Automatic Recognition of Facial Expressions

A Thesis Submitted in Fulfillment of the Requirements

for the Degree of Doctor of Philosophy

Seyed Mehdi Lajevardi

B.Eng.

School of Electrical and Computer Engineering

College of Science, Engineering and Health

RMIT University

September 2011

Declaration

I certify that except where due acknowledgement has been made, the work is that of the author alone; the work has not been submitted previously, in whole or in part, to qualify for any other academic award; the content of the thesis is the result of work which has been carried out since the official commencement date of the approved research program; and, any editorial work, paid or unpaid, carried out by a third party is acknowledged.

Seyed Mehdi Lajevardi

September 2011

This dissertation is dedicated to my parents and my wife

Acknowledgements

First of all, thanks are due to the Almighty God for giving me the strength and patience to complete my studies and write this dissertation, which I sincerely hope will contribute to the welfare of people. It is a pleasure to thank those who made this thesis possible. I would like to express my deepest gratitude to my supervisors Adjunct Professor Zahir Hussain, Professor Henry Wu, Associate Professor James Scott and Dr. Margaret Lech for their support, guidance and trust through out my candidature at RMIT University. I would like to acknowledge the financial support of the Australian Government and RMIT University through the Australian Postgraduate Awards (APA).

Special thanks go to my friends and colleagues in the School of Electrical and Computer Engineering at RMIT University. The staff have been very helpful and cooperative. I would like to thank them all for their kindness and friendliness.

Last but not least, I would like to give my deepest and most sincere thanks to my parents and my family whose precious support, patience and inspiration are invaluable.

Abstract

Facial expression is a visible manifestation of the affective state, cognitive activity, intention, personality and psychopathology of a person; it not only expresses our expressions, but also provides important communicative cues during social interaction. Expression recognition can be embedded into a face recognition system to improve its robustness. In a real-time face recognition system where a series of images of an individual are captured, facial expression recognition (FER) module picks the one which is most similar to a neutral expression for recognition, because normally a face recognition system is trained using neutral expression images. In the case where only one image is available, the estimated expression can be used either to decide which classifier to choose or to add some kind of compensation. In a human-computer interaction (HCI), expression is an input of great potential in terms of communicative cues. This is especially true in voice-activated control systems. This implies an FER module can markedly improve the performance of such systems. Customer's facial expressions can also be collected by service providers as implicit user feedback to improve their service. Compared with a conventional questionnaire-based method, this should be more reliable and furthermore, has virtually no cost.

The main challenge for FER system is to attain the highest possible classification rate for the recognition of six expressions (Anger, Disgust, Fear, Happy, Sad, Surprise). The other challenges are the illumination variation, rotation and noise. In this thesis, different methods for image pre-processing,

feature extraction, feature selection and classification are investigated and several techniques are proposed for feature extraction, selection and classification. Furthermore, the effect of colour information components in different colour spaces is investigated and studied for FER system.

Publications and Awards

Below are the publications and the awards in conjunction with the author's PhD candidacy:

Journal Publications

1. S. M. Lajevardi and Z. M. Hussain, "Higher order orthogonal moments for invariant facial expression recognition", *Digital Signal Processing,* vol. 20, pp. 1771-1779, 2010.
2. S. M. Lajevardi and Z. M. Hussain, "Novel higher-order local autocorrelation-like feature extraction methodology for facial expression recognition", *IET on Image Processing,* vol. 4, pp. 114-119, 2010.
3. S. M. Lajevardi and Z. M. Hussain, "Automatic facial expression recognition: feature extraction and selection", *Signal, Image and Video Processing,* pp. 1-11, 2010.
4. S. M. Lajevardi and Z. M. Hussain, "Hybrid feature extraction for facial expression recognition", *Advances in Modelling Series B: Signal Processing and Pattern Recognition,* vol. 53, pp. 34-50, 2009.
5. S. M. Lajevardi and Z. M. Hussain, "Feature extraction for facial expression recognition based on hybrid face regions", *Advances in Electrical and Computer Engineering,* vol. 9, pp. 63-67, 2009.

Refereed Conference Publications

1. S. M. Lajevardi and Z. M. Hussain, "A novel Gabor filter selection based on spectral difference and minimum error rate for facial expression recognition", *in Proceeding of International Conference on Digital Image Computing: Techniques and Applications(DICTA 2010),* Sydney, Australia, 2010, pp. 137-140.
2. S. M. Lajevardi and Z. M. Hussain, "Emotion recognition from color facial images based on multilinear image analysis and log-Gabor filters", *in Proceeding of 25th International Conference on Image and Vision Computing New Zealand, (IVCNZ 2010),* Queenstown, New Zealand, 2010, pp. 10-14.
3. S. M. Lajevardi and Z. M. Hussain, "Contourlet structural similarity for facial expression recognition", *in Proceeding of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2010),* Dallas, USA, 2010, pp. 1118-1121.
4. S. M. Lajevardi and Z. M. Hussain, "Feature selection for facial expression recognition based on mutual information", *in Proceeding of 5th IEEE GCC Conference and Exhibition,* Kuwait, Kuwait, 2009, pp. 1-5.
5. S. M. Lajevardi and Z. M. Hussain, "Local correlation for noisy facial expression images", *in Proceeding of International Symposium on Bioelectronics and Bioinformatics,* Melbourne, Australia, 2009, pp. 64-67.
6. S. M. Lajevardi and Z. M. Hussain, "Zernike moments for facial expression recognition", *in Proceeding of International Conference on Communication, Computer and Power (ICCCP'09),* Muscat, Oman, 2009, pp. 378-381.
7. S. M. Lajevardi and Z. M. Hussain, "Facial expression recognition using log-Gabor filters and local binary pattern operators", *in Proceeding of International Conference on Communication, Computer and Power (ICCCP'09),* Muscat, Oman, 2009, pp. 349-353.

8. S. M. Lajevardi and Z. M. Hussain, "Facial expression recognition: Gabor filters versus higher-order correlators", *in Proceeding of International Conference on Communication, Computer and Power (ICCCP'09),* Muscat, Oman, 2009, pp. 354-358.
9. S. M. Lajevardi and Z. M. Hussain, "Local feature extraction methods for facial expression recognition", *in Proceeding of 17th European Signal Processing Conference (EUSIPCO 2009),* Glasgow, Scotland, 2009, pp. 60-64.
10. S. M. Lajevardi and Z. M. Hussain, "Feature selection for facial expression recognition based on optimization algorithm", *in Proceeding of 2nd International Workshop on Nonlinear Dynamics and Synchronization (INDS'09)*, Klagenfurt, Austria, 2009, pp. 182-185.
11. S. M. Lajevardi, K. L. Neville and Z. M. Hussain, "Facial expression recognition over FFT-OFDM", *in Proceeding of International Conference on Advanced Technologies for Communications (ATC'09),* Haiphong, Vietnam, 2009, pp. 35-38.
12. S. M. Lajevardi, K. Abdullah and Z. M. Hussain, "Modulation comparison over OFDM channel for facial expression recognition", *in Proceeding of International Conference on Advanced Technologies for Communications (ATC'09),* Haiphong, Vietnam, 2009, pp. 137-140.
13. K. Abdullah, S. M. Lajevardi and Z. M. Hussain, "QAM modulations over wavelet based OFDM channel for facial expression recognition", *in Proceeding of Australasian Telecommunication Networks and Applications Conference (ATNAC),* 2009, pp. 1-5.
14. S. M. Lajevardi and M. Lech, "Averaged Gabor filter features for facial expression recognition", *in Proceeding of Digital Image Computing: Techniques and Applications (DICTA '08),* Canberra, Australia, 2008, pp. 71-76.
15. S. M. Lajevardi and M. Lech, "Facial expression recognition using neural networks and log-Gabor filters", *in Proceeding of Digital Image Comput-*

ing: Techniques and Applications (DICTA '08), Canberra, Australia, 2008, pp. 77-83.

16. S. M. Lajevardi and M. Lech, "Facial expression recognition from image sequences using optimized feature selection", *in Proceeding of 23rd International Conference on Image and Vision Computing New Zealand, (IVCNZ 2008),* Christchurch, New Zealand, 2008, pp. 1-6.

Submitted Papers

1. S. M. Lajevardi and H. R. Wu, "Facial expression recognition using component color space information and multilinear image analysis", *Submitted to Computer Vision and Image Understanding*, 2011.
2. S. M. Lajevardi and H. R. Wu, "Facial expression recognition in perceptual color space", *Submitted to IEEE Transactions on Image Processing,* 2011.

Awards

1. Awarded diploma and golden medal of international federation of inventors associations (IFIA) for best scientific papers in 2010.
2. Awarded Australian Postgraduate Awards (APA) scholarship for research student 2010-2011.
3. Awarded ECE scholarship for research student 2009-2010.
4. 2008 Postgraduate Research Presentation Award at RMIT University.

Keywords

Automated facial expression recognition system, facial feature utilization, appearance modelling, expression recognition, facial action processing, facial local models, Gabor filters, logarithmic Gabor filters, contourlet transform, local binary pattern operator, Zernike moments, mutual information, facial expression recognition, structural similarity, noisy pattern recognition, perceptual colour spaces, multi-linear image analysis, CIELab, CIELuv.

