masculine look after applying Pre-processing Sequence (PS) [119] on both faces.

4.3 Face Representation

As a matter of fact, the illumination normalization methods help to standardize the photometric characteristics of the face image in different illumination conditions. Nonetheless, the classifier may still suffer from the negative effects of the geometrical displacement of the key features on the face image due to variations in facial pose and expression. Definitely, a proper face alignment (Section 4.1) can alleviate the negative effects, but these variations as well as the morphological facial differences can still deteriorate the classifier's performance. Particularly, this problem is more noticeable in holistic approaches that use pixel intensity values to represent the faces.

A robust candidate to overcome localization errors is the Local Binary Pattern (LBP) operator which has been widely exploited as a means of extracting local features of texture. Basically, for each pixel at a center of a neighborhood, the $LBP_{P,r}$ operator builds a binary sequence by applying the value of the center pixel as a threshold to P pixels in a circular neighborhood of radius r. Figures 4.3.1(a) and 4.3.1(b) demonstrate the result of applying

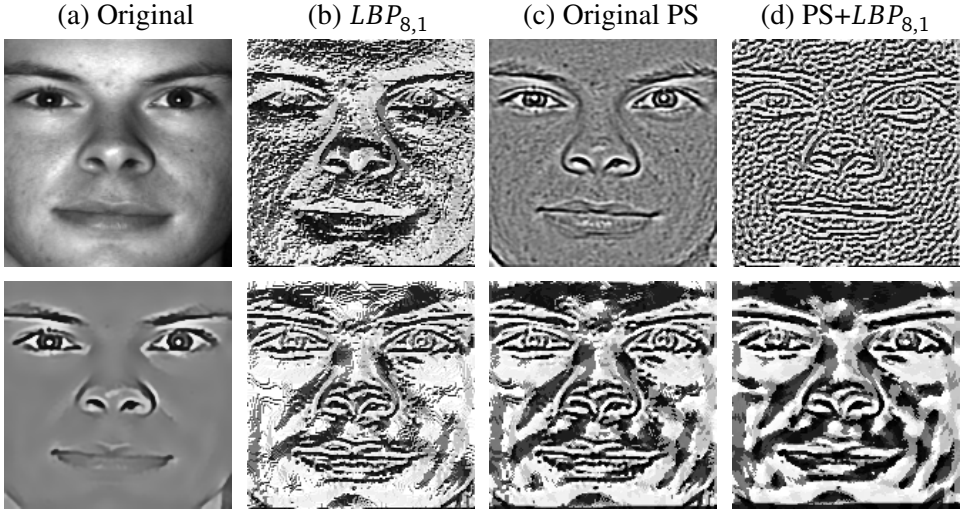


Figure 4.3.1: Applying the LBP operator and the PS illumination normalization on a face image [51]. Filtered PS (FPS) is our approach which filters the LBP noise.

the LBP operator on a face image. Later, the uniform variant of this operator, $LBP^{u2}_{P,R}$, was introduced to capture binary patterns that contain at most two bit-wise transitions from 1 to 0, or 0 to 1. The uniform patterns not only reduce the redundancy, but also can effectively describe the features in corners, edges, spots, and flat areas [92] (see Section 3.3.2).

Furthermore, to reduce the size of the texture descriptor and mitigate the effects of misalignment, the LBP histogram (LBPH) was used to represent the features. Ahonen *et al*. [4] extended this strategy by first dividing the LBP image into J non-overlapping regions $[M_{0,}M_1,\ldots,M_{J-1}]$, then extracting the *local* histograms of regions, and finally concatenating the histograms into a single and spatially enhanced feature vector. Figure 4.3.2 illustrates the feature extraction process from an LBP image.

In essence, LBP operator performs robustly in the presence of monotonic intensity transformations. However, as can be seen in Figures 4.3.1(a) and 4.3.1(b), the thresholding process in LBP is highly sensitive to noise and non-monotonic transformations. A solution is to apply the Pre-processing Sequence (PS) normalization prior to LBP (Section 4.2). Surprisingly, as shown in Figures 4.3.1(c) and 4.3.1(d), the PS only intensified the negative effects of the LBP noise and tuning its default parameters could not improve the results.

Tan *et al*. [119] introduced the Local Ternary Pattern (LTP) operator that employs hysteresis thresholding for noise reduction, and a user-defined threshold to build a ternary pattern (Section 3.3.2). Regardless of the effectiveness of this method, a problem is that the LTP's feature vector has double the size of LBP, and also a proper value for the user-defined threshold is content dependent and cannot be generalized.

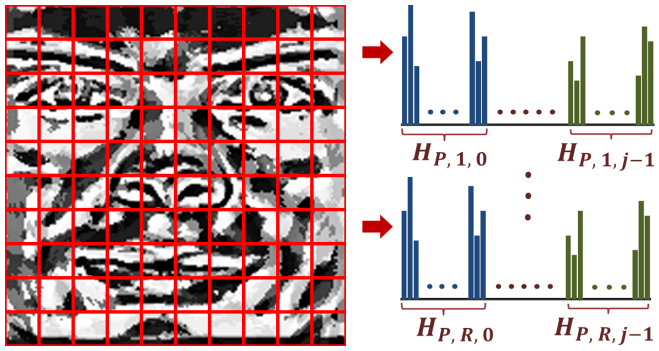


Figure 4.3.2: Extracting multi-scale local histograms

To suppress the noise, we propose to add a *Bilateral filtering* stage to the PS approach. Unlike Gaussian filter, a bilateral filter can effectively suppress the noise while preserving important image features like edges. Also, several fast and embedded-friendly implementations of bilateral filter exist [93]. It is noteworthy that, as advised in [93], we apply the bilateral filtering in two separate iterations: before and after the PS approach.

The illumination normalization stage of the architecture in Figure 4.0.1 shows the order of the applied filters. Filtering the image in Figure 4.3.1(c), we obtain the photometrically enhanced image in Figure 4.3.1(e). As a result, the corresponding LBP images are invariant to variations in illumination and noise. Figures 4.3.1(f), 4.3.1(g), and 4.3.1(h), show the LBP images extracted at three different radii from the Filtered PS image. In addition, Figure 3.2.2 shows the output of our Filtered PS (FPS) approach in comparison to the original PS and other photometric normalization methods.

As a further enhancement, we employ Multi-scale Local Binary Patterns (MSLBP) [92] operator to build a scale-invariant feature vector. In our experiments, it has demonstrated its superior descriptive performance against face localization errors compared to regular LBP. The MSLBP reinforces the face descriptor by combining the histograms from multiple LBP transformations at R different radii in J regions. Figure 4.3.3 illustrates the MSLBP features extraction using four different radii. Equation 4.3.1 defines the uniform LBP histogram of region M_j at radius r and bin $i \in [0, L)$ [21]. Herein, L denotes the total number of bins in uniform LBP histogram. An extra bin has been added for non-uniform feature

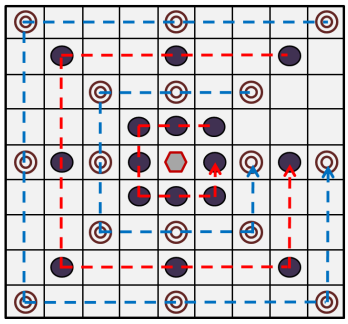


Figure 4.3.3: Four radii of Multi-scale Local Binary Patterns ($R = 4$). The circles represent the position of each surrounding pixel in the circular neighborhood for each radius.

accumulation; therefore, $L = P(P-1)+3$.

$$H^{u2}_{P,r,j}(i) = \sum_{x,y \in M_j} B(LBP^{u2}_{P,r}(x,y) = i) \tag{4.3.1}$$

where $r \in [1,R]$, and $B(u)$ is 1 if $u \geq 0$ and 0 otherwise. Fusing R histograms at each region j, we obtain the raw face descriptor segment $\mathrm{Q}_j \in \mathbb{R}^{1\times(L.R)}$:

$$\mathrm{Q}_{j=}\left[H^{u2}_{P,1,j}, H^{u2}_{P,2,j}, \ldots, H^{u2}_{P,R,j}\right] \tag{4.3.2}$$

$$\mathrm{Q} = [\mathrm{Q}_0, \mathrm{Q}_1, \ldots, \mathrm{Q}_{J-1}] \tag{4.3.3}$$

In this paper, we refer to partitions of the LBP image as *regions*, and partitions of the feature vector as *segments*. The raw feature vector $\mathrm{Q} \in \mathbb{R}^{1\times(J.L.R)}$ is the ensemble of face descriptor segments for each sample, and is meant to feed the classifier's input with multi-resolution LBP features. However, its high dimensionality makes this impractical due to large time and space complexity. This curse of dimensionality also contributes to accuracy degradation due to data redundancy and noise. Inspired by [114, 13], we minimize these problems by applying a *segmental* dimensionality reduction on each descriptor segment Q_j, separately. With respect to face recognition applications, we emphasize three major advantages of using LDA dimensionality reduction on a partitioned feature vector in demographics classification:

1. In holistic models LDA suffers from the curse of dimensionality, and a large dimension reduction prior to LDA can overly discard texture information. In contrast, applying LDA on separate small regions can mitigate its singularity problems while preserving important local texture information.

2. In demographics classification the number of classes is finite, but theoretically, an infinite number of samples can be used to train the classifier. A low dimensional feature vector along with a large number of training samples work best to lift the curse of dimensionality from discriminant analysis.

3. Unlike face recognition, the resource-demanding Eigen-decomposition and PCA+LDA computations are only required in training stage, and not in testing stage. We take advantage of this fact in our real-time embedded application, because we only perform a simple computation for subspace data projection in the testing stage.

4.4 Segmental Dimensionality Reduction

Regardless of the use of image-based or feature-based representation of the face image data, there are several prohibitive problems associated with the dimension of such data, known as the *curse of dimensionality*. Specifically, most of the embedded systems with limited resources cannot afford the large memory and computation requirements of such face representations. On the other hand, the high degree of redundancy and presence of textural noise can drastically reduce the comparability of the face representations and degrade the accuracy of classifier.

Therefore, reducing the dimensionality of the representation vector data, and extracting only the most descriptive and discriminative features from the face image can help to overcome these problems. To this end, we can take advantage of two natural facts about the face [21]. First, the appearance of face from a frontal view is almost symmetrical, and the relative positions of the key features of the face such as eyes, nose, and mouth are constrained. Second, the texture of the facial skin is mostly consistent and there exists a high correlation among the adjacent pixels in different regions of the face image. Thus, we conclude that the face representation can be confined into a discriminative and low-dimensional subspace that can assist to deal with the curse of dimensionality problem.

Referring to Section 3.4, the two well-known methods for dimensionality reduction are the Principal Component Analysis (PCA), and the Linear Discriminant Analysis (LDA). Unlike PCA, the LDA is a supervised reduction method that can linearly separate the classes to capture the most *discriminant* features from the face representation. It aims to maximize the ratio of between-class and within-class separability among N samples of C classes by projecting samples into a new subspace with $C-1$ dimensions (Section 3.4.2). Herein, we have partitioned the feature vector into J smaller segments; therefore, the low dimension of the face descriptor segments Q_j can prevent singularity. Nevertheless, the redundancy and noise in Q_j can still deteriorate the classifier's performance.