Contents

List of Figures

List of Tables

List of Acronyms and Principal Symbols

1-D	One Dimension
2-D	Two Dimensions
3-D	Three Dimensions
AAM	Active Appearance Model
ACO	Ant Colony Optimization
AGF	Average Gabor Filters
ALGF	Average Log-Gabor Filters
AI	Artificial Intelligence
ANNs	Artificial Neural Networks
ARD	Automatic Relevance Determination
AU	Action Unit
BnB	Branch-and-Bound Methods
BU-3DFE	Binghamton University 3-D Facial Expression
CIE	International Commission on Illumination
CPU	Central processing unit
CT	Contourlet Transform
CTSSIM	Contourlet Transform Structural Similarity
CWSSIM	Complex Wavelet Structural Similarity
dB	Decibel
DFB	Directional Filter Bank
DFT	Discrete Fourier Transform
DSP	Digital Signal Processing

FACS	Facial Action Coding System
FER	Facial Expression Recognition
FLD	Fisher Linear Discriminant
FPS	Frame Per Second
FP	False Positive
FSA	Feature Selection Algorithm
GA	Genetic Algorithm
GF	Gabor Filters
GFSSIM	Gabor Filters Structural Similarity
GHz	Gigahertz
HCI	Human-Computer Interaction
HFR	Hybrid Face Region
HLAC	Higher Order Local Autocorrelation
HPF	High Pass filter
HOS	Higher Order Spectra
HVS	Human Vision System
Hz	Hertz
ICA	Independent Component Analysis
JAFFE	Japanese Female Facial Expression
KNN	K-nearest neighbour
L	Lighting
LBP	Local Binary Pattern
LDA	Linear Discriminant Analysis
LFA	Local Feature Analysis
LGF	Logarithmic Gabor Filters
LP	Laplacian pyramid
LPF	Low Pass Filter
MHz	Megahertz

MI	Mutual Information
MID	Mutual Information Difference
MIFS	Mutual Information Feature Selection
MIQ	Mutual Information Quotient
MLP	Multi-layer perceptron
MRMR	Minimum Redundancy Maximum Relevance
MSE	Mean Squared Error
NB	Naive Bayesian
NN	Neural Network
NTSC	National Television System Committee
PAL	Phase Alternating Line
PCA	Principal Component Analysis
PDF	Probability Density Function
PDFB	Pyramidal Directional Filter Bank
PSNR	Peak Signal to Noise Ratio
RBF	Radial Basis Function
R	Red
RGB	Red, Green, Blue
ROC	Receiver Operation Characteristic
SEC	Second
SNR	Signal to Noise Ratio
SPD	Spectral Power Distribution
SSIM	Structural Similarity
SVD	Singular Value Decomposition
SVM	Support Vector Machine
THz	Tera Hertz
TP	True Positive
TPCF	Tensor Perceptual Colour Framework
Y	Luminance
ZM	Zernike Moments

Preface

A key requirement for developing any innovative system in a computing environment is to integrate a sufficiently friendly interface with the average end user. In human-to-human dialogue, the articulation and the perception of facial expressions form a communication channel that is supplementary to voice and that carries crucial information about the mental, emotional and even physical states of the conversation partners. In their simplest form, facial expressions can indicate whether a person is happy or angry. More subtly, expressions can provide either conscious or subconscious feedback from a listener to a speaker to indicate understanding of, empathy for, or even skepticism toward what the speaker is saying. Therefore, it is obvious that analysis and automatic recognition of facial expression can improve human-computer interaction (HCI) or even social interaction. The work of this thesis aims at designing a robust Facial Expression Recognition (FER) system by combining various techniques from image processing, computer vision and pattern recognition. FER can be considered as a special case of a pattern recognition problem and many techniques are available. In the designing of an FER system, these resources and existing algorithms are used to build blocks of FER system. A major aim of this work is to determine an optimal combination of algorithms and robust methods for different components of an FER system. The system that is proposed in this dissertation has four modules, i.e., image pre-processing, feature extraction, feature selection and classification, for each of which several candidate methods are implemented and eventually

the optimal configuration is sought by comparing the performance of different combinations.

In this thesis, several innovative methods based on image processing and pattern recognition theory have been devised and implemented. The main contributions of algorithms and advanced modelling techniques are summarized as follows. 1) A new feature extraction approach called HLAC-like (higher-order local autocorrelation-like) features has been presented to detect and to extract facial features from face images. 2) An innovative design is introduced with the ability to detect cases using face feature extraction method based on orthogonal moments for images with noise and/or rotation. Using this technique, the expression from face images with high levels of noise and even rotation has been recognized properly. 3) A facial expression recognition system is designed based on the combination region. In this system, a method called hybrid face regions (HFR) according to the combined part of an image is presented. Using this method, the features are extracted from the components of the face (eyes, nose and mouth) and then the expression is identified based on these features. 4) A novel classification methodology has been proposed based on structural similarity algorithm in facial expression recognition scenarios. 5) A new methodology for expression recognition is presented using colour facial images based on multi-linear image analysis. In this scenario, the colour images are unfolded to two dimensional (2-D) matrix based on multi-linear algebra and then classified based on multi-linear discriminant analysis (LDA) classifier. Furthermore, the colour effect on facial images of various resolutions is studied for FER system. The addressed issues are challenging problems and are substantial for developing a facial expression recognition system.

I hope that this work will help researchers working in image processing, computer vision and related topics and inspire further research in these fields.

Seyed Mehdi Lajevardi

Melbourne, 2011

Chapter 1

Introduction

1.1 Background

Facial expression plays an important role in the cognition of human expressions. Recognition of facial expressions can be a vital component of natural human-machine interfaces. It may also be used in behavioural affective state, cognitive science and clinical practice. A facial expression is a visible manifestation of activity, intention, personality, and psychopathology of a person [1]. Facial expressions, as well as other gestures, convey non-verbal communication cues in human face-to-face interactions. These cues may also complement speech by helping the listener to elicit the intended meaning of spoken words. Experiment in [2] reported that facial expressions played a very important part in the process of verbal communication between speakers. The facial expression of a speaker accounts for about 55 percent of the cognition effect, 38 percent of information is conveyed by voice intonation and only 7 percent by the spoken words.

Basic facial expressions typically recognized in automatic affect-recognition tasks are happy, sad, fear, anger, disgust and surprise [3]. "Neutral" is also considered as a basic facial expression for convenience. Although the appearance of these expressions may vary between individuals, humans can still recognize a wide range of different expressions. For example, even if we are not familiar with someone's face, we can recognize the person's facial expression

due to the universality of affect expressions [4]. Similarly, we can recognize a familiar person regardless of the person's facial expression. However, it is a challenging task for a computer vision system to recognize an individual across different expressions or to classify the basic facial expressions across different persons. Numerous methodologies have been proposed for facial expression analysis from both static images and image sequences [1, 5]. This chapter states the main objective of thesis and problems. Furthermore, the thesis organization is presented along with brief descriptions of all chapters at the end of this chapter.

1.2 Thesis Objectives

The primary goal of this research is to design, implement and evaluate a novel facial expression recognition system using various statistical learning techniques. This goal will be realized through the following objectives:

- To advance the knowledge on Facial Expression Recognition (FER) systems.
- Developing robust computer vision techniques for the analysis and recognition of facial expressions from static images, image sequences and videos.
- Improving and optimizing both feature extraction and selection methods for facial expression recognition.
- To overcome some of the disadvantages and limitations of existing facial expression recognition methods.

1.2.1 Research Questions

This PhD thesis attempts to answer the following questions:

1. What are the advantages and deficiencies of existing facial recognition techniques?

2. Is it possible to make the image pre-processing for facial expression recognition fully automatic?
3. How to design new feature extraction methods for FER system that overcome deficiencies of existing methods?
4. To what degree can the characteristic feature data be compressed without significant loss of the classification accuracy?
5. How to improve the existing methods or how to design new classification strategies providing higher recognition accuracy and better computational efficiency when compared with the existing techniques?

1.2.2 Research Aims

The research reported in this thesis aims at targeting the disadvantages and limitations of existing facial expression recognition methods. The majority of current facial recognition techniques applies strong restrictions to the image quality, background, illumination and scene. The key challenge will be to achieve optimal FER performance via pre-processing, feature extraction and classification, under conditions of wide input variability. To attain successful recognition performance, feature extraction and classification techniques, which have high level of insensitivity to translation, scaling and in plane rotation of the image, will be investigated. Problems related to the complexity of the image background, occlusions and uncontrolled lighting will be also addressed. Miscellaneous sources of facial variability such as age, gender, race, facial hair and make-up will be tested.

The new efficient facial expression recognition algorithms that can be implemented into a number of various applications where computers take on a social role such as an instructor or a helper. The functionality of the system may be enhanced by recognizing user's expressions.

1.3 Original Contributions

This thesis makes a number of original contributions to the body of image processing and pattern recognition knowledge both in theory and in implementation. It presents a number of novel algorithms and techniques to model and design new magical systems.

The main contributions of this dissertation include:

1. A new methodology called TPCF, for expression recognition from colour facial images with different resolutions based on multi-linear image analysis [6];
2. A new approach to extraction of the facial features based on orthogonal moments for images with noise and/or rotation [7];
3. A novel classification methodology based on structural similarity algorithm in facial expression recognition [8];
4. Design of a new feature extraction method based on higher-order spectra [9];
5. A novel approach to feature extraction from different parts of the face [10]; and
6. A new image pre-processing approach called facial detection using image sequences [11].

1.4 Thesis Organization

This thesis is structured as follows. Chapter 2 includes the recent state-of-the-art methods for facial expression recognition systems. At the beginning, the system modules are introduced, and then various up to date contributions to each module are described and analyzed. The face pre-processing methods, facial feature extraction and classification presented in detail. Furthermore, face databases which are important to FER research are also discussed in this chapter. In Chapter 3, the facial expression pre-processing module is discussed. It consists of face detection, facial detection, alignment and normalization. The

colour image pre-processing based on multi-linear analysis is discussed as well. Chapter 4 contains facial feature extraction approaches. This chapter investigates different feature extraction methods. Several local and holistic feature extraction methods are examined theoretically and implemented. Chapter 5 discusses different feature selection methods. Due to high dimensionality features, feature selection is employed to determine the most discriminative and informative features. Chapter 6 elaborates on classification algorithms and finalizes the system configuration. Different classification techniques are implemented and their performance on different features are tested. The final system configuration is determined based on the experimental results. Chapter 7 describes the effect of colour information on expression recognition. The final conclusions and possible future directions are summarized in Chapter 8.

Chapter 2

FER System Review

2.1 Introduction

Automatic facial expression recognition is a sub-area of face analysis research that is based heavily on methods of computer vision, machine learning and image processing. Facial expressions can play an important role wherever humans interact with machines. An automatic recognition of facial expressions may act as a component of natural human-machine interfaces. Synthetic speech with expressions in the voice may sound more pleasing than a monotonous voice. Computer "agents" can learn the user's preferences through the user's expressions. FER system can also help the human users monitor their stress level. In clinical settings, recognizing a person's inability to express certain facial expressions may help diagnose early psychological disorders [12, 13].

Facial expression recognition and expression analysis can help humanize computers and robots. Knowing the user's expressions, the computer can, e.g., become a more effective tutor. In addition, FER will be helpful for companies which are important for them to know about their customer services. Furthermore, it can modify the way of teaching for e-learning and visual classes. Many efforts either to create novel or to improve existing FER systems are thus inspired by advances in these related fields. Due to the wide range of applications in human-computer interaction, telecommunication, law enforcement and psychological research, facial expression analysis has become an active research

area. The existing literature in this area is reviewed before describing the author's contributions to the field of automatic facial expression recognition.

This survey includes the major algorithms that have significantly impacted the development of FER systems. In this chapter, the more generic algorithms of facial expression recognition are presented and both the sake of comprehensiveness and the subtle benefits achieved by these techniques are discussed in the following sections.

2.2 Facial Expression Analysis

2.2.1 Facial Action Coding System

Facial Action Coding System (FACS), which was developed by Paul Ekman and Wallace Friesen in 1976, is a system to taxonomies human facial expressions [14, 15]. It is a common standard to systematically categorize the physical expression of expressions, which has been proven useful to psychologists and to animators. FACS consists of forty four (44) action units (AUs) which are related to contraction of a specific set of facial muscles. Thirty (30) AUs are anatomically related to the contractions of specific facial muscles: twelve (12) are for the upper face and eighteen (18) are for the lower face. AUs can occur either singly or in combination. FACS provides the descriptive power necessary to describe the details of facial expression. Conventionally, FACS code is manually labelled by trained observers while viewing videotaped facial behaviour in slow motion. In recent years, several attempts were made to do this automatically [16, 17]. The advantage of FACS is its ability to capture the subtlety of facial expression, however FACS itself is purely descriptive and includes no inferential labels. That means in order to get the expression estimation, the FACS code needs to be converted into the Emotional Facial Action System or similar systems.

2.2.2 Prototypical Facial Expressions

Instead of describing the detailed facial features, most FER systems attempt to recognize a small set of prototypic emotional expressions. The most widely-used set is six basic expression categories that have been shown to be recognizable across human cultures [Figure 2.1]. These expressions, or facial configurations, have been recognized in people from widely divergent cultural and social backgrounds [3] and they have been observed even in the faces of individuals born deaf and blind. Basic facial expressions typically recognized in automatic affect-recognition tasks are anger, disgust, fear, happy, sad and surprise [3].

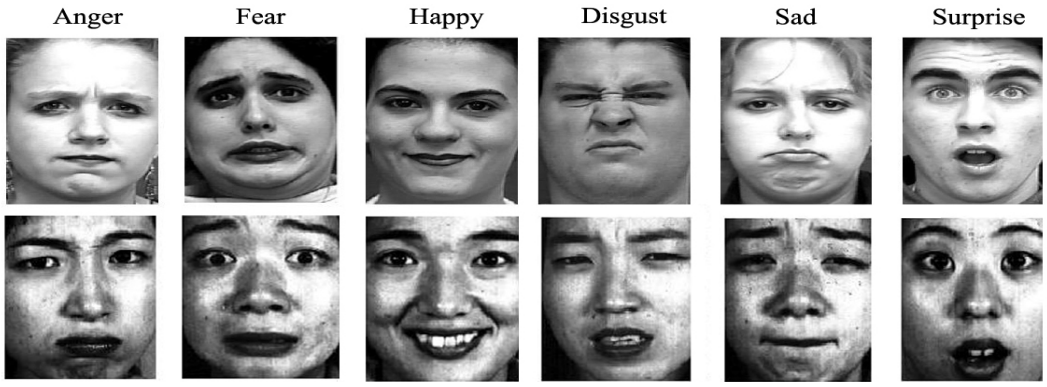


Figure 2.1: Six facial expressions. [1]

2.3 FER System Modules

FER can be considered as a special pattern recognition system plus image processing module. Generally FER systems contain four modules. In this work, one more module called feature selection is added to the FER system. Figure 2.2 illustrates the Block diagram of FER systems.

Normally, image pre-processing module includes face detection and image normalization. Face detection is a methodology that determines the face area or areas in the input (digital) image. For digital video input, since the number

[1] Images from Cohn-Kanade database and JAFFE database [13, 18]

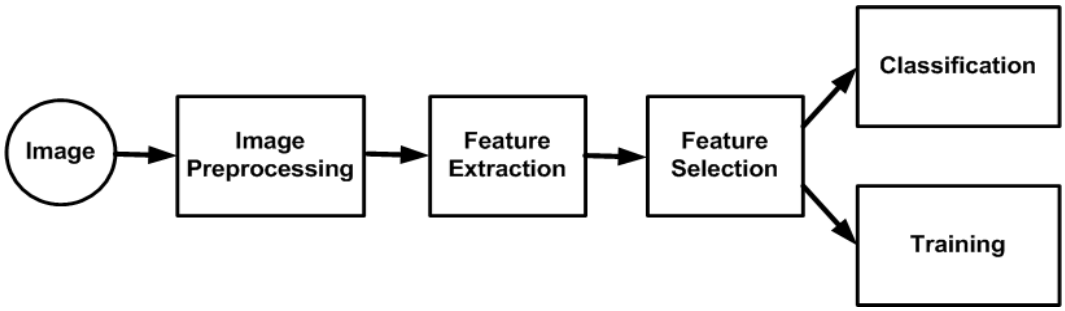


Figure 2.2: Modules of FER system.

of images is huge (25 frame per second (fps) for the PAL standard derived videos or 30 fps for the NTSC standard derived videos [19]), applying the face detection to all frames are complex and time consuming. To be more efficient and also to achieve more robust performance, a proposed facial detection algorithm [11] is applied on frames. In image normalization step, the image is normalized with respect to photometrical properties such as illumination and gray-scale level. Then, the size of the image is scaled to an appropriate dimension [12, 19]. The feature extraction module involves simplifying the amount of resources required to describe a large set of data accurately. It is performed on a face image to provide effective information that should be useful for recognition and classification [12, 13]. The extracted features are sent to feature selection module from which an appropriate sub-set of features is selected to represent the inputs (feature vectors) [20, 21]. These inputs have an effect on both the computational complexity and the quality of the classification results. In this stage, the most discriminative features are selected for classification. The feature vector is sent to a classifier and compared with the training data to produce a recognition output [12, 18, 22]. The output can be one of the expressions (class labels).

2.4 Image Pre-processing

The image pre-processing procedure comes as a very important step in the facial expression recognition task [12, 19]. The aim of the pre-processing phase is to obtain images which have normalized intensity, uniform size and shape and depict only a face expressing certain expression. Furthermore, it should eliminate or minimize the effects of illumination and lighting.

2.4.1 Face Detection

The first step of image pre-processing module is detecting the locations of the face in the image. Because of variability in scale, location, orientation and pose, face detection from a single image is a challenging task. Facial expression, occlusion and lighting conditions make the detecting of the face complex. The goal of face detection is to determine whether or not any face is in the image and, if present, return the face location and extent of each face. The challenges associated with face detection can be attributed to the following factors [23, 24, 25]:

- **Pose**: The images of a face vary due to the relative camera-face pose (frontal, 45 degree, profile, upside down) and some facial features such as an eye or the nose may become partially or completely occluded.
- **Structural components**: Facial features such as beards, mustaches and glasses may or may not be present and there is a variation among these components such as shape, colour and size.
- **Facial expression**: The appearance of faces are directly affected by the facial expressions.
- **Occlusion**: Faces may be partially occluded by other objects (i.e., in an image with a group of people, some faces may partially occlude other faces).
- **Image orientation**: Face images directly vary for different rotations about the camera's optical axis.

- **Imaging conditions**: When the image is formed, factors such as lighting (spectra, source distribution and intensity) and camera characteristics (sensor response, lenses) affect the appearance of a face.

Several methods have been proposed to deal with above challenges. The appearance-based algorithms avoid difficulties in 3-D modelling structures of faces. Furthermore, due to facial expression and head pose, the variation of 3-D structures affects the facial appearance and makes the boundary between face and non-face highly complex [26]. Therefore, the successful face detection algorithms are mostly focused only on appearance [27]. Numerous representations have been proposed for face detection, including pixel-based [23, 24, 28], parts-based [29, 30], local edge features [31, 32], Haar wavelets [33] and Haar-like features [25, 34, 35]. While earlier holistic representation schemes are able to detect faces [23, 24, 28], the recent systems with Haar-like features [25, 36] have demonstrated impressive empirical results in detecting faces under occlusion. A large and representative training set of face images is essential for the success of learning-based face detectors. From the set of collected data, more positive examples can be synthetically generated by perturbing, mirroring, rotating and scaling the original face images. On the other hand, it is relatively easier to collect negative examples by randomly sampling images without face images [24]. As face detection can be mainly formulated as a pattern recognition problem, numerous algorithms have been proposed to learn their generic templates [23, 24, 25, 37] (e.g., eigenface and statistical distribution) or discriminant classifiers (e.g., neural networks, Fisher linear discriminant, sparse network of Winnows, decision tree, Bayes classifiers, support vector machines and AdaBoost). Typically, a good face detection system needs to be trained with several iterations. One common method to further improve the system is to bootstrap a trained face detector with test sets and re-train the system with the false positives as well as negatives. This process is repeated several times in order to further improve the performance of a face detector.