In some researches [12], an oval mask is used to eliminate the background noise. However, the eyeglasses, facial pose, facial expression, and the lighting and skin conditions may still influence the results. Hence, prior to LDA, we can wisely make use of PCA along

with a robust feature preservation criterion in order to only retain the most *descriptive* features. PCA is formulated as a maximization problem (see Section 3.4.1), and its segmental projection matrix can be computed as:

$$\mathrm{W}_j^{PCA} = \underset{\mathrm{W}_j}{\operatorname{argmax}} \operatorname{tr}\left(\mathrm{W}_j^T \left(\mathrm{S}_\Sigma\right)_j \mathrm{W}_j\right) \tag{4.4.1}$$

$$\left(\mathrm{S}_\Sigma\right)_j = \sum_{k=1}^{N} \left(\left(Q_k\right)_j - \mu_j\right)\left(\left(Q_k\right)_j - \mu_j\right)^T \tag{4.4.2}$$

where $\left(\mathrm{S}_\Sigma\right)_j$ in Equation 4.4.2 is the total scatter matrix computed from each feature segment $\left(\mathrm{Q}_k\right)_j$ of every k-th sample and j-th region, which are centered using the mean of all N samples $\mu_j \in \mathbb{R}^{1\times(L.R)}$. Our criterion for eigenvector selection in PCA is that the i-th eigenvector can be preserved only if the retained energy e_i (Equation 4.4.3) from the first i eigenvalues λ_m is greater than a threshold τ_e [69].

$$e_i = \frac{\sum_{m=1}^{i} \lambda_m}{\sum_{m=1}^{n} \lambda_m} \tag{4.4.3}$$

This enhancement stage can be considered as an efficient *weighting* mechanism to attain more influence from more discriminative regions of face. Afterwards, the preserved information can be passed for discriminant analysis. In Section 5.4 and Figure 5.3.2 the result of applying this criterion for eigenvector selection is illustrated.

In LDA, we model the segmental between-class and within-class separation of samples with scatter matrices $\left(\mathrm{S}_B\right)_j$ and $\left(\mathrm{S}_W\right)_j$, respectively (see Section 3.4.2). For each segment, the LDA projection matrix W_j^{LDA} can be obtained from maximizing the modified Fisher's criterion [13]:

$$\mathrm{W}_j^{LDA} = \underset{\mathrm{W}_j}{\operatorname{argmax}} \operatorname{tr}\left(\frac{\mathrm{W}_j^T \left(\mathrm{W}_j^{PCA}\right)^T \left(\mathrm{S}_B\right)_j \mathrm{W}_j^{PCA} \mathrm{W}_j}{\mathrm{W}_j^T \left(\mathrm{W}_j^{PCA}\right)^T \left(\mathrm{S}_W\right)_j \mathrm{W}_j^{PCA} \mathrm{W}_j}\right) \tag{4.4.4}$$

where $\left(\mathrm{S}_B\right)_j$ is calculated from the number of samples N_c and the mean μ_j^c of the samples in the class $c \in [1, C]$ (Equation 4.4.5). Also, $\left(\mathrm{S}_W\right)_j$ is computed from the segment $\left(Q_k^c\right)_j$ of every k-th sample of each class c in region j (Equation 4.4.6).

$$(\mathrm{S}_B)_j = \sum_{c=1}^{C} N_c \left(\mu_j^c - \mu_j\right)\left(\mu_j^c - \mu_j\right)^T \tag{4.4.5}$$

$$(\mathrm{S}_W)_j = \sum_{c=1}^{C}\sum_{k=1}^{N_c} \left((Q_k^c)_j - \mu_j^c\right)\left((Q_k^c)_j - \mu_j^c\right)^T \tag{4.4.6}$$

In our method, Q_j is already low-dimensional, and N is large, so the matrix $(\mathrm{S}_W)_j$ will be non-singular. As a consequence, the matrix W_j^{LDA} can be composed from the $(C-1)$ largest eigenvectors u_m of the matrix $(\mathrm{S}_W)_j^{-1}(\mathrm{S}_B)_j$ (Equation 4.4.7).

$$\begin{aligned} &\mathrm{S}_B \mathrm{u}_m = \lambda_m \mathrm{S}_W \mathrm{u}_m \\ &\mathrm{W}^{LDA} = \mathrm{u}_m, \text{ where } m \in [1, C-1] \end{aligned} \tag{4.4.7}$$

An often neglected issue in using LDA for face processing applications is the generalization problem. Although a minimized within-class measure is desirable for matrix $(\mathrm{S}_W)_j$, the within-class samples may be transformed into such a narrow region that the LDA may lose its ability to generalize test data. In addition, the *inverse* of the matrix $(\mathrm{S}_W)_j$ is used to compute the LDA transformation matrix which is *ill-posed* by nature, and is easily prone to numerical instability. In other words, the very small values in matrix $(\mathrm{S}_W)_j$ which may represent noise data can be magnified by the inverse computation $(\mathrm{S}_W)_j{}^{-1}$. To prevent over-fitting and improve the numerical stability, we add a regularization term to the diagonal of the matrix $(\mathrm{S}_W)_j$ using a small positive constant γ and the same-size identity matrix I [94], as follows:

$$(\mathrm{S}_W)_j = (\mathrm{S}_W)_j + \gamma \mathrm{I} \tag{4.4.8}$$

Now, to acquire the most *descriptive* and *discriminant* set of features, each segment $(\mathrm{Q}_k)_j$ of the k-th sample can be projected into our *Enhanced Discriminant Analysis (EDA)* subspace $(\mathrm{F}_k)_j \in \mathbb{R}^{1\times(C-1)}$ using the EDA transformation matrix $\mathrm{W}_j^{EDA} \in \mathbb{R}^{(LR)\times(C-1)}$ (Equation 4.4.9). It is noteworthy that $(\mathrm{Q}_k)_j$ must be normalized to have a zero mean, as Equation 4.4.10 illustrates.

$$\left(\mathrm{W}_j^{EDA}\right)^T = \left(\mathrm{W}_j^{LDA}\right)^T \left(\mathrm{W}_j^{PCA}\right)^T \tag{4.4.9}$$

$$(\mathrm{F}_k)_j = \left(\mathrm{W}_j^{EDA}\right)^T \left((\mathrm{Q}_k)_j - \mu_j\right) \tag{4.4.10}$$

Finally, we concatenate the $(\mathrm{F}_k)_j$ of all N samples into a single feature matrix $\mathrm{F} \in \mathbb{R}^{N\times(J.(C-1))}$ to feed the training stage (Section 4.5). However, prior to concatenation we L2-normalize the rows of matrix F in order to provide the classifier with a coherent descriptor and regularize the similarity quantification among the samples (Equation 4.4.11). Needless to say, each row (F_k) of this matrix represents the EDA projection of the feature vector (Q_k) extracted from the k-th training image for all regions. In testing stage, F only has a single row representing the query image.

$$\mathrm{F} = \begin{bmatrix} (\mathrm{F}_1)_0 & (\mathrm{F}_1)_1 & \cdots & (\mathrm{F}_1)_{J-1} \\ (\mathrm{F}_2)_0 & (\mathrm{F}_2)_1 & \cdots & (\mathrm{F}_2)_{J-1} \\ \vdots & \vdots & \ddots & \vdots \\ (\mathrm{F}_N)_0 & (\mathrm{F}_N)_1 & \cdots & (\mathrm{F}_N)_{J-1} \end{bmatrix} \tag{4.4.11}$$

4.5 Classification on Embedded System

In face-based classification, the objective of classifier is to compare the representation of a probe (or query) face image with those of the training set templates, and determine the category to which the probe image belongs. There exist various classification and similarity measurement techniques in LDA space, such as Euclidean or cosine distance measurement between samples. However, in this work we employ the supervised Support Vector Machine (SVM) classifier [17].

As introduced in Section 3.5.2, SVM is a discriminative binary classifier that finds a maximum-margin hyperplane that has the widest margin to the closest training data points, or the so-called support vectors, of any class. In fact, this hyperplane is a decision function to predict the category to which a new query sample belongs. In our architecture, with have used SVM with an RBF kernel to guarantee an accurate classification in LDA subspace.

Typically, a soft margin SVM with a penalty cost C_p is utilized to compensate for misclassification due to asymmetric class sizes and over-proportional influence of larger classes. We obtain the optimal values for RBF constants γ and C_p using a 10-fold cross-validation method to avoid the under or over-fitting in training stage (see Section 3.5.2). We chose the symbol C_p to not confuse it with the number of classes C.

However, in a multi-class problem ($C > 2$) with disproportionate class sizes, the classifier must be balanced using a dedicated weight for each class. For instance, in Age classification we tune the weight of a smaller data set (*e.g.*, Senior) to counterbalance and diminish the influence of a larger data set (*e.g.*, Adult). After training, the resulting support vectors are of dimension $\mathbb{R}^{1\times(J.(C-1))}$ each, where C in here is the class size. We model the multi-class age classifier as a binary classification problem using one vs. one comparison amongst all classes, and a max-wins voting scheme to determine the age group.

4.5.1 Demography-based Classification

We generalize the work in [55] to improve the performance on embedded system using a demography-based discriminative model for classification. As shown in Figure 4.5.1, we build a tree that discriminates the classification of gender based on ethnicity (n groups), and age (m groups) based on the recognized gender, in sequential stages using n separate classifiers for gender, and $2n$ separate classifiers for age recognition. The rationale behind this method roots in the differences of facial structures among different races and genders. For instance, usually middle-aged females and males do not show the same facial aging signs due to better skin-care in females. Or, different cranial structures or skin colors

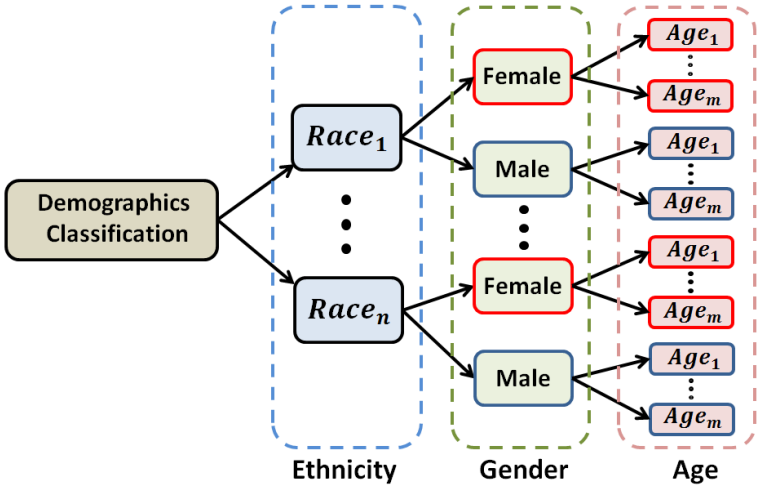


Figure 4.5.1: Demography-based discriminative tree model for classification. Gender is recognized based on ethnicity, and age is estimated based on the recognized gender.

among races may impact the results. Therefore, discrimination based on the parent stage within this tree can effectively improve the success rate in gender or age recognition. More importantly, splitting the training database into separate and smaller sets speeds-up the recognition, significantly. Since in training stage we only include a very limited number of samples per group of training sets (*e.g.*, Asian Females), then much fewer support vectors will be generated for each group. Consequently, the number of similarity measurements (query image vs. training data) and computations will be drastically reduced in testing stage, which is favorable for embedded systems.