2.4.2 Image Normalization

Image normalization is a process that changes the range of pixel intensity values [19]. In digital signal processing (DSP), it is referred to as dynamic range expansion [38]. The purpose of dynamic range expansion in the various applications is usually to bring the image, or other type of signal, into a range that is more familiar or normal to the senses, hence the term normalization. The motivation is to achieve consistency in dynamic range for a set of images to avoid mental distraction or fatigue. Auto-normalization in image processing software typically normalizes to the full dynamic range of the number system specified in the image file format. Normalization is sometimes called contrast stretching. Contrast stretching is a simple image enhancement technique that attempts to improve the contrast of an image by stretching the range of intensity values. It can apply a scaling function to the image pixel values. To perform the contrast stretching technique, it is necessary to specify the upper and lower pixel value limits over which the image is to be normalized. Often these limits will just be the minimum and maximum pixel values that the image type concerned allows. Let $\mathbf{x}_{\text{Original}} = \{\mathbf{x}_{\text{Original}}[n_1, n_2, n_3] | 1 \leq n_1 \leq N_1', 1 \leq n_2 \leq N_2', 1 \leq n_3 \leq 3\}$ denotes an original image, where N_1' and N_2' are the original size of image given from database and n_3 represents the three colour components (Red, Green, Blue) for colour images. The image, $\mathbf{x} = \{\mathbf{x}[n_1, n_2, n_3] | 1 \leq n_1 \leq N_1, 1 \leq n_2 \leq N_2, 1 \leq n_3 \leq 3\}$, represents a certain face image which only contains the face as shown in Figure 2.3(a). For gray-scale images, there is no colour components, $\mathbf{x}_{\text{Original}}[n_1, n_2]$. For 8 bits gray-scale face images, $\mathbf{x}[n_1, n_2]$, the lower and the upper limits are normally 0 and 255, respectively. The simplest sort of normalization then scans the image to find the lowest and highest pixel values currently present in the image. Then each pixel is scaled to new value by

$$\mathbf{x}_{norm}[n_1, n_2] = (\mathbf{x}[n_1, n_2] - x_{\min})(\frac{255 - 0}{x_{\max} - x_{\min}}) + x_{\min} \tag{2.1}$$

and

$$x_{\max} = \max_{\forall n_1, \forall n_2} \{\mathbf{x}[n_1, n_2]\} \tag{2.2}$$

$$x_{\min} = \min_{\forall n_1, \forall n_2} \{\mathbf{x}[n_1, n_2]\} \tag{2.3}$$

where $x_{\min}$ and $x_{\max}$ are the lowest and highest pixel values currently present in the image. Figure 2.3 shows an example of face images before and after the normalization process.

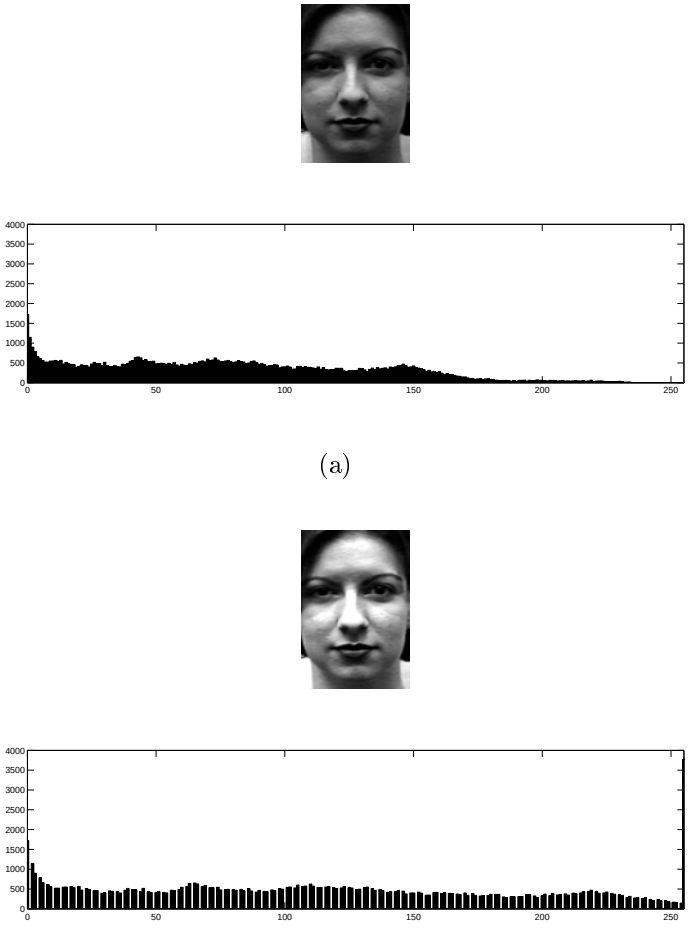


Figure 2.3: (a) Original face image with its histogram. (b) Normalized image with its histogram.

2.5 Feature Extraction

The two most common approaches to the facial feature extraction are the appearance-based methods and the geometric feature-based methods [39]. In appearance-based method, colour information about the image pixels of the face are used to infer the facial expression. In geometric feature-based systems, major face components and/or feature points are used for feature extraction. The appearance-based methods and geometry-based methods are presented in Section 2.5.1 and in Section 2.5.2, respectively.

2.5.1 Appearance-based Method

One of the methods to extract the features from images for automatic facial expression recognition system is the appearance-based approach. As stated earlier, these are methods that classify facial expressions based on the colour of the face pixels. Appearance-based algorithms are wide-ranging and include optical flow, principal component analysis (PCA), independent component analysis (ICA), local feature analysis (LFA), linear discriminant analysis (LDA) and image filters [40]. The following subsections present a number of them which are associated with FER systems.

Optical Flow

One of the earliest developed appearance-based methods of FER system was optical flow analysis [41, 42, 43]. Optical flow analysis endeavours to track object movement within an image by analyzing the change in pixel intensity of each image location $[n_1, n_2]$ over multiple frames in a time ordered sequence. The output of an optical flow computation for a particular image is a vector $[\mathbf{v}_{n_1}, \mathbf{v}_{n_2}]$ for each pixel in the input image; $\mathbf{v}_{n_1}$ and $\mathbf{v}_{n_2}$ represent the magnitudes of the image velocities in the n_1 and n_2 directions, respectively. The $\mathbf{v} = [\mathbf{v}_{n_1}, \mathbf{v}_{n_2}]$ vectors over multiple pixel locations can be combined into feature vectors and then classified as a particular facial expression. Feature

vectors based on optic flow can consist of the image velocities of certain fiducial points or of flow fields computed over entire image patches.

Principal Component Analysis

In appearance-based facial expression recognition systems, the fundamental unit of information is the pixel value. The features may be extracted from a pixel set by means of cropping, scaling and filtering. Even at low resolution, the number of pixels in a face image is on the order of hundreds. Moreover, many of the pixels in feature vector may contain little information which is useful for classifier. It is possible, for example, that pixels located in certain regions of the face may not change from one facial expression to another, thus rendering useless the corresponding coordinate of the feature vector. Another possibility is that one pixel value in the feature vector might be completely dependent on other pixels. In both cases, the feature vector contains redundant information and classification performance might improve by removing the superfluous components.

One approach to cope with the problem of excessive dimensionality is to reduce the dimensionality by linear combining features [44]. The feature compression is a linear method which projects the high-dimensional data onto a lower dimensional space. There are several approaches to finding effective linear transformations. Principal component analysis (PCA) seeks a projection that best represents the original data in a least-squares sense. Several appearance-based FER systems use principal component analysis (PCA) prior to expression classification [1, 12, 45]. Given that each feature vector, $\mathbf{z}$, is a column vector of M-Dimensions or M-elements, let us consider a set of N_f feature vectors, forming a feature matrix with one feature vector per column.

$$\mathbf{z} = \{\mathbf{z}[m,n] | 1 \leq m \leq M, 1 \leq n \leq N_f\} \tag{2.4}$$

The PCA [44, 46] can be used to find a linear transformation mapping the orig-

inal M-dimensional feature space into L-dimensional feature subspace, where normally $L << M$. The new feature vectors are defined by:

$$\mathbf{y} = \{\mathbf{y}[m', n] | 1 \leq m' \leq L, 1 \leq n \leq N_f\} \tag{2.5}$$

and

$$\mathbf{y} = \mathbf{w}_{\text{pca}}^T \mathbf{z} \tag{2.6}$$

where $\mathbf{w}_{\text{pca}}$ is the linear $M \times L$ transformation matrix, the superscript T denotes the transpose of a matrix and the columns of $\mathbf{w}_{\text{pca}}$ are the L eigenvectors associated with the L largest eigenvalues of the covariance matrix $\mathbf{c}$ which is given by

$$\mathbf{c} = \frac{1}{N_f} \sum_{n=1}^{N_f} (\mathbf{z}[m, n] - \mathbf{z}_\mu[m])(\mathbf{z}[m, n] - \mathbf{z}_\mu[m])^T \tag{2.7}$$

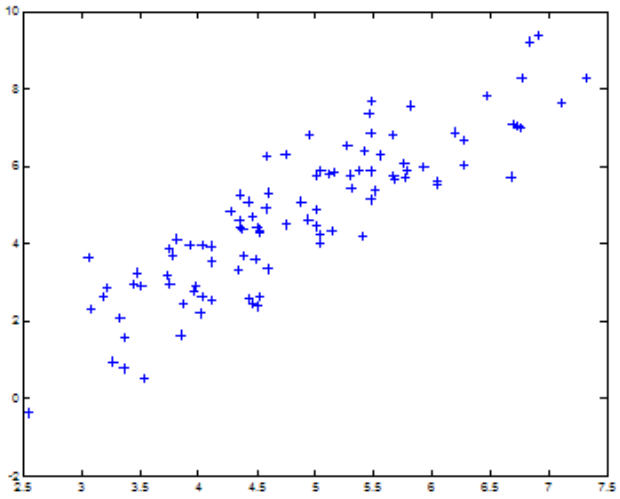
and

$$\mathbf{z}_\mu[m] = \frac{1}{N_f} \sum_{n=1}^{N_f} \mathbf{z}[m, n] \tag{2.8}$$

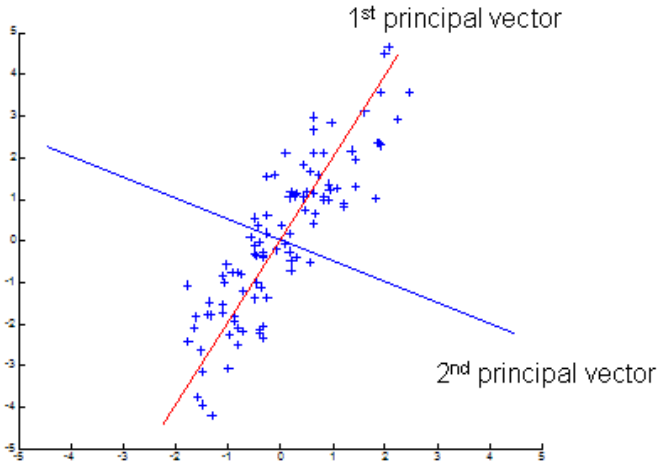
where $\mathbf{z}_\mu[m]$ for $1 \leq m \leq M$, is the mean of all feature vectors. Figure 2.4 shows an example of two principal components (PCs) for random data.