It should be noted that, in our current system, the ethnicity recognition stage has not been implemented yet, and will be postponed for the future work. However, the effect of ethnicity on age and gender recognition has been demonstrated in several studies (*e.g.*, in [55]) and, therefore, adding an ethnicity recognition stage to our current system shall improve the generalization capability of our age and gender classifiers for different races.

4.5.2 Video-based Classification

In fact, video-based classification is more challenging than still-image-based techniques, since still-to-still classification in video sequences is an *ill-posed* problem [58]. In this case, regardless of the robustness of the classifiers, the transient variations in head-pose, facial expressions, or improper photometric conditions can cause misclassification in each frame of the video. To stabilize the results, a solution is to employ a majority voting scheme to vote for the best decisions across multiple video frames. We have integrated this *temporal voting* technique in our real-time architecture (Figure 4.0.1) to effectively increase the confidence and reliability of decisions.

Moreover, a face tracker not only improves the face detection performance (Section 4.1), but can accelerate and stabilize the recognition process by continuously preserving the best classification results until the tracked face is lost. As presented in [41], detecting the best quality face images among the frames of a video sequence is another viable strategy to feed the real-time classifiers with only high quality face images, and ignore the non-informative video frames.

4.5.3 Embedded Design Considerations

Given the limitations of the available resources in embedded platforms, a real-time age and gender recognition system requires several design principles to be taken into consideration in order to minimize the memory and computation requirements.

- **Computational Requirements:** Technically, without dimension reduction, linear SVM performs much faster than its RBF counterpart, but it sacrifices the accuracy. On the other hand, SVM+RBF is accurate but is computation-intensive and cannot perform in real-time using a large and high-dimensional training set. For this purpose, our enhanced segmental dimension reduction approach (Section 4.4) is designed to supply the SVM classifier with a low-dimensional enhanced feature vector which is most desirable for the real-time classification on embedded systems. Also, our demography-based classification approach (4.5.1) speeds-up the classification by splitting a large training set into several smaller training sets that are dedicated to specific group of gender or ethnicity. As a consequence, much fewer support vectors are generated for each group and fewer computations are required for classification on the embedded platform. The evaluation results of the computational improvements are discussed in Section 5.5.

- **Memory Requirements:** In essence, linear SVM generates much fewer support vectors than its RBF counterpart due to the linear nature of the maximum-margin hyperplane, but it is not accurate. However, considering that the size of training data is proportional to the number of support vectors, the numerous high-dimensional support vectors generated by SVM+RBF can increase the training data size such that it becomes so large that it cannot fit on an embedded platform. Again, thanks to our enhanced segmental dimension reduction technique the training data size is reduced and, consequently, both volatile and non-volatile memory requirements are reduced. Section 5.6 provides a thorough memory analysis of the designed architecture.

- **Portability of training data and configuration parameters:** Originally, the OpenCV's SVM trainer stores the support vectors in a very large human/machine readable file format (*i.e.*, YAML) that is too bulky to be stored on embedded architectures, and

requires a computation-intensive parser to read the training data. Furthermore, in addition to the required transformation matrices, there are a plethora of parameters that need to be transferred to the embedded platform to configure different modules such as illumination normalization, face alignment, LBP transformation, and classifiers. For these reasons, a self-contained and portable binary file format is designed which includes all the configuration parameters, support vectors, EDA projection matrices W_j^{EDA}, and the mean matrices μ_j for J segments (Section 4.4). For our embedded system we have created a training data file for gender, and two separate training data files for discriminative age recognition based on the subject's gender (Section 4.5.1).

4.6 Conclusion

This chapter has provided a detailed explanation of our novel contributions to the methodology of age and gender recognition for resource-limited systems. We presented a full block diagram of the system's architecture and described different integrated modules of this system. First, an improvement in face alignment was proposed to rectify the over-scaling problems in existing face alignment approaches. Next, the effect of illumination on gender perception was illustrated by an example, and a robust illumination normalization strategy was presented to standardize the photometric characteristics.

We utilized multi-scale local binary patterns to represent the face image, and introduced an enhanced segmental dimensionality reduction technique to extract the most descriptive and discriminative features. To end the chapter, we presented an accurate and resource-efficient classifier along with a demography-based approach to improve the performance on embedded systems. In the next chapter, we will describe our experimental setup for evaluating the performance of our system, and provide the results and discussions to demonstrate the robustness of our methodology.

Chapter 5

Results and Evaluation

Technically, conducting a *fair* comparison between the relevant facial trait classification approaches is very difficult due to diversity of experimental conditions such as the evaluation database size and the quality of test images, and also variations in environmental illumination, facial expression and head pose. In fact, a common protocol is necessary to standardize the experimental conditions and evaluations methods. For this reason, several public face image databases are provided to be used for evaluating the accuracy of the face-based classifiers. On the other hand, for a typical embedded application, a well-defined benchmarking framework is required to measure the computational performance and memory requirements of the embedded implementation.

In this chapter, we present the experimental conditions and the evaluation results of our age and gender classifiers as well as the computational and memory analysis of the classifiers. We begin this chapter by introducing the face image databases that have been used for training and evaluation in Section 5.1. Next, our benchmark setup and classifier parameters are described in Section 5.2, and a series of experiments are presented in Section 5.3 to investigate the effects of these parameters on recognition rate. Then, the evaluation results in terms of accuracy, computational performance, and memory requirements are discussed in Sections 5.4, 5.5, and 5.6, respectively. Lastly, we end this chapter by presenting a conclusion in Section 5.7.

5.1 Databases

Nowadays, many face image databases exist which are each appropriate for certain applications such as face recognition, age estimation, or gender classification. Most of these face image databases are collected from controlled environments in terms of illumination, facial expression, and head pose conditions. Therefore, they are not sufficient for demonstrating the performance and the generalization capability of a classifier in realistic scenarios. A major breakthrough in evaluation methods was the introduction of the so-called "in-the-wild" databases that contain face images collected from unconstrained environments. These uncontrolled databases can help to assess the robustness of classifier against the real-world conditions. In this section, we introduce several well-known face image databases that we have used to train and evaluate our age and gender classifiers.

- **FERET database**[96]**:** The Facial Recognition Technology (FERET) database is one of the earliest and most comprehensive datasets that provide face images labeled with actual age, gender, and ethnicity. These gray-scale images are captured in a controlled environment at a resolution of 256×384, and categorized into several gallery sets for different facial expressions, head poses, and illumination conditions. The two widely-used galleries from this database are called "Fa" (1762 images) and "Fb" (1518 images) which both include frontal pose face images, but with slightly different facial expressions. In this work, we have included 1,267 images from gallery "Fa" into the training set of the age classifier, and used 1,330 images from gallery "Fb" to test the gender classifier.

- **PAL database** [87]**:** The Park Aging Laboratory (PAL) database contains 576 frontal pose face images of 219 male and 357 female subjects. This controlled database is divided into four age groups: 18-28, 30-49, 50-69, and 70-93. We have used 515 images of this database in order to test the gender classifier.

- **BioID database** [68]**:** Originally, published as a controlled face dataset to test and compare the face detection algorithms. It is consisted of 1,521 face images of 23 subjects from which 467 male and 341 female face images are used in our work to evaluate the gender classifier.

- **MORPH database [103]:** A multi-ethnic database consisted of 46,645 male and 8,487 female face images which 77% of them are African-American subjects, 19% White subjects, and the remaining 4% are Asian, Hispanic, and Indian subjects. In our approach, 1,260 images are included in gender training set and 6,573 images in age training set. Although, this database is collected in a controlled environment, the diversity of ethnicity in this database poses a serious challenge for age and gender classification approaches. For instance, if the classifiers are trained using only White subjects, then they may not be able to generalize their performance on the African-American subjects due to differences in facial structures or skin colors. As describe in Section 4.5.1, separating the classifiers based on ethnicity is a practical solution to this problem.

- **Gallagher database [48]:** A very challenging and uncontrolled benchmark composed of 28,231 face images from 5,080 subjects collected from Flickr. Originally, this "in-the-wild" database was created to study group photos of people (*e.g.*, family photos), which most of them are posing for the camera. Therefore, the majority of face images are captured in frontal facial pose under unconstrained illumination and expression conditions. A face detection algorithm was used to locate 86% of the faces in group photos and the rest of faces are located manually. As a common protocol [32], only 14,760 high quality near-frontal face images are used for age and gender classification from which 7380 images are male subjects and 7380 images are female subjects. Moreover, the images are labeled and categorized into seven age groups: 0-2, 3-7, 8-12, 13-19, 20-36, 37-65, and 66+. The majority of the face images in our age and gender training sets were selected from this dataset.

- **Adience database [36]:** Similar to Gallagher dataset, the face images of Adience database are captured in unconstrained environments. There are a total of 26,580 images from 2,284 subjects collected from Flickr which are divided into eight age groups: 0-2, 4-6, 8-13, 15-20, 25-32, 38-43, 48-53, 60+. Specifically, 13,649 face images of this database are captured in near-frontal pose ($\pm 5^\circ$ yaw angle) and in our evaluation method, these images were merely used for testing our age and gender classifiers in order to assess their performances under unconstrained conditions.

5.2 Benchmark Setup

In this work, our embedded benchmarking platform was an Android system running on a multi-core 1.7 GHz Snapdragon 600 (ARMv7) SoC, with 2 GB of RAM and camera resolution 720×1280 pixels. We have implemented our framework in C++, and used Java Native Interface (JNI) to connect with Android system. Notably, several standard routines from OpenCV [18] have been integrated into our framework for face detection, photometric corrections, and SVM training. Also, a self-contained and portable binary file format is designed which includes all the parameters, support vectors, and the segmental projection matrices W_j^{EDA} and μ_j for J segments (Section 4.5.3). The floating-point values have single-precision for support vectors and double-precision for projection matrices. For our embedded system we have created a training data file for gender, and two separate training data files for discriminative age recognition based on the subject's gender (Section 4.5.1). Our age classifier categorizes four age groups of: 0-19, 20-36, 37-65, and 66+ years old.

To evaluate these classifiers, a variety of face databases have been used as a *cross-database* benchmark for training and testing stages. As demonstrated in [12], the single-database evaluations in many researches are optimistically biased due to disproportionate diversity of races and ages, or specific lighting or head pose conditions in each database. Hence, we have trained our classifiers using a combination of selected face images from the databases listed in Table 1, and evaluated the same classifiers on a different set of databases in Table 2. Except the in-the-wild face images of Gallagher [48] and Adience [36] databases, the rest are captured in controlled lighting and head pose conditions.

From Adience database, even though we have used only near-frontal version (13,649 images with ±5° yaw angle), the evaluation on this unconstrained dataset is still very challenging. Eidinger *et al*. [36] demonstrated that the difficulty level of this data set is more than Gallagher dataset. Moreover, unlike some researches [9] that have performed evaluation on manually aligned and normalized images, we have evaluated the classifiers using our full recognition pipeline; from face detection to age and gender recognition. Therefore, our evaluation results closely reflect the real-world conditions.