Eigenfaces

Eigenfaces refer to an appearance-based approach to face recognition that seeks to capture the variation in a collection of face images and uses this information to encode and compare images of individual faces in a holistic (as opposed to a parts-based or feature-based) manner. Specifically, the eigenfaces are the principal components (PCs) of a distribution of faces, or equivalently, the eigenvectors of the covariance matrix of the set of face images, where an image with N pixels is considered a point (or vector) in N-dimensional space. The idea of using principal components to represent human faces was developed by Sirovich and Kirby [47] and used by Turk and Pentland [48] for face detection and recognition.



(a)



(b)

Figure 2.4: (a) 2-D Random data. (b) Two principal component vectors.

The eigenface approach is considered by many to be the first working facial recognition technology [48] and it has served as the basis for one of the top commercial face recognition technology products. Since its initial development and publication, there have been many extensions to the original method and many new developments in automatic face recognition systems. Eigenface method is still often considered as a baseline method for comparison to demonstrate the minimum expected performance of such a system. The motivation behind eigenfaces is to find which features are important for classification and

which are not. This analysis reduces the dimensionality of the training set, leaving only those features that are critical for classification. Eigenvectors and eigenvalues are computed of the covariance matrix of the training images. The L highest eigenvectors are kept. Finally, the known individuals are projected into the face space and their weights are stored. The images are converted from two-dimensional (2-D) matrix of $N \times N$ pixels ($\mathbf{x}[n_1, n_2]$) into a one-dimensional (1-D) vector ($\mathbf{x}^f[\bullet]$) of size N^2. This conversion is necessary because a 2-D square matrix consisting of feature vectors is required to compute eigenvectors. The procedure for computing the eigenface or PCA of the face images is as follows:

- Find the mean of N_s training feature vectors

$$\mathbf{x}^f = \{\mathbf{x}^f[\bullet, i] | 1 \leq i \leq N_s\} \tag{2.9}$$

$$\mathbf{x}^f_\mu = \frac{1}{N_s} \sum_{i=1}^{N_s} \mathbf{x}^f[\bullet, i] \tag{2.10}$$

- Each training feature vector differs from the average feature vector by:

$$\mathbf{x}^f_d[\bullet, i] = \mathbf{x}^f[\bullet, i] - \mathbf{x}^f_\mu \tag{2.11}$$

- Compute a covariance matrix, $\mathbf{c}$, for the training set:

$$\mathbf{c} = \frac{1}{N_s} \sum_{i=1}^{N_s} \mathbf{x}^f_d[\bullet, i] \mathbf{x}^f_d[\bullet, i]^T = \frac{1}{N_s} \mathbf{A}\mathbf{A}^T \tag{2.12}$$

  where $N^2 \times N_s$ matrix $\mathbf{A} = [\mathbf{x}^f_d[\bullet, 1], \mathbf{x}^f_d[\bullet, 2], \cdots, \mathbf{x}^f_d[\bullet, N_s]]$.

- Find the eigenvectors, $\mathbf{u}_k$, of $\mathbf{c}$ that has maximum eigenvalues (λ_k) subject to Eq. (2.14):

$$\lambda_k = \frac{1}{N_s} \sum_{i=1}^{N_s} (\mathbf{u}_k^T \mathbf{x}^f_d[\bullet, i])^2 \tag{2.13}$$

$$\mathbf{u}_l^T \mathbf{u}_k = \begin{cases} 1 & \text{If } l = k \\ 0 & \text{Otherwise} \end{cases} \tag{2.14}$$

It is impractical to calculate the covariance based on Eq. (2.12). The major difficultly with this is that this computation will produce a large number of eigenvectors ($N^2 \times N^2$). For a 256×256 image that means that one must compute a 65536×65536 matrix and calculate 65536 eigenfaces. Computationally, this is not very efficient as most of those eigenfaces are not useful for classification. A novel method was introduced by Turk [48] based on the PCA to calculate the covariance for eigenfaces. The advantage of this method is that one only needs to evaluate N_s instead of N^2 elements. Usually, $N_s << N^2$ as only a few principal components (eigenfaces) will be relevant. The amount of calculations to be performed is reduced from the number of eigenvectors ($N^2 \times N^2$) to the number of feature vectors in the training set (N_s). Let $\mathbf{c}' = \mathbf{A}^T\mathbf{A}$, where $\mathbf{c}'$ is an $N_s \times N_s$ matrix and ν are N_s eigenvectors of $\mathbf{c}'$. The eigenfaces are given by

$$\mathbf{u}_l = \sum_{k=1}^{N_s} \nu_{lk} \mathbf{x}_d^f[\bullet, k] \tag{2.15}$$

where $\mathbf{u}_l$ for $1 \leq l \leq M$, are eigenfaces. The detail of eigenface method is explained in Appendix A.

Independent Component Analysis

In PCA, the projections of data along the principal components are uncorrelated, but they are not necessarily statistically independent. Hence, certain higher-order image dependencies such as facial lines may remain across the data dimensions even after PCA is performed [1]. Independent component analysis (ICA) is a technique for removing such dependencies from the input data set [49]. In contrast to PCA, the independent components of ICA are inherently unordered. Thus, when using ICA for dimension reduction of a fea-

ture set, a metric of ordering must be defined externally and then applied to the set of components. One possible metric is the class discriminability, defined as the ratio of the between class to within class variance of an independent component when applied to the training set [1].

2.5.2 Geometric-based Method

Many modern FER systems use the geometric positions of certain key facial points as well as these point's relative positions to each other as the input feature vector. Such FER systems are refereed as geometry-based systems. The key facial points whose positions are localized are known as fiducial points of the face. Typically, these face locations are located along the eyes, eyebrows, nose and mouth; however, some FER systems use dozens of fiducial points distributed over the entire face. The motivation for employing a geometry-based method is that facial expressions affect the relative position and size of various facial features and that by measuring the movement of certain facial points, the underlying facial expression can be determined. In order for geometric methods to be effective, the locations of these fiducial points must be determined precisely and, in real-time systems, they must also be found quickly. Various methods exist which can locate the face and its parts, such as elastic graph matching, Active Appearance Models (AAM) [50] and optical flow. Many FER systems require manual localization of the facial features for the first frame in a video sequence and thereafter, these points can be tracked automatically. Other approaches to fiducial point location is relocated them in each frame of the video. The exact type of feature vector that is extracted in a geometry-based FER systems depends on several parameters including the sensitivity of tracking points on the face, whether using 2-D or 3-D location of tracking point and converting a set of feature positions into the feature vector. Each of these parameters has its advantage and disadvantage. For location of tracking point, the advantage of 3-D fiducial point tracking is that the resulting FER systems are more robust to out-of-plane head rotation than 2-D systems.

The disadvantage is that these 3-D locations must usually be re-constructed from 2-D image data; the algorithms used to track fiducial points are thus more complex and slower. In terms of feature extraction [12], the most distinguishing factor in the design of geometry-based FER system is how the set of facial location vectors is converted into features. The simplest kind of feature vector in such systems contains either the relative positions of different facial landmarks or the displacements of the same feature points between frames in a video. In the former case, relative positions are often normalized by the face size to improve generalization performance across different human subjects. The following subsection presents a review of geometry-based FER systems based on Active Appearance Models.

Active Appearance Models

The Active Appearance Model (AAM) is a powerful method for matching a combined model of shape and texture for facial feature points detection [50]. In the AAM, the shape variation, texture variation and their interdependence are modelled using three linear systems respectively. The shape model, $\mathbf{x}_s$ and texture model, $\mathbf{x}_t$, can be illustrated as follows:

$$\mathbf{x}_s = \mathbf{x}_{\mu_s} + \mathbf{E}_s \mathbf{c}_s \tag{2.16}$$

$$\mathbf{x}_t = \mathbf{x}_{\mu_t} + \mathbf{E}_t \mathbf{c}_t \tag{2.17}$$

where $\mathbf{x}_s$ is the synthesized shape, $\mathbf{x}_t$ is the shape-free texture, $\mathbf{x}_{\mu_s}$ is the mean shape and $\mathbf{x}_{\mu_t}$ is the mean texture in a mean shaped patch. $\mathbf{E}_s$ and $\mathbf{E}_t$ are the matrices describing respectively, the modes of variation derived from the training set, $\mathbf{c}_s$ and $\mathbf{c}_t$ are the vectors controlling, respectively, the synthesized shape and the shape free texture.

To construct an AAM, a labelled training set is needed in which each image is accompanied by data specifying the coordinates of landmark points

around the main facial features. The appearance model is then obtained by constructing a shape model using the coordinate data and a texture model using both the image data and the coordinate data. After obtaining the shape and the texture models, a combined appearance model is obtained by applying principal component analysis (PCA) to the shape and the texture coefficients. The statistical model is then given by:

$$\mathbf{x}'_s = \mathbf{x}'_{\mu_s} + \Phi_s \mathbf{c}'_s \tag{2.18}$$

$$\mathbf{x}'_t = \mathbf{x}'_{\mu_t} + \Phi_t \mathbf{c}'_t \tag{2.19}$$

where Φ_s and Φ_t are truncated matrices describing the principal modes of combined appearance variations, which are derived from the training set. $\mathbf{x}'_{\mu_s}$ and $\mathbf{x}'_{\mu_t}$ are the mean shape and the mean texture of samples in the training set. $\mathbf{c}'_s$ and $\mathbf{c}'_t$ are the vector of appearance parameters controlling the shape $\mathbf{x}'_s$ and the texture $\mathbf{x}'_t$ simultaneously. Figure 2.5 shows the feature point in the face image around the main facial features.