Table 1: Databases and the number of images used for **training**

Training		Number of images used					
Database	**Image size**	**Gender**		**Age Group (male+female)**			
		Male	**Female**	**0 - 19**	**20 - 36**	**37 - 65**	**66+**
FERET (Fa) [96]	*256×384*	*0*	*0*	*0*	*489+357*	*270+101*	*20+0*
MORPH [103]	*200×240*	*790*	*470*	*1590+800*	*1111+1332*	*850+875*	*15+0*
Gallagher [48]	*Variable*	*7,350*	*7,350*	*1410+1350*	*4000+3911*	*1650+1800*	*307+312*
	Total	***8140***	***7820***	***3000+2150***	***5600+5600***	***2770+2776***	***342+312***

Table 2: Databases and the number of images used for **evaluation**

Evaluation		Number of images used					
Database	**Image Size**	**Gender**		**Age Group (male+female)**			
		Male	**Female**	**0 - 19**	**20 - 36**	**37 - 65**	**66+**
FERET (Fb) [96]	*256×384*	*840*	*490*	*0*	*0*	*0*	*0*
Adience [36]	*Variable*	*3948*	*5060*	*1608+2294*	*1330+1724*	*921+1008*	*56+78*
BioID [68]	*384×286*	*467*	*341*	*0*	*0*	*0*	*0*
PAL [87]	*638×480*	*200*	*315*	*0*	*0*	*0*	*0*
	Total	***5455***	***6206***	***1608+2294***	***1330+1724***	***921+1008***	***56+78***

5.3 Experiments and Discussions

This section provides a detailed explanation of our experiments and the parameters of our age and gender classifiers. Also, several graphs are provided in order to illustrate the effects of different parameters on the recognition rate of gender classifier. The trends of these graphs are similar for gender and age group classifiers, because their underlying methodologies are identical. Considering the same notations used in Chapter 4, we begin by aligning the detected face and cropping it to size $L_o = 100$ pixels with the left eye offset at $\Omega_o = \frac{L_o}{4}$ (Section 4.1). The bar graph in Figure 5.3.8 shows the effectiveness of our correction method for face alignment in controlled and uncontrolled environments. Clearly, the improvement in uncontrolled environments like Adience benchmark is more than controlled environments like FERET database due to higher variations in yaw angles (head), and higher number of evaluation samples in Adience database.

In the next stage, the 100×100 aligned image is photometrically corrected utilizing our Filtered PS method (Sections 4.2). As the samples in Figure 5.3.1 show, this method along with uniform LBP can effectively reduce the effects of illusion of sex (Figure 4.2.1), difference in skin colors, facial cosmetics and lighting conditions while preserving facial wrinkles for age classification. As can be seen in Figures 5.3.3 and 5.3.4, the performance of Retinex method (Section 3.2.2) is superior compared to other methods, but it cannot normalize facial skin colors and its computation time is slightly more than our Filtered P.S method (see Section 5.5). According to Figure 5.3.3, illumination normalization in controlled environments is not required and even can degrade the recognition rate.

In order to compensate for face localization errors, each feature segment Q_j is composed of five different radii ($R = 5$) of uniform ($L = 59$, if $P = 8$) multi-scale LBP histograms (Section 4.3). However, as Figure 5.3.5 shows, the accuracy of six radii configuration is slightly better; therefore, depending on the availability of resources on embedded platform, our classifiers can be easily configured to use six radii of LBP (larger training data file). According to our experiments, greater radii ($R > 6$) in uniform LBP could not improve the results further.

Similar to Figure 4.3.2, the resulting LBP images are partitioned into 10×10 non-overlapping regions ($J = 100$) to extract the feature vector $Q \in \mathbb{R}^{100\times(59*5)}$ for each sample.

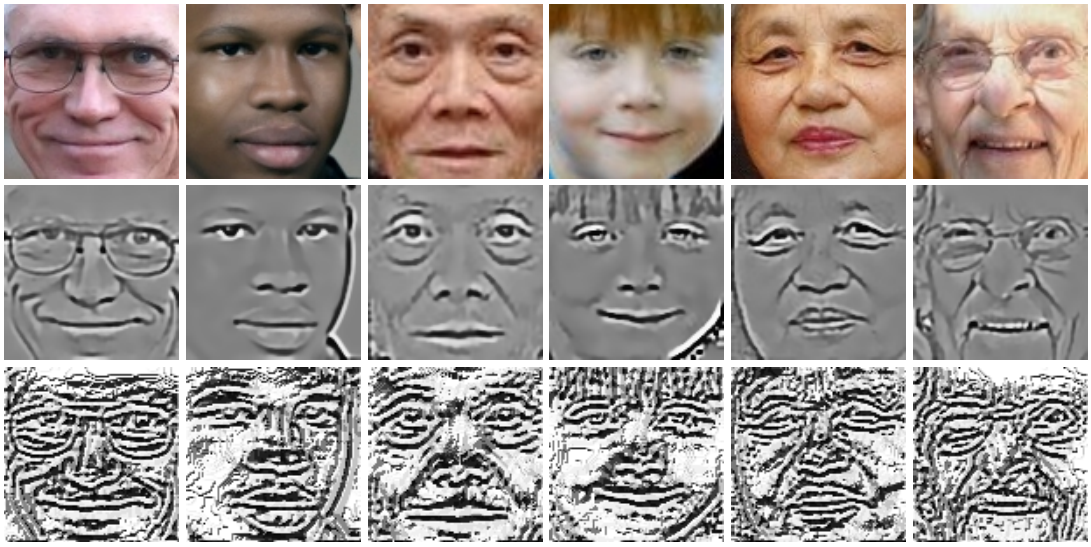


Figure 5.3.1: Our illumination normalization approach. Original images (top), Filtered PS (middle), and corresponding $LBP_{8,1}$ images (bottom).

It is worth noting that the size of the regions should not be necessarily the same; for instance, on a 100×100 image, a 12×12 overlapping configuration or any other number of regions with different sizes can be configured in our system. Figure 5.3.6, illustrates how the recognition rate varies as a function of increasing number of regions.

For eigenvector selection in our segmental dimensionality reduction approach (Section 4.4), we have obtained the energy threshold values τ_e (Table 3), experimentally. The color maps in Figure 5.3.2, illustrate the percentage of retained eigenvectors in each segment Q_j for age and gender classifiers. Matching the regions in the color maps and the LBP image of Figure 4.3.2, the importance of discriminative regions around the eyes and mouth is evident. Thereby, the effects of eyeglasses and facial expressions can be minimized. Moreover, as Figure 5.3.7 demonstrates, the success rates of the classifiers are sensitive to the value of threshold τ_e. As a rule of thumb, the fewer training samples are used, the fewer eigenvectors must be retained in order to prevent singularity or overfitting problems in LDA subspace.

To further improve the numerical stability in discriminant analysis we chose the regularization constant $\gamma = 0.01$ to avoid near-zero eigenvalues (Section 4.4). Tuning the constant

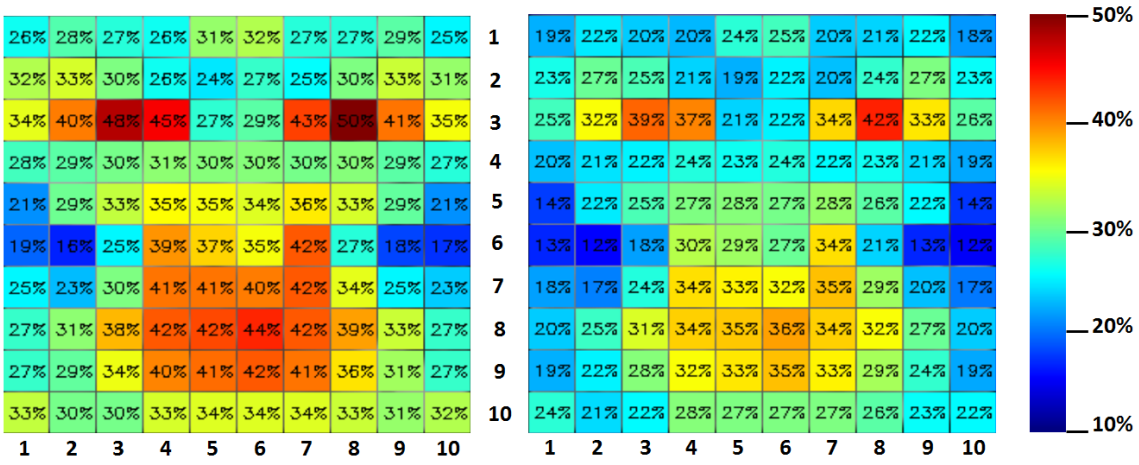


Figure 5.3.2: Color maps showing the percentage of retained energy from PCA in each region for gender (left; $\tau_e = 0.98$) and age (right; $\tau_e = 0.97$). Notice the high variance regions around the eyes and mouth.

γ with other values did not affect the results significantly. Table 3 lists the configuration of our classifiers such as the total number of training images, values for the threshold τ_e, and the RBF parameters. To balance the age training set, the class weights were adjusted experimentally, based on the size of each class and their influence on other classes (see Section 4.5).

Table 3: Configuration of the age and gender classifiers

Classifier	**#Classes**	**#Training Images**	**PCA**	**RBF**	
			τ_e	γ	C_p
Gender	*2*	*15,960*	*0.98*	*1.0125*	*2.5*
Age (M)	*4*	*11,712*	*0.97*	*1.0125*	*2.5*
Age (F)	*4*	*10,838*	*0.97*	*1.5187*	*2.5*

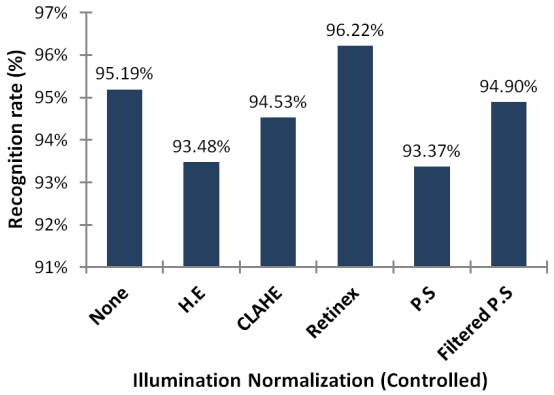


Figure 5.3.3: Effect of illumination normalization in controlled environments (FERET).

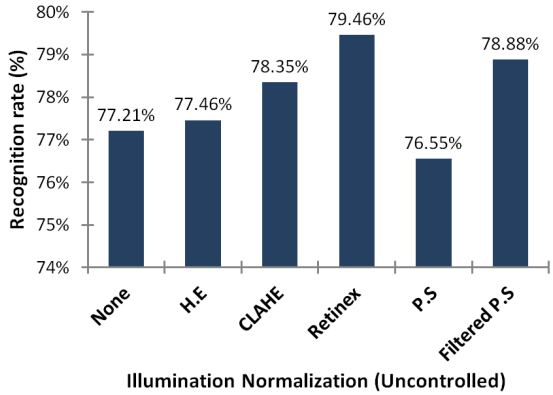


Figure 5.3.4: Effect of illumination normalization in uncontrolled environments (Adience).