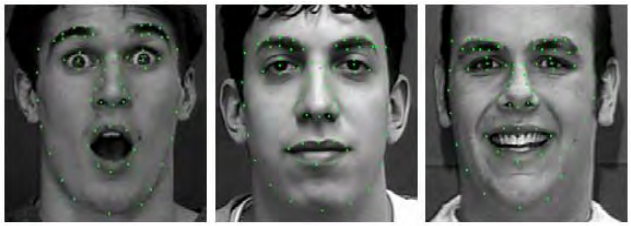


Figure 2.5: Feature location for different expressions. [2]

2.6 Feature Selection

The feature selection is the technique of selecting a subset of relevant features for building robust learning models. It has an effect on both the computational complexity and the quality of the classification results in terms of recognition

[2]Images from Cohn-Kanade database

rate and error rate. It is essential that the information contained in the selected features is sufficient to determine the input class correctly. Too many features may unnecessarily increase the complexity of the training and classification tasks, whereas poor selection of features may have a detrimental effect on the classification results. Feature selection methods that are adequate for simple distributions of patterns belonging to different classes fail in classification tasks with more complex distributions and overlapping boundaries. There are two general approaches to feature selection: filters and wrappers [20, 51].

The filter based approaches aim at ranking features or feature subsets independently of the predictor (classifier). Therefore, the classification results are not taken into account during the selection process. The wrapper-based methods on the other hand, select an optimal sub-set of features by minimizing the classification error [20]. Wrapper methods typically require extensive computation to search for the best features.

The hybrid approaches use objective criteria that take into account the inter-class and intra-class variations as well as the classification results. The filter-based approaches are of the least computational complexity, whereas the wrapper and the hybrid methods are more complex and more computationally demanding. The literature has shown no clear superiority of any particular feature selection method [52], however, any non-exhaustive search method does not guarantee to find the optimal feature set and usually provides a reasonable local optimum. Methods based on measures such as correlation assume linear dependencies between data and cannot handle arbitrary relations between the pattern coordinates and the different classes. Commonly used data reduction techniques are not invariant under linear transformations such as data scaling used in the pre-processing stage. Different feature reduction and selection methods, including the PCA [46], the Mutual Information (MI) [53] and global optimization algorithms, i.e., genetic algorithm (GA), ant colony optimization (ACO), has been used for feature selection [20].

2.7 Classification

A wide range of classifiers have been applied to the automatic expression recognition problem [43, 54, 55, 56]. K-Nearest Neighbours (KNN), Bayesian Classifier, Multi-class LDA classifier, support vector machine (SVM), neural network (NN) are widely used in statistical learning for classification. The KNN classifier and SVM classifier are presented in the following subsections.

2.7.1 K-Nearest Neighbour Classifier

The K-Nearest Neighbours method is a classical classification algorithm where the input feature vector is classified based on the class represented by the majority of the K nearest feature vectors obtained during the training process [57]. Given an input feature vector, the algorithm finds K closest feature vectors representing different expressions. The Euclidean vector distance measure is used [Eq. (2.20)]. The expression represented by the majority of the K nearest feature vectors is assigned to the input vector. The Euclidean distance ($||\mathbf{x}^f - \mathbf{y}^f||$) between input feature vector ($\mathbf{x}^f = [x_1^f, x_2^f, \cdots, x_n^f]^T$) and the training feature vector ($\mathbf{y}^f = [y_1^f, y_2^f, \cdots, y_n^f]^T$) is defined by

$$||\mathbf{x}^f - \mathbf{y}^f|| = \sqrt{\sum_{i=1}^{n}(x_i^f - y_i^f)^2} \tag{2.20}$$

2.7.2 Support Vector Machine

Improving classifier effectiveness has been an area of intensive machine learning research over the last two decades and this work has led to a new generation of state-of-the-art classifiers, such as support vector machines, boosted decision trees, regularized logistic regression, neural networks and random forests [44, 58, 59].

Many of these methods, including support vector machines, have been applied with success to pattern classification problem, particularly facial expres-

sion classification [56]. An SVM is a kind of large margin classifier. It is a vector space based machine learning method where the goal is to find a decision boundary between two classes that is maximally far from any point in the training data [60]. SVM performs classification by constructing an N-dimensional hyperplane that optimally separates the data into two categories [59]. A decision hyperplane can be defined by an intercept term, z and a decision hyperplane normal vector $\mathbf{d}$ which is perpendicular to the hyperplane. This vector is commonly referred to in the machine learning literature as the weight vector [61]. Let $\mathcal{X}$ be a set of N_s labelled training set

$$\mathcal{X} = \{(\mathbf{x}_i^f, c_i) | c_i \in \{-1, 1\}, 1 \leq i \leq N_s\} \tag{2.21}$$

where $\mathbf{x}_i^f$ are the training samples and c_i is the class label of $\mathbf{x}_i^f$. The linear classifier ($h(\mathbf{x}^f)$) typically try to find a decision function given by

$$h(\mathbf{x}^f) = sgn(\langle \mathbf{d}, \mathbf{x}^f \rangle + z) \tag{2.22}$$

and

$$sgn(\tau) = \begin{cases} 1 & \tau \geq 0 \\ -1 & \tau < 0 \end{cases} \tag{2.23}$$

where $\mathbf{d}$ is the decision hyperplane normal vector and z is intercept term. $h(\mathbf{x}^f) \in \{-1, 1\}$ yields a label (-1 indicates one class and $+1$ the other class) for an unseen example $\mathbf{x}^f$. The SVM linear classifier relies on a dot product between data point vectors. Let $K(\mathbf{x}_i^f, \mathbf{x}_j^f) = \mathbf{x}_i^{f^T} \mathbf{x}_j^f$, the SVM classifier ($h_{\text{svm}}$) is given by

$$h_{\text{svm}}(\mathbf{x}^f) = sgn(\sum_{i=1}^{N_s} \alpha_i c_i K(\mathbf{x}_i^f, \mathbf{x}^f) + z) \tag{2.24}$$

where the α_i are Lagrange multipliers of a dual optimization problem [59].

It is possible to show that only some of the α_i are non-zero in the optimal solution, namely those arising from training points nearest to the hyperplane, called support vectors. These induce sparseness in the solution and give rise to efficient approaches to optimization. Once a decision function is obtained, classification of an unseen example, $\mathbf{x}^f$, amounts to checking on what side of the hyperplane the example lies. SVMs perform an implicit embedding of data into a high dimensional feature space, where linear algebra and geometry may be used to separate data that is only separable with nonlinear rules in input space. To do so, the learning algorithm is formulated to make use of kernel functions, allowing efficient computation of inner products directly in feature space, without need for explicit embedding. A kernel function (K) is such a function that corresponds to a dot product in some expanded feature space. The kernel is defined by

$$K(\mathbf{x}_i^f, \mathbf{x}_j^f) = \langle \phi(\mathbf{x}_i^f)^T \cdot \phi(\mathbf{x}_j^f) \rangle \tag{2.25}$$

where ϕ is a nonlinear mapping function which embeds input vectors into feature space. Using a kernel function, SVM is an alternative training method for polynomial, radial basis function (RBF) and multi-layer perceptron (MLP) neural network classifiers in which the weights of the network are found by solving a quadratic programming problem with linear constraints, rather than by solving a non-convex, unconstrained minimization problem as in standard neural network training [60], [61]. The most common form of RBF is normally a Gaussian distribution given by

$$K(\mathbf{x}_i^f, \mathbf{x}_j^f) = \exp[\frac{||\mathbf{x}_i^f - \mathbf{x}_j^f||^2}{2\sigma^2}] \tag{2.26}$$

where $||\bullet||$ is the Euclidean distance between two vectors. The output of the kernel is dependent on the Euclidean distance of $\mathbf{x}_j^f$ from $\mathbf{x}_i^f$ (one of these will be the support vector and the other will be the testing data point). The support vector will be the centre of the RBF and σ will determine the area of

influence this support vector has over the data space. A larger value of σ will give a smoother decision surface and more regular decision boundary. This is because an RBF with large σ will allow a support vector to have a strong influence over a larger area. Furthermore, a larger σ value also increases the α_i value in Eq. (2.24) for the classifier. When one support vector influences a larger area, all other support vectors in the area will increase in α_i value to counter this influence. Therefore, all α_i values will reach a balance at a larger magnitude. In addition, a larger α_i value will also reduce the number of support vectors. Since each support vector can cover a larger space, fewer are needed to define a boundary.

2.8 Image Databases

There are few databases available for expression recognition. As mentioned in Section 2.2, there are two ways to describe facial expressions. Available databases can be categorized into two classes according to the description they used. In one group expressions are coded in FACS, while in the other group images are labelled by their prototypic emotional expressions.

2.8.1 JAFFE Database

The Japanese Female Facial Expression (JAFFE) database [18] contains 213 images of 6 basic facial expressions: anger, disgust, fear, happy, sad and surprise, as well as the neutral expression. The images were taken from 10 Japanese female models. The expressions expressed by each picture were subjectively tested on 60 Japanese volunteers. Figure 2.6 shows an example of images from JAFFE database.