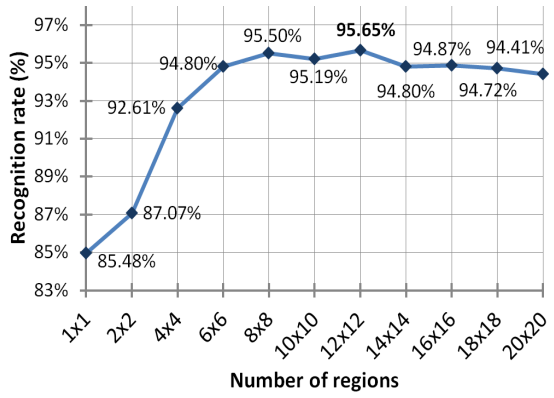


Figure 5.3.5: The recognition rates per different number of regions.

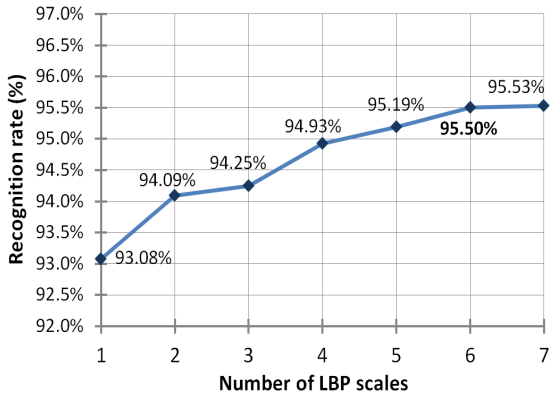


Figure 5.3.6: The recognition rates per different number of concatenated LBP scales.

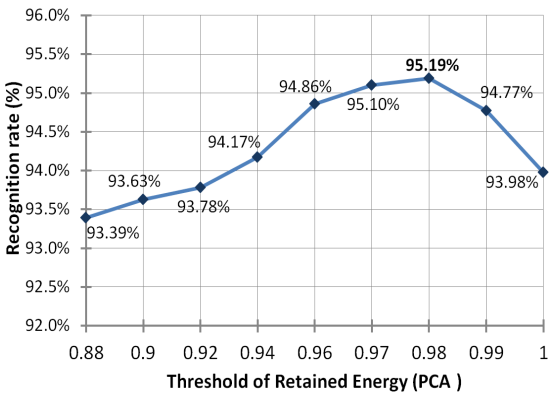


Figure 5.3.7: The recognition rates per different threshold values τ_e for retaining eigenvectors (PCA).

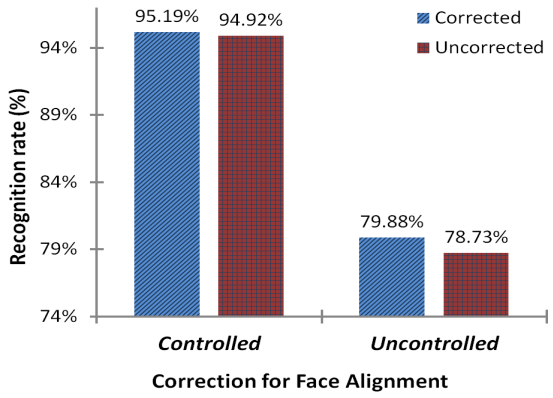


Figure 5.3.8: Effect of our correction method for face alignment in controlled and uncontrolled environments.

5.4 Accuracy Analysis

In this section, we present the evaluation results of our age and gender classifiers, and compare their accuracy against several state-of-the-art methods. In spite of the memory-efficient and real-time performance of our method, the success rates are closely comparable with other state of the art but resource-demanding approaches. Although the classification parameters can be tuned to achieve a high success rate for a specific database, it may fail to generalize the success on other databases.

Some of such non-generic parameterization include: retaining eigenvectors selectively per database [12], existence of multiple same identity subjects in evaluation [88], or targeted and very low number of evaluation samples [67]. In contrast, we aim to evaluate our classifiers with the same configurations on every database. It should be noted that, unlike many other studies, we have provided the recognition rates for male and female groups

Table 4: Gender recognition rates (our *MSLBP+EDA+SVM* method vs. the state-of-the-art classifiers). ***Note:*** only the total success rate is available for the cited papers (**#**: No. of images used). See Table 2 for no. images we used for evaluation.

Database	**Classifier** (*:embedded system)	**Female**	**Male**	**Total**
BioID	***MSLBP+EDA+SVM****	***92.08%***	***98.50%***	***95.79%***
	*SHORE[42]**	*N/A*		*94.3%*
FERET	***MSLBP+EDA+SVM****	***96.12%***	***94.64%***	***95.19%***
	LUT Adaboost[85]	*#450*	*#450*	*93.33%*
	SVM+RBF[12]	*#403*	*#591*	*93.95%*
PAL	***MSLBP+EDA+SVM****	***91.43%***	***90.50%***	***91.07%***
	Adaboost[9, 12]	*#357*	*#219*	*87.24%*
	SVM+RBF[12]	*#357*	*#219*	*89.81%*
Adience	***MSLBP+EDA+SVM****	***90.77%***	***65.93%***	***79.88%***
	Dropout-SVM[36]	*#6455*	*#5824*	*75.8%*

separately in order to present a detailed analysis of accuracy. Also, for the sake of fair comparison, the total recognition rates along with the number of evaluation images in the cited papers are given in order to compare with the number of images that we have used for evaluation (Table 2).

Table 4 shows the recognition rates obtained from our cross-database evaluations for gender classification on the databases mentioned in Table 2, and the comparisons to some existing robust classifiers. According to our observations, the reason for lower gender recognition rate in male group (*65.93%*) of Adience database can be attributed to the existence of numerous children of under 6 years old who are very similar in appearance to females.

Also, the low gender recognition rate on PAL database, confirms the influence of ethnicity on demographics classification. Provided that our training set is mostly consisted of White subjects (Table 1), the gender (or age) classifier may fail for some African subjects in this database due to different facial structures and features. Likewise, the same conditions may apply for other missing races in the training set. Exploiting the demographics discriminative classification strategy (Section 4.5.1), the classifier can better generalize on faces of different races.

As Table 5 shows, our evaluation results for age classification on Adience dataset outperform the results of the state-of-the-art dropout-SVM method of Eidinger *et al.* [36]. The improved accuracy can be attributed to the utilization of our filtered PS illumination normalization technique, and a multi-scale representation of face images. Nevertheless, in

Table 5: Age recognition rates per age group and gender (our *MSLBP+EDA+SVM* method vs. the state-of-the-art classifier) using the **Adience** uncontrolled benchmark. ***Note:*** only the total success rate is available for the cited paper (**#**: No. of images used). See Table 2 for no. images we used for evaluation.

Age group / **Classifier**	**0 - 19**		**20 - 36**		**37 - 65**		**66+**		**Total**	
	F	**M**	**F**	**M**	**F**	**M**	**F**	**M**	**F**	**M**
MSLBP+EDA+SVM	***82.7%***	***93.0%***	***85.5%***	***83.5%***	***75.8%***	***75.3%***	***80.47%***	***83.6%***	***82.2%***	***85.4%***
Dropout-SVM[36]	*#2989*	*#2487*	*#1692*	*#1602*	*#1027*	*#1148*	*#309*	*#272*	*80.7%*	

Adience database the existence of numerous faces with masks, makeup, occlusions, and severe distortions, increases the classification errors, considerably.

Particularly, in contrast to males of this data set, many *15-19* years old females are misclassified due to high resemblance to the *20-36* age group. We believe the lower intensity of facial aging signs due to cosmetics and skin-care in females may contribute to these errors. Also, the senior age group in Gallagher (training database) starts from *66* years old, but in Adience (evaluation database) from *60* years old. This discrepancy and confusion could be the reason for the lower success rates in our *37-65* and *66+* age groups.

5.4.1 Limitations for Single-database Evaluation

A common protocol to demonstrate the accuracy of age classifiers is to provide a 7-classes "Confusion Matrix" using an uncontrolled database such as Gallagher [48, 112, 36, 32]. For such a single-database evaluation, first the database is divided into N folds, then the classifier is evaluated N times by training with $N-1$ folds and testing with the remaining fold, and finally the accuracy is represented by the mean of all N evaluations. However, from a technical standpoint this evaluation method is at odds with our segmental dimensionality reduction technique and, therefore, a confusion matrix cannot be created.

The reason roots in the oversensitivity of LDA to the curse of dimensionality in face representation (see Section 3.4.2), and the highly unbalanced nature of Gallagher and Adience databases. For instance, the Gallagher database contains only *417* face images for 8-12 years old, compared to *7,921* face images for 20-36 years old. Similar to the protocol used in [32, 112, 36], if we divide this database into 5 folds, only 83 images will be available for training the age group of 8-12.

This small number of training samples along with a high dimensional feature vector (*e.g.*, 295 for five scales of LBP in each region) is often a recipe for overfitting and singularity problems in LDA subspace. Even utilizing PCA before LDA to reduce the dimensionality cannot solve this *ill-conditioned* problem, because the available samples are too few and PCA is a "lossy" compression method that overly discards useful texture information and deteriorates the recognition rate, significantly.

As a matter of fact, our main objective in this thesis was to design and implement a

real-time and accurate (see Tables 5 and 4) age and gender classifier for embedded systems which could be achieved by collecting an adequate number of training samples.

5.5 Computational Analysis

In this section, using the same embedded benchmark setup of Section 5.2, we analyze the computational requirements for different stages of our age and gender classification system. As can be seen in Table 6, most of the computation time for face alignment stage is spent on landmark detection. In our system, the detection-based face tracker [18] (see Section 4.1) runs on a separate thread (using POSIX library) and we do not take its computation time into account. This face tracker searches the *whole* image only at specific intervals, and otherwise limits the searching scope to the neighborhood of the previously detected faces in each frame of the video. Therefore, it performs very fast, and also its performance is less dependent on the dimensions of the input image.

In the next stage, our system provides five different options to perform illumination normalization on face image which we discussed in Section 5.3. Table 6 shows that our Filtered P.S method is slightly faster than Retinex. In general, bilateral filters are computation-intensive, but there exist several fast approximation algorithms [93] for bilateral filtering that can perform in real-time.

Table 6: Computational analysis for preprocessing stage

Face Alignment		**Illumination Normalization**				
flandmark	**Alignment**	**H.E**	**CLAHE**	**Retinex**	**P.S**	**Filtered P.S**
21.7 ms	*5.1 ms*	*0.6 ms*	*1.7 ms*	*17.3 ms*	*7.1 ms*	*15.5 ms*

Table 7: Computational analysis for classification stage

Classifier \ **Stage**	**EDA Projection**	**SVM Classification**
Gender	*5.9 ms*	*2.3 ms*
Age	*11.1 ms*	*3.5 ms*

Moreover, by applying a discriminative classification strategy on a low dimensional enhanced feature vector, the computation time of classifiers can be minimized. As discussed in Section 4.5.1, the separation of classifiers for different races and gender types, not only improves the accuracy, but also accelerates the recognition process due to fewer number of input training samples and support vectors. Table 7 shows the required computation times for subspace projection and SVM classification. Although an RBF kernel is employed for SVM classification, it performs very fast thanks to our segmental Enhanced Discriminant Analysis (EDA) dimensionality reduction technique that we discussed in Section 4.4.