2.8.2 Cohn-Kanade Database

Cohn-Kanade database [13] included 388 image sequences from 100 subjects. Each sequence contained 12-16 frames. The subject's ages ranged from 18 to

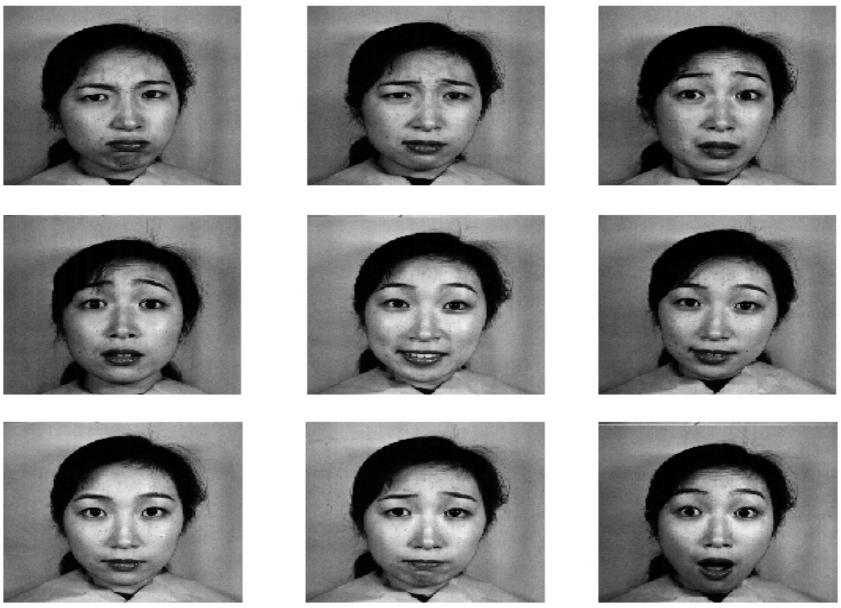


Figure 2.6: Static images from JAFFE database. [3]

30 years. Sixty-five percent of subjects were female and thirty five percent were male. Fifteen percent of subjects come from the African-American background and three percent from the Asian or the Latino-American background. The image sequences represented 100 different subjects expressing different stages of an expression development, starting from a low arousal stage, reaching a peak of arousal and then declining. The facial expressions of each subject represented six basic expressions: anger, disgust, fear, happy, sad and surprise. Some subjects did not have image sequences corresponding to all of the six expressions and in some cases, only one image sequence per expression was available. An example of image sequences has been shown in Figure 2.7.

[3]Images from JAFFE database [18].

Figure 2.7: Image sequences from Cohn-Kanade database. [4]

2.8.3 BU-3DFE Database

3-D facial models have been extensively used for 3-D face recognition and 3-D face animation, the usefulness of such data for 3-D facial expression recognition is unknown. The 3-D facial expression database called BU-3DFE database is used to foster the research in expression recognition field [62]. The BU-3DFE database includes 100 subjects with 2500 facial expression models. The database presently contains 100 subjects (56% female, 44% male), with age ranging from 18 years to 70 years old, with a variety of ethnic/racial ancestries, including White, Black, East-Asian, Middle-east Asian, Indian and Hispanic

[4] Images from Cohn-Kanade database [13].

Latino. Participants in face scans include undergraduates, graduates and faculty from departments of Psychology, Arts and Engineering. The majority of participants were undergraduates from the Psychology Department. Each subject performed seven expressions in front of the 3-D face scanner. With the exception of the neutral expression, each of the six prototypic expressions includes four levels of intensity. Therefore, there are 25 instant 3-D expression models for each subject, resulting in a total of 2500 3-D facial expression models in the database. Associated with each expression shape model is a corresponding facial texture image captured at two views (about $+45°$ and $-45°$). As a result, the database consists of 2500 two-view's texture images and 2500 geometric shape models. Figure 2.8 shows an example images of this database.

Figure 2.8: Six facial expression images from BU 3-D database. [5]

2.9 Summary

A brief literature survey is conducted in this chapter on FER system modules. Each module is explained in detail and then an overview of the development in this area was given. For image pre-processing, the image normalization technique is described in Section 2.4 for gray-scale images. Section 2.5 presented both appearance-based and geometric-based methods for feature extraction. For appearance-based, the optical flow, PCA and ICA are explained and AAM is described for geometric-based method. The geometric-based methods are more complex than appearance-based methods in terms of implementing for

[5] Images from BU 3-D database [62].

FER system. Furthermore, the feature points need to be set manually before using AAM. In Section 2.6, the feature selection is briefly explained and the detail will be in Chapter 5. The two classifiers, KNN and SVM are presented in Section 2.7. At the end, the face databases for expression recognition are presented with full detail in Section 2.8. Next Chapter will explain the system architecture and image pre-processing module.

Chapter 3

Facial Image Pre-processing

3.1 Introduction

Pre-processing of face images prior to image classification is essential [12]. Face images from different databases have diverse resolutions, backgrounds and are captured under varying illumination. Pre-processing module is used to make the images more comparable. In FER system presented here, this module consists of following components: face detection, facial detection, face component detection and normalization. The Viola-Jones method based on the Haar-like features and AdaBoost learning algorithm is performed for face detection [25]. Furthermore, A novel face detection method based on morphological method is proposed based on Sobel operator and blob analysis [19]. Facial detection component is introduced to detect the face with maximum intensity of expression in sequence images. The face template is used to detect the parts of the face such as eyes and mouth. The last part of image pre-processing module is normalized the face images to same size and scale. In the following sections, every component of image pre-processing is described in detail.

3.2 Face Detection

The Viola-Jones method and proposed morphological method for face detection are explained in the following sections.

3.2.1 Viola-Jones Method

The AdaBoost-based face detector by Viola and Jones [25] demonstrated that faces can be fairly reliably detected in real time (i.e., more than 15 frames per second on 320×240 image pixels with desktop computers) under partial occlusion. While Haar wavelets were used in [33] for representing faces and pedestrians, they proposed the use of Haar-like features which can be computed efficiently with integral image [25]. Haar-like features that can directly represent the local relation between the image regions perform efficiently in face detection. Figure 3.1 shows an example of Haar-like features that are used to encode different intensity information of face images at different position and scale.

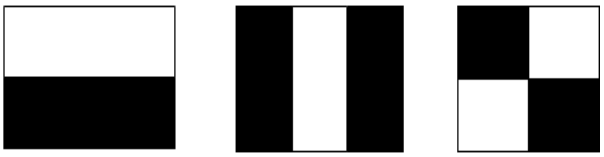


Figure 3.1: Three types of 2-D Haar-like features.

The real value of Haar-like features can be computed in the way of the pixel sum of the black rectangular region (B_i) subtracted from that of the white region (W_j) [Eq. (3.1)] and the pixel sum of each rectangular region feature can be obtained by integral images. Figure 3.2(b) illustrates the computation process using integral method.

$$\mathrm{V_h} = \sum_{\mathrm{i=1}}^{\mathrm{N_b}} (\mathrm{B_i}) - \sum_{\mathrm{j=1}}^{\mathrm{N_w}} (\mathrm{W_j}) \tag{3.1}$$

where V_h is feature value, N_b and N_w are the numbers of pixels in black and white areas, respectively. The integral image, denoted $\mathbf{x}_{\mathrm{sum}}[n_1, n_2]$, at location $[n_1, n_2]$ contains the sum of the pixel values above and to the left of $[n_1, n_2]$ as shown in Figure 3.2(a).

$$\mathbf{x}_{\mathrm{sum}}[n_1, n_2] = \sum_{n_1=1}^{N_1'} \sum_{n_2=1}^{N_2'} \mathbf{x}_{\mathrm{Original}}[n_1, n_2] \tag{3.2}$$

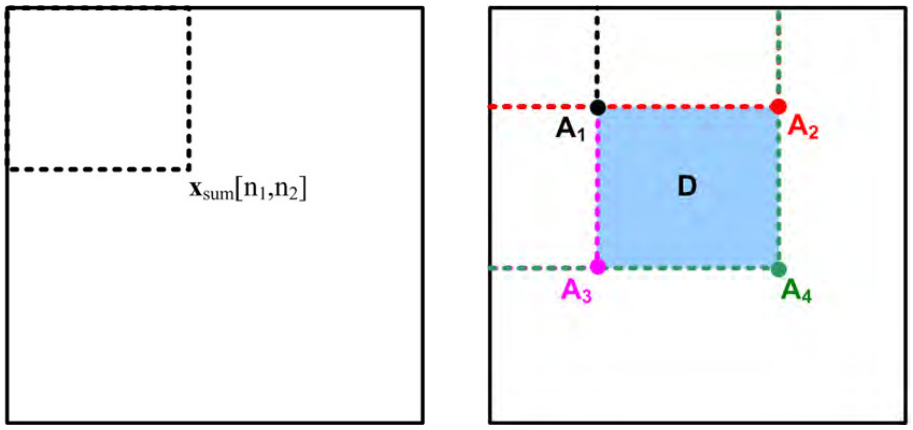


Figure 3.2: (a) The upright integral image. (b) Calculation scheme of the pixel sum of upright rectangle feature.

where $\mathbf{x}_{\text{Original}}[n_1, n_2]$ is the pixel value of input image $\mathbf{x}_{\text{Original}}$. The integral image can be computed in one-pass over the image using the following recurrence relation [25]:

$$\mathbf{x}_{\text{cum}}[n_1, n_2] = \mathbf{x}_{\text{cum}}[n_1, n_2 - 1] + \mathbf{x}_{\text{Original}}[n_1, n_2] \tag{3.3}$$

$$\mathbf{x}_{\text{sum}}[n_1, n_2] = \mathbf{x}_{\text{sum}}[n_1 - 1, n_2] + \mathbf{x}_{\text{cum}}[n_1, n_2] \tag{3.4}$$

where $\mathbf{x}_{\text{cum}}[n_1, n_2]$ denotes the cumulative row sum and

$$\mathbf{x}_{\text{cum}}[n_1, 0] = 0, \mathbf{x}_{\text{sum}}[0, n_2] = 0 \tag{3.5}$$

Given the integral image, the sum of pixel values within a rectangular region of the image aligned with the coordinate axes can be computed with four array references (i.e., constant time). For example, to compute the sum of region D in Figure 3.2(b), the following four references are required: $A4+A1-(A2+A3)$.