On the embedded platform described in Section 5.2, our experiments demonstrate a total performance of *15* to *20* frames per second depending on the input frame rate, on-screen display parameters, illumination normalization technique used and, more importantly, the status of face tracker. In the latter case, the last recognition results are preserved for the tracked face, and it is not required to re-perform the classification until the tracking is lost.

5.6 Memory Analysis

In our system, in terms of space complexity, both volatile and non-volatile memory requirements are minimized. Originally, the OpenCV's SVM trainer stores the support vectors in a very large human/machine readable file format (*i.e.* YAML) that is too bulky to be stored on embedded architectures. As Table 8 shows, our self-contained file format along with low dimensional training data is appropriate for most embedded architectures due to its high compression ratio of up to *99.5%* (see Section 4.5.3).

Normally, without dimensionality reduction (*i.e.*, compression) a regular multi-scale LBP face representation with an SVM+RBF classifier would need a training data (single-precision floating-point) of dimension $\mathbb{R}^{V\times(R.L.J)}$, where V denotes the number of support vectors. However, utilizing our enhance dimensionality reduction technique, the dimension is reduced to $\mathbb{R}^{V\times(J.(C-1))}$ along with a small overhead to store the EDA transformation matrix $\mathrm{W}_j^{EDA} \in \mathbb{R}^{(LR)\times(C-1)}$ and the mean of all samples $\mu_j \in \mathbb{R}^{1\times(L.R)}$ for J segments (Section 4.4). Based on these dimensions, we formulate the uncompressed training data size s_u

Table 8: Memory Requirements: Regular *MSLBP+SVM+RBF* vs. Our compressed file format

Classifier	#Support Vectors	MSLBP+ SVM+RBF	Our Portable File Format	Compression Ratio
Gender	*5,978*	*~672 MB*	*~2.8 MB*	***99.5%***
Age (M)	*8,085*	*~909 MB*	*~10.3 MB*	***98.8%***
Age (F)	*8,311*	*~935 MB*	*~10.5 MB*	***98.8%***

(Equations 5.6.1) and the compressed training data size s_c (Equation 5.6.2).

$$s_u = V \times R \times L \times J \times E \tag{5.6.1}$$

$$s_c = (V\,(C-1) + 2LR\,(C-1) + 2LR)\,(JE) \tag{5.6.2}$$

where E denotes the number of bytes for the floating-point type. For example, for gender classifier in Table 8, if $V = 5978$ support vectors, $C = 2$ classes, $R = 5$ radii of multi-scale LBP, $L = 59$ LBP histogram bins, $J = 100$ regions, and $E = 4$ bytes floating-point, then the training data of size:

$$s_u = 5978 \times 5 \times 59 \times 100 \times 4 = \frac{705,404,000 \text{ bytes}}{1024 \times 1024} =\sim 672 \text{ MB}$$

is compressed to size (including a small meta-data size in final approximation):

$$s_c = (5978 \times 1 + 2 \times 59 \times 5 \times 1 + 2 \times 59 \times 5)(100 \times 4) = \frac{2,863,200 \text{ bytes}}{1024 \times 1024} =\sim 2.8 \text{ MB}$$

Although we have fewer samples for age classifier, its training data (*i.e.*, support vectors) is larger than gender classifier due to higher number of classes C and larger value for RBF parameter γ that generates more support vectors, proportionally.

5.7 Conclusion

In this chapter, the evaluation results of our age and gender classifiers on an embedded benchmarking platform were presented. In order to demonstrate the robustness and performance of these classifiers a thorough analysis in terms of accuracy, computation, and

memory was conducted. Also, we introduced several databases and described the benchmark setup and the rationale behind choosing different parameters of the classifiers. The impact of these parameters on the final recognition rates were illustrated by providing several graphs.

Our accuracy analysis clearly demonstrates the robustness and the generalization capability of our classifiers in unconstrained environments with difficult illumination conditions. Also, the complete computational and memory analysis that were presented in this chapter could acknowledge the very low computational and memory requirements of our approach.

Chapter 6

Conclusions and Future Work

In this chapter, we conclude this thesis with a summary of the material presented in the previous chapters, and discuss a number of limitations and potential solutions as our future work to extend the capabilities of our age and gender classification system.

6.1 Conclusions

This thesis has presented a complete framework for real-time and accurate age and gender classification on embedded systems in unconstrained environments. We began this thesis by introducing the challenges of age and gender classification on resource-limited systems in Chapter 1. Next, we presented a chronological overview of the robust and state-of-the-art approaches in the realm of gender classification and age estimation, and their potential applications in Chapter 2.

In Chapter 3, we introduced the relevant theories and the common components of all facial trait classification systems which were the prerequisites to prepare for describing our contributions, and the embedded implementation of our classifiers. We described and compared different photometric and illumination normalization techniques such as Histogram Equalization (HE), Contrast Limited Adaptive Histogram Equalization (CLAHE), Retinex, and Preprocessing Sequence (PS). Also, we explored the variants of Local Binary Patterns (LBP) such as Uniform and Rotation Invariant LBP, and the Local Ternary Pattern (LTP) which could overcome the noise sensitivity problem of LBP for representing the face

image. In order to reduce the dimensionality of face image representation, we described the Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) approaches, and also the drawbacks of LDA such as singularity problem in high-dimensional space were discussed. Next, we introduced the theory of Support Vector Machine (SVM) classifier, and investigated several problems such as the large resource requirements of the non-linear kernels, the overfitting, and the underfitting in training stage.

In Chapter 4, we presented the details of our contributions to the methodology of video-based age and gender classification for embedded systems, and in Chapter 5 we evaluated the accuracy of our system and analyzed memory and computational requirements. Our first contribution was an improvement in face alignment using two positions on the nose to rectify the over-scaling problem in existing approaches that use the distance between the eyes to find the cropping area of the face image. This approach could increase the success rate of evaluation up to 1.5% depending on the head's yaw angle.

Next, the effect of illumination on gender perception was illustrated by an example, and an enhanced illumination normalization strategy using the bilateral filtering and Preprocessing Sequence (PS) method was proposed to standardize the photometric characteristics of the face image. In our experiments, although the Retinex approach was slightly superior (~1%), but our approach performed faster on embedded systems and could normalize the skin color as well. For a robust face representation, we utilized a multi-scale variant of LBP operator in order to reduce the localization errors caused by variations in facial expression or head pose. Concatenating five scales of LBP, our experiments clearly demonstrated the effectiveness of this approach that increase the recognition rate up to 2.5%, compared to single-scale LBP operator.

However, this approach added a large amount of redundancy to the feature vector, and increased its dimensionality, significantly. To counter this problem, we introduced an enhanced segmental dimensionality reduction technique utilizing Enhanced Discriminant Analysis (EDA) to extract the most descriptive and discriminative features from the face representation. This technique not only improved the accuracy by reducing the noise and redundancy, but also enabled the implementation of our classifier on the resource-limited embedded systems. Thanks to this strategy we could achieve 99.5% compression ratio in the size of age and gender training sets. As another advantage, we were able to employ

a *non-linear* SVM classifier with RBF kernel in order to perform an accurate classification in the EDA subspace which otherwise was not possible without our dimensionality reduction approach. Furthermore, our generalization of demography-based gender and age classification was another major contribution to minimize the computation time of the classification by decreasing the number of generated support vectors. For this purpose, we have trained two separate age classifiers for male and female subjects, and in testing stage the recognized gender determines which age model to use.

In spite of the memory-efficient and real-time performance of our method, the recognition rates are closely comparable with other state-of-the-art but resource-demanding approaches. For instance, we have improved the age recognition rate up to 3% compared to the Drop-out SVM classification approach by performing a cross-database evaluation on the uncontrolled Adience dataset. This improvement can be attributed to our illumination normalization technique, multi-scale face image representation, and demography-based age classification using SVM classifier with non-linear RBF kernel.

For gender classification, we could achieve 95.79% recognition rate on BioID database which is 1.5% better than the embedded SHORE project [42]. We have provided a detailed accuracy analysis in Section 5.4. In addition, our computational analysis in Section 5.5 demonstrated the real-time performance of our resource-efficient classifiers at a frame rate of 15-20 fps on an Android embedded platform. The low memory and computation requirements of our methodology, makes it a viable choice for real-time pattern recognition in embedded vision applications.

6.2 Limitations and Future Work

As discussed in Section 5.4.1, perhaps the most limiting factor in all systems based on the Linear Discriminant Analysis (LDA), including our system, is the inadequacy of training data and the curse of dimensionality [14, 94]. Essentially, for a given number of training samples, there is a limit for the maximum number of features, or the so-called dimensionality of the feature vector. With the growth of dimensionality (*i.e.*, volume of the space), the number of required training samples must grow exponentially, otherwise, the data in high-dimensional space will become sparse.

This sparsity of data can lead to overfitting since there are not enough training samples so that the LDA can learn to generalize well to predict the unforeseen samples (*i.e.*, the query face images in our system). Also, utilizing PCA before LDA to reduce the dimensionality cannot solve this ill-conditioned problem, because the available samples are too few and PCA is a "lossy" compression method that overly discards useful texture information and deteriorates the recognition rate, significantly.

For our future work, we will test different enhanced variants of Discriminant Analysis (DA) such as Regularized DA, Null-space DA, and Orthogonal DA, and will employ a variant that is robust to singularity problems in high-dimensional feature space.

Another limitation of our work, and many demographics classification approaches, is the effect of ethnicity on age and gender classification due to differences in facial structure and skin color in different races [84, 55]. According to our experiments, when we mixed African face images into a training set consisting of mostly White subjects, the recognition rate of our classifiers decreased, moderately (even by evaluating only on White subjects).

As mentioned in Section 4.5.1, in addition to ethnicity, the gender of a subject can also affect the age classification. In this work, we have separated the age classifiers for male and female subjects in order to overcome the effect of gender on age group classification.

Nevertheless, the remaining limitation in our system is the lack of an ethnicity classifier that enables us to separate the age and gender classifiers based on the subject's ethnicity (see Figure 4.5.1). Such an ethnicity classifier is expected to bring three advantages: (1) obviously, it can complete our demographics classification system by reporting the ethnicity as well, (2) our age and gender classifiers can generalize their prediction capabilities to non-white races, (3) the computational performance of age and gender classifiers can be further increased by separating them based on ethnicity which generates much fewer support vectors for each group.

In future, we will design an ethnicity classifier that can complete our system. We plan to utilize the same methodology of this thesis, but utilize a "color" type of LBP in order to capture the skin color, and add it to our existing multi-scale LBP face representation. We expect this enhancement to significantly increase the accuracy and the computational performance of our framework.

References

[1] The FG-NET Aging Database. http://sting.cycollege.ac.cy/alanitis/fgnetaging/.

[2] H. Abdi, D. Valentin, B. Edelman, and A. J. O'Toole. More about the difference between men and women: evidence from linear neural network and the principal-component approach. *PERCEPTION-LONDON-*, 24:539–539, 1995.

[3] J. Aghajanian, J. Warrell, S. J. Prince, P. Li, J. L. Rohn, and B. Baum. Patch-based within-object classification. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1125–1132. IEEE, 2009.

[4] T. Ahonen, A. Hadid, and M. Pietikainen. Face description with local binary patterns: Application to face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(12):2037–2041, Dec 2006.

[5] L. A. Alexandre. Gender recognition: A multiscale decision fusion approach. *Pattern Recognition Letters*, 31(11):1422–1427, 2010.

[6] T. R. Alley. *Social and applied aspects of perceiving faces*. Psychology Press, 2013.

[7] F. Alnajar, C. Shan, T. Gevers, and J.-M. Geusebroek. Learning-based encoding with soft assignment for age estimation under unconstrained imaging conditions. *Image and Vision Computing*, 30(12):946–953, 2012.

[8] R. Azarmehr, R. Laganiere, W.-S. Lee, C. Xu, and D. Laroche. Real-time embedded age and gender classification in unconstrained video. In *IEEE Computer Society*

Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), June 2015.

[9] S. Baluja and H. A. Rowley. Boosting sex identification performance. *International Journal of Computer Vision*, 71(1):111–119, Jun 2006.

[10] A. Bansal, R. Agarwal, and R. Sharma. SVM based gender classification using iris images. *2012 Fourth International Conference on Computational Intelligence and Communication Networks*, Nov 2012.

[11] J. Y. Baudouin and G. Tiberghien. Gender is a dimension of face recognition. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 28(2):362–365, 2002.

[12] J. Bekios-Calfa, J. M. Buenaposada, and L. Baumela. Revisiting linear discriminant techniques in gender recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(4):858–864, Apr 2011.

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

[14] R. Bellman, R. E. Bellman, R. E. Bellman, and R. E. Bellman. *Adaptive control processes: a guided tour*, volume 4. Princeton university press Princeton, 1961.

[15] A. Benoit, A. Caplier, B. Durette, and J. Hérault. Using human visual system modeling for bio-inspired low level image processing. *Computer vision and Image understanding*, 114(7):758–773, 2010.

[16] O. Bilaniuk, E. Fazl-Ersi, R. Laganière, C. Xu, D. Laroche, and C. Moulder. Fast lbp face detection on low-power simd architectures. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, CVPRW '14, pages 630–636, Washington, DC, USA, 2014. IEEE Computer Society.

[17] B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. *Proceedings of the fifth annual workshop on Computational learning theory - COLT 92*, 1992.

[18] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[19] Z. Cao, Q. Yin, X. Tang, and J. Sun. Face recognition with learning-based descriptor. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2707–2714. IEEE, 2010.

[20] E. B. Ceyhan, S. Sagroglu, S. Tatoglu, and E. Atagun. Age estimation from fingerprints: Examination of the population in Turkey. *2014 13th International Conference on Machine Learning and Applications*, Dec 2014.

[21] C.-H. Chan, J. Kittler, and K. Messer. Multi-scale local binary pattern histograms for face recognition. *Lecture Notes in Computer Science*, pages 809–818, 2007.

[22] K.-Y. Chang, C.-S. Chen, and Y.-P. Hung. A ranking approach for human ages estimation based on face images. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 3396–3399. IEEE, 2010.

[23] K.-Y. Chang, C.-S. Chen, and Y.-P. Hung. Ordinal hyperplanes ranker with cost sensitivities for age estimation. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 585–592. IEEE, 2011.

[24] K. Chen, S. Gong, T. Xiang, and C. C. Loy. Cumulative attribute space for age and crowd density estimation. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 2467–2474. IEEE, 2013.

[25] H. Cheng, Z. Qin, W. Qian, and W. Liu. Conditional mutual information based feature selection. In *Knowledge Acquisition and Modeling, 2008. KAM'08. International Symposium on*, pages 103–107. IEEE, 2008.

[26] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. *IEEE Transactions on pattern analysis and machine intelligence*, 23(6):681–685, 2001.

[27] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[28] N. Costen, M. Brown, and S. Akamatsu. Sparse models for gender classification. In *Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference on*, pages 201–206. IEEE, 2004.

[29] G. W. Cottrell and J. Metcalfe. Empath: Face, emotion, and gender recognition using holons. In *Proceedings of the 1990 Conference on Advances in Neural Information Processing Systems 3*, NIPS-3, pages 564–571, San Francisco, CA, USA, 1990. Morgan Kaufmann Publishers Inc.

[30] D. Crandall, P. Felzenszwalb, and D. Huttenlocher. Spatial priors for part-based recognition using statistical models. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 10–17. IEEE, 2005.

[31] N. Cristianini and J. Shawe-Taylor. *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press, 2000.

[32] P. Dago-Casas, D. Gonzalez-Jimenez, L. L. Yu, and J. L. Alba-Castro. Single- and cross- database benchmarks for gender classification under unconstrained settings. *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, Nov 2011.

[33] J. G. Daugman. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *JOSA A*, 2(7):1160–1169, 1985.

[34] M. Demirkus, M. Toews, J. J. Clark, and T. Arbel. Gender classification from unconstrained video sequences. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, pages 55–62. IEEE, 2010.

[35] C. Ding and H. Peng. Minimum redundancy feature selection from microarray gene expression data. *Journal of bioinformatics and computational biology*, 3(02):185–205, 2005.

[36] E. Eidinger, R. Enbar, and T. Hassner. Age and gender estimation of unfiltered faces. *IEEE Transactions on Information Forensics and Security*, 9(12):2170–2179, Dec 2014.

[37] P. A. Estévez, M. Tesmer, C. A. Perez, and J. M. Zurada. Normalized mutual information feature selection. *Neural Networks, IEEE Transactions on*, 20(2):189–201, 2009.

[38] E. Fazl-Ersi, M. E. Mousa-Pasandi, R. Laganiere, and M. Awad. Age and gender recognition using informative features of various types. *2014 IEEE International Conference on Image Processing (ICIP)*, Oct 2014.

[39] M. Feld, F. Burkhardt, and C. A. Müller. Automatic speaker age and gender recognition in the car for tailoring dialog and mobile services. In *INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, September 26-30, 2010*, pages 2834–2837, 2010.

[40] R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179–188, 1936.

[41] A. Fourney and R. Laganiere. Constructing face image logs that are both complete and concise. *Fourth Canadian Conference on Computer and Robot Vision (CRV 2007)*, pages 488–494, May 2007.

[42] Fraunhofer IIS. SHORE - Sophisticated High-speed Object Recognition Engine. http://www.iis.fraunhofer.de/en/ff/bsy/dl/shore.html.

[43] Y. Freund and R. E. Schapire. A desicion-theoretic generalization of on-line learning and an application to boosting. In *Computational learning theory*, pages 23–37. Springer, 1995.

[44] Y. Fu, G. Guo, and T. S. Huang. Age synthesis and estimation via faces: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(11):1955–1976, Nov 2010.

[45] Y. Fu, T. M. Hospedales, T. Xiang, and S. Gong. Attribute learning for understanding unstructured social activity. In *Computer Vision–ECCV 2012*, pages 530–543. Springer, 2012.

[46] Y. Fu and T. S. Huang. Human age estimation with regression on discriminative aging manifold. *Multimedia, IEEE Transactions on*, 10(4):578–584, 2008.

[47] Y. Fu, Y. Xu, and T. S. Huang. Estimating human age by manifold analysis of face pictures and regression on aging features. In *Multimedia and Expo, 2007 IEEE International Conference on*, pages 1383–1386. IEEE, 2007.

[48] A. Gallagher and T. Chen. Understanding images of groups of people. In *Proc. CVPR*, 2009.

[49] X. Geng, Z.-H. Zhou, and K. Smith-Miles. Automatic age estimation based on facial aging patterns. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(12):2234–2240, 2007.

[50] X. Geng, Z.-H. Zhou, Y. Zhang, G. Li, and H. Dai. Learning from facial aging patterns for automatic age estimation. In *Proceedings of the 14th annual ACM international conference on Multimedia*, pages 307–316. ACM, 2006.

[51] A. Georghiades, P. Belhumeur, and D. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Trans. Pattern Anal. Mach. Intelligence*, 23(6):643–660, 2001.

[52] B. A. Golomb, D. T. Lawrence, and T. J. Sejnowski. Sexnet: A neural network identifies sex from human faces. In *Proceedings of the 1990 Conference on Advances in Neural Information Processing Systems 3*, NIPS-3, pages 572–577, San Francisco, CA, USA, 1990. Morgan Kaufmann Publishers Inc.

[53] R. Gross and V. Brajovic. An image preprocessing algorithm for illumination invariant face recognition. In *Audio-and Video-Based Biometric Person Authentication*, pages 10–18. Springer, 2003.

[54] G. Guo, Y. Fu, T. S. Huang, and C. R. Dyer. Locally adjusted robust regression for human age estimation. *Urbana*, 51:61801, 2008.

[55] G. Guo and G. Mu. Human age estimation: What is the influence across race and gender? *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops*, Jun 2010.

[56] G. Guo, G. Mu, Y. Fu, C. Dyer, and T. Huang. A study on automatic age estimation using a large database. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1986–1991. IEEE, 2009.

[57] G. Guo, G. Mu, Y. Fu, and T. S. Huang. Human age estimation using bio-inspired features. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 112–119. IEEE, 2009.

[58] A. Hadid and M. Pietikainen. From still image to video-based face recognition: an experimental analysis. *Sixth IEEE International Conference on Automatic Face and Gesture Recognition, 2004. Proceedings.*, 2004.

[59] J. Hayashi, M. Yasumoto, H. Ito, and H. Koshimizu. Method for estimating and modeling age and gender using facial image processing. In *Virtual Systems and Multimedia, 2001. Proceedings. Seventh International Conference on*, pages 439–448. IEEE, 2001.

[60] J. Hayashi, M. Yasumoto, H. Ito, Y. Niwa, and H. Koshimizu. Age and gender estimation from facial image processing. In *SICE 2002. Proceedings of the 41st SICE Annual Conference*, volume 1, pages 13–18. IEEE, 2002.

[61] B. Heisele, P. Ho, and T. Poggio. Face recognition with support vector machines: Global versus component-based approach. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 2, pages 688–694. IEEE, 2001.

[62] G. Heusch, F. Cardinaux, and S. Marcel. Lighting normalization algorithms for face verification. Technical report, IDIAP, 2005.

[63] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

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

[65] R. Iga, K. Izumi, H. Hayashi, G. Fukano, and T. Ohtani. A gender and age estimation system from face images. *Age*, 25:34, 2003.

[66] K. Irick, M. DeBole, V. Narayanan, R. Sharma, H. Moon, and S. Mummareddy. A unified streaming architecture for real time face detection and gender classification. *International Conference on Field Programmable Logic and Applications*, 2007.

[67] A. Jain and J. Huang. Integrating independent components and linear discriminant analysis for gender classification. In *Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference on*, pages 159–163. IEEE, 2004.

[68] O. Jesorsky, K. J. Kirchberg, and R. W. Frischholz. Robust face detection using the hausdorff distance. pages 90–95. Springer, 2001.

[69] R. A. Johnson and D. W. Wichern, editors. *Applied Multivariate Statistical Analysis*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1988.

[70] T. Kanno, M. Akiba, Y. Teramachi, H. Nagahashi, and A. Takeshi. Classification of age group based on facial images of young males by using neural networks. *IEICE TRANSACTIONS on Information and Systems*, 84(8):1094–1101, 2001.

[71] K. Karhunen. *Über lineare Methoden in der Wahrscheinlichkeitsrechnung*, volume 37. Universitat Helsinki, 1947.

[72] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International journal of computer vision*, 1(4):321–331, 1988.

[73] M. Kirby. *Geometric data analysis: an empirical approach to dimensionality reduction and the study of patterns*. John Wiley & Sons, Inc., 2000.

[74] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[75] Y. H. Kwon and N. da Vitoria Lobo. Age classification from facial images. In *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on*, pages 762–767. IEEE, 1994.

[76] A. Lanitis, C. Draganova, and C. Christodoulou. Comparing different classifiers for automatic age estimation. *IEEE Trans. Syst., Man, Cybern. B*, 34(1):621–628, Feb 2004.

[77] S. Z. Li, R. Chu, S. Liao, and L. Zhang. Illumination invariant face recognition using near-infrared images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(4):627–639, 2007.

[78] S. Z. Li, C. Zhao, M. Ao, and Z. Lei. Learning to fuse 3d+ 2d based face recognition at both feature and decision levels. In *Analysis and Modelling of Faces and Gestures*, pages 44–54. Springer, 2005.

[79] H.-C. Lian and B.-L. Lu. Multi-view gender classification using local binary patterns and support vector machines. In *Advances in Neural Networks-ISNN 2006*, pages 202–209. Springer, 2006.

[80] M. Loeve. Probability theory ii (graduate texts in mathematics), 1994.

[81] D. G. Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999.

[82] L. Lu and P. Shi. A novel fusion-based method for expression-invariant gender classification. In *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, pages 1065–1068. IEEE, 2009.

[83] M. Lyons, J. Budynek, A. Plante, and S. Akamatsu. Classifying facial attributes using a 2-D Gabor wavelet representation and discriminant analysis. *Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition (Cat. No. PR00580)*, 2000.

[84] E. Makinen and R. Raisamo. Evaluation of gender classification methods with automatically detected and aligned faces. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(3):541–547, 2008.

[85] E. Mäkinen and R. Raisamo. An experimental comparison of gender classification methods. *Pattern Recogn. Lett.*, 29(10):1544–1556, July 2008.

[86] 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*, volume 964, pages 965–966. Citeseer, 1999.

[87] M. Minear and D. C. Park. A lifespan database of adult facial stimuli. *Behavior Research Methods, Instruments, & Computers*, 36(4):630–633, 2004.

[88] B. Moghaddam and M.-H. Yang. Learning gender with support faces. *IEEE Trans Pattern Anal Machine Intell*, 24(5):707–711, 2002.

[89] H. Moon and P. J. Phillips. Analysis of pca-based face recognition algorithms. *Empirical Evaluation Techniques in Computer Vision*, 6:667–676, 1998.

[90] C. B. Ng, Y. H. Tay, and B. Goi. Vision-based human gender recognition: A survey. *CoRR*, abs/1204.1611, 2012.

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

[92] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans Pattern Anal Machine Intell*, 24(7):971–987, 2002.

[93] S. Paris and F. Durand. A fast approximation of the bilateral filter using a signal processing approach. *Lecture Notes in Computer Science*, pages 568–580, 2006.

[94] H. Park. Fast linear discriminant analysis using qr decomposition and regularization, 2007.

[95] K. Pearson. On lines and planes of closest fit to systems of points in space. *philosophical magazine*, 2(6):559–572, 1901.

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

[97] M. Pietikaeinen, T. Ojala, J. Nisula, and J. Heikkinen. Experiments with two industrial problems using texture classification based on feature distributions. In *Photonics for Industrial Applications*, pages 197–204. International Society for Optics and Photonics, 1994.

[98] S. M. Pizer, E. P. Amburn, J. D. Austin, R. Cromartie, A. Geselowitz, T. Greer, B. ter Haar Romeny, J. B. Zimmerman, and K. Zuiderveld. Adaptive histogram equalization and its variations. *Computer vision, graphics, and image processing*, 39(3):355–368, 1987.

[99] B. Poggio, R. Brunelli, and T. Poggio. Hyberbf networks for gender classification.

[100] Z.-U. Rahman, D. J. Jobson, and G. A. Woodell. Multi-scale retinex for color image enhancement. In *Image Processing, 1996. Proceedings., International Conference on*, volume 3, pages 1003–1006. IEEE, 1996.

[101] N. Ramanathan, R. Chellappa, and S. Biswas. Computational methods for modeling facial aging: A survey. *Journal of Visual Languages & Computing*, 20(3):131–144, 2009.

[102] T. Reponen, editor. *Information Technology Enabled Global Customer Service*. IGI Global, Hershey, PA, USA, 2002.

[103] K. Ricanek and T. Tesafaye. Morph: A longitudinal image database of normal adult age-progression. *7th International Conference on Automatic Face and Gesture Recognition (FGR06)*, 2006.

[104] M. Riesenhuber and T. Poggio. Hierarchical models of object recognition in cortex. *Nature neuroscience*, 2(11):1019–1025, 1999.

[105] R. Russell. A sex difference in facial contrast and its exaggeration by cosmetics. *Perception*, 38(8):1211–1219, 2009.

[106] Y. Saatci and C. Town. Cascaded classification of gender and facial expression using active appearance models. In *Automatic Face and Gesture Recognition, 2006. FGR 2006. 7th International Conference on*, pages 393–398. IEEE, 2006.

[107] F. Scalzo, G. Bebis, M. Nicolescu, L. Loss, and A. Tavakkoli. Feature fusion hierarchies for gender classification. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–4. IEEE, 2008.

[108] R. E. Schapire. A brief introduction to boosting. In *Ijcai*, volume 99, pages 1401–1406, 1999.

[109] H. S. Seung and D. D. Lee. The manifold ways of perception. *Science*, 290(5500):2268–2269, 2000.

[110] A. Sgroi, K. W. Bowyer, and P. J. Flynn. The prediction of old and young subjects from iris texture. *2013 International Conference on Biometrics (ICB)*, Jun 2013.

[111] G. Shakhnarovich, P. A. Viola, and B. Moghaddam. A unified learning framework for real time face detection and classification. In *Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on*, pages 14–21. IEEE, 2002.

[112] C. Shan. Learning local features for age estimation on real-life faces. *Proceedings of the 1st ACM international workshop on Multimodal pervasive video analysis - MPVA10*, 2010.

[113] C. Shan. Learning local binary patterns for gender classification on real-world face images. *Pattern Recognition Letters*, 33(4):431–437, 2012.

[114] S. Shan, W. Zhang, Y. Su, X. Chen, and W. Gao. Ensemble of piecewise fda based on spatial histograms of local (Gabor) binary patterns for face recognition. *18th International Conference on Pattern Recognition*, 2006.

[115] S. K. Smith and C. J. DeFrances. Assessing measurement techniques for identifying race, ethnicity, and gender: Observation-based data collection in airports and at immigration checkpoints. 2003.

[116] N. Sun, W. Zheng, C. Sun, C. Zou, and L. Zhao. Gender classification based on boosting local binary pattern. In *Advances in Neural Networks-ISNN 2006*, pages 194–201. Springer, 2006.

[117] Z. Sun, G. Bebis, X. Yuan, and S. J. Louis. Genetic feature subset selection for gender classification: A comparison study. In *Applications of Computer Vision, 2002.(WACV 2002). Proceedings. Sixth IEEE Workshop on*, pages 165–170. IEEE, 2002.

[118] S. Tamura, H. Kawai, and H. Mitsumoto. Male/female identification from 8 to 6 very low resolution face images by neural network. *Pattern Recognition*, 29(2):331–335, 1996.

[119] X. Tan and B. Triggs. Enhanced local texture feature sets for face recognition under difficult lighting conditions. *IEEE Transactions on Image Processing*, 19(6):1635–1650, Jun 2010.

[120] J. E. Tapia and C. A. Pérez Flores. Gender classification based on fusion of different spatial scale features selected by mutual information from histogram of lbp, intensity, and shape. 2013.

[121] M. A. Turk and A. P. Pentland. Face recognition using eigenfaces. In *Computer Vision and Pattern Recognition, 1991. Proceedings CVPR'91., IEEE Computer Society Conference on*, pages 586–591. IEEE, 1991.

[122] K. Ueki, T. Hayashida, and T. Kobayashi. Subspace-based age-group classification using facial images under various lighting conditions. In *Automatic Face and Gesture Recognition, 2006. FGR 2006. 7th International Conference on*, pages 6–pp. IEEE, 2006.

[123] I. Ullah, M. Hussain, H. Aboalsamh, G. Muhammad, A. M. Mirza, and G. Bebis. Gender recognition from face images with dyadic wavelet transform and local binary pattern. In *Advances in Visual Computing*, pages 409–419. Springer, 2012.

[124] M. Uřičář, V. Franc, and V. Hlaváč. Detector of facial landmarks learned by the structured output SVM. In *VISAPP '12: Proceedings of the 7th International Conference on Computer Vision Theory and Applications*, volume 1, pages 547–556, Feb. 2012.

[125] M. Verma and S. Agarwal. Fingerprint based male-female classification. *Proceedings of the International Workshop on Computational Intelligence in Security for Information Systems CISIS08*, pages 251–257, 2009.

[126] M. Vidal-Naquet and S. Ullman. Object recognition with informative features and linear classification. In *ICCV*, volume 3, page 281, 2003.

[127] P. Viola and M. Jones. Robust real-time face detection. *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, 2001.

[128] J.-G. Wang, J. Li, C. Y. Lee, and W.-Y. Yau. Dense sift and gabor descriptors-based face representation with applications to gender recognition. In *Control Automation Robotics & Vision (ICARCV), 2010 11th International Conference on*, pages 1860–1864. IEEE, 2010.

[129] Wikipedia. Google glass — wikipedia, the free encyclopedia, 2015. [Online; accessed 20-March-2015].

[130] L. Wiskott, J.-M. Fellous, N. Krüger, and C. von der Malsburg. Face recognition and gender determination. 1995.

[131] L. Wiskott, J.-M. Fellous, N. Kuiger, and C. von der Malsburg. Face recognition by elastic bunch graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):775–779, Jul 1997.

[132] L. Wolf, T. Hassner, and Y. Taigman. Descriptor based methods in the wild. In *Workshop on Faces in'Real-Life'Images: Detection, Alignment, and Recognition*, 2008.

[133] B. Wu, H. Ai, and C. Huang. Lut-based adaboost for gender classification. In *Audio- and Video-Based Biometric Person Authentication*, pages 104–110. Springer, 2003.

[134] B. Xia, H. Sun, and B.-L. Lu. Multi-view gender classification based on local gabor binary mapping pattern and support vector machines. In *Neural Networks, 2008. IJCNN 2008.(IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on*, pages 3388–3395. IEEE, 2008.

[135] M.-H. Yang, D. Kriegman, and N. Ahuja. Detecting faces in images: A survey. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(1):34–58, 2002.

[136] J. Ylioinas, A. Hadid, and M. Pietikainen. Age classification in unconstrained conditions using lbp variants. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 1257–1260. IEEE, 2012.

[137] S. Zafeiriou, A. Tefas, and I. Pitas. Gender determination using a support vector machine variant. In *16th European Signal Processing Conference (EUSIPCO-2008), Lausanne, Switzerland*, pages 2–6, 2008.

[138] J. Zheng and B.-L. Lu. A support vector machine classifier with automatic confidence and its application to gender classification. *Neurocomputing*, 74(11):1926–1935, 2011.