Given a sample image of 24×24 pixels, the exhaustive set of parameterized Haar-like features (at different position and scale) is very large (about $160,000$). Contrary to most of the prior algorithms that use one single strong classifier (e.g., neural networks and support vector machines [23, 37]), they

used an ensemble of weak classifiers where each one is constructed by thresholding of one Haar-like feature. The weak classifiers are selected and weighted using the AdaBoost algorithm. Since there is a large number of weak classifiers, a method is presented to rank these classifiers into several cascades using a set of optimization criteria. Within each stage, an ensemble of several weak classifiers is trained using the AdaBoost algorithm.

Let $\mathbf{x}_i^h$ denotes the samples including face and non-face images. The AdaBoost learning algorithm is shown in Algorithm 1. A set of N_s labelled training examples is given as $\{(\mathbf{x}_1^h, y_1), \cdots, (\mathbf{x}_{N_s}^h, y_{N_s})\}$, where $y_i \in \{+1, -1\}$ is the class label associated with i^{th} example, $\mathbf{x}_i^h$, for $1 \leq i \leq N_s$. $D_t(i)$ is a weight of i^{th} example $\mathbf{x}_i^h$. The weights are initialized by $D_1(i) = 1/N_s$. The final strong classifier $H(\mathbf{x}^h)$ is a linear combination of weak classifiers:

$$H(\mathbf{x}^h) = sgn(\sum_{t=1}^{N_c} \alpha_t h_t(\mathbf{x}^h)) \tag{3.6}$$

and

$$\alpha_t = \frac{1}{2}\log(\frac{1-\epsilon_t}{\epsilon_t}) \tag{3.7}$$

where $h_t(\mathbf{x}^h)$ is the weak classifier, N_c is the number of classifiers, $\alpha_t \in \mathbb{R}$ and ϵ_t is the weighted error rate of classifier $h_t(\mathbf{x}^h)$

The motivation behind the cascade of classifiers is that simple classifiers at early stage can filter out most negative examples efficiently and stronger classifiers at later stage are only necessary to deal with instances that look like faces. The final detector, a 38 layer cascade of classifiers with 6,060 Haar-like features, demonstrated impressive real-time performance with fairly high detection and low false positive rates. Several extensions to detect faces in multiple views have been proposed [36]. An implementation of the AdaBoost-based face detector [34] can be found in the Intel OpenCV library. Despite the excellent run-time performance of boosted cascade classifier, the training time of such a system is rather lengthy [25]. In addition, the classifier cascade is an example of degenerate decision tree with an unbalanced data set (i.e., a small

Algorithm 1 Learning procedure based on AdaBoost

Input:
Given example images $(\mathbf{x}_i^h, y_i), \cdots, (\mathbf{x}_{N_s}^h, y_{N_s})$, $y_i \in \{+1, -1\}$ for face and non-face examples respectively.
Initialization:
$D_t(i) = 1/N_s$
for $t = 1$ to N_c **do**
1. For each feature, calculate a feature value (V_h).
2. Train a weak classifier based on a combination of features. The error is evaluated with respect to $D_t(i)$,
$\epsilon_t = \sum_{i=1}^{N_s} D_t(i); \quad y_i \neq h_t(\mathbf{x}_i^h)$
3.Choose optimal h_t to minimize the error.
4.Update the weights:
$D_{t+1}(i) = \frac{D_t(i)\exp(-\alpha_t y_i h_t(\mathbf{x}_i^h))}{\sum_i D_t(i)\exp(-\alpha_t y_i h_t(\mathbf{x}_i^h))}$
end for
Output: The final strong classifier based on Eq. (3.6)

set of positive examples and a huge set of negative ones). Numerous algorithms have been proposed to address these issues and extended to detect faces in multiple views. To handle the asymmetry between the positive and negative data sets, Viola and Jones proposed the asymmetric AdaBoost algorithm [34] which keeps most of the weights on the positive examples. Figure 3.4 illustrate the cascade classifier for face detection based on Viola-Jones method.

Figure 3.3: Two main rectangular Haar-like features for face detection

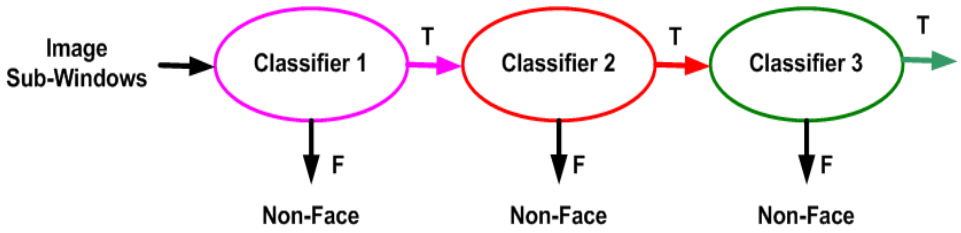


Figure 3.4: Cascade weak classifiers for face detection

3.2.2 Morphological Method

The databases which are used for this investigation are well-defined. This means that face images have already been centred and backgrounds of images are plain without any texture. Therefore, edge detection and blob analysis can be used to detect the face. Several kind of filters are used to find the gradient image. Sobel filter is less complex than other filters like Canny filter and also more practical. In addition, the time process for Sobel filter is 10 times faster than Canny filter [19, 58].

The parts of images that contained features of the face images and other areas are extracted using the Sobel operator [19, 44]. The Sobel operator consists of a pair of 3×3 convolution kernels shown in Figure 3.5. The kernel $\mathbf{G}_y$ is kernel $\mathbf{G}_x$ rotated by 90 deg.

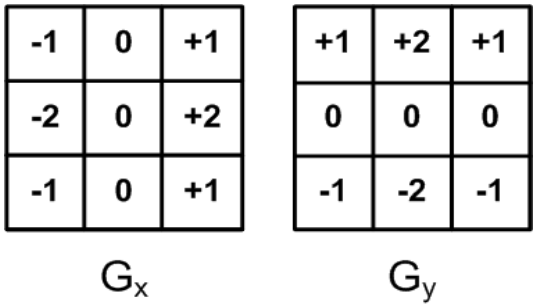


Figure 3.5: A pair of the Sobel convolution kernels.

The Sobel kernels were designed to respond maximally to edges running vertically and horizontally relative to the pixel grid. In Figure 3.5, kernel $\mathbf{G}_x$ shows the highest sensitivity along the horizontal direction, whereas, kernel

$\mathbf{G}_y$ has the highest sensitivity along the vertical direction. The kernels can be applied separately to the input image, to produce separate measurements of the gradient component $\mathbf{G}_x$ and $\mathbf{G}_y$ in each orientation, respectively. These can then be combined together to find the absolute magnitude of the gradient at each point and the orientation of that gradient. The gradient magnitude of a given image, $\mathbf{x}_{\text{Original}}[n_1, n_2]$, is defined by:

$$|\mathbf{x}_{\text{Original}_{\text{G}}}| = \sqrt{\mathbf{x}^2_{\text{Original}_{\text{Gx}}} + \mathbf{x}^2_{\text{Original}_{\text{Gy}}}} \tag{3.8}$$

where $\mathbf{x}_{\text{Original}_{\text{Gx}}}$ and $\mathbf{x}_{\text{Original}_{\text{Gy}}}$ are the horizontal and vertical derivative approximations given by

$$\mathbf{x}_{\text{Original}_{\text{Gx}}} = \mathbf{G}_x * \mathbf{x}_{\text{Original}}, \quad \mathbf{x}_{\text{Original}_{\text{Gy}}} = \mathbf{G}_y * \mathbf{x}_{\text{Original}} \tag{3.9}$$

where $(*)$ denotes the 2-D convolution operation. After finding the gradient of the image, the face is founded based on blob analysis [19, 63]. In image processing, a blob is defined as a region of connected pixels. The blob analysis algorithm identifies these regions in an image and places them in one of two categories: the foreground (typically pixels with a non-zero value) or the background (pixels with a zero value).

The edge of the images are not clear and they will be missed in further process. To overcome this problem, the features of the gradient image are exaggerated based on Dilation operator [63]. Dilation is a morphological operation used to enhance the features of an image. The basic effect of this operator on a binary image is to gradually enlarge the boundaries of regions of foreground pixels. Thus areas of foreground pixels grow in size while holes within those regions become smaller. The part representing a face (foreground) was cropped automatically from the image. The foreground area of exaggerated features are then filled and for each area the numbers are allocated. Finally the part representing a face (foreground) is cropped automatically from the image based on threshold. Figure 3.6 is shown the process for face detection.

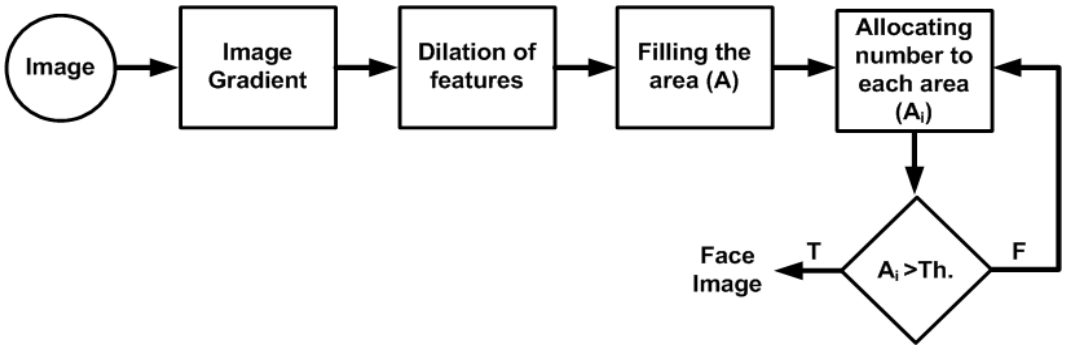


Figure 3.6: Face detection based on Sobel operator and blob analysis.

3.3 Facial Detection

One of the stages for pre-processing of a sequence of images involves detection of an image which depicts certain expression with the maximum level of arousal (expression intensity). The novel method is proposed which is called facial detection using the inter-frame mutual information criterion [11]. For each frame, the mutual information between the current frame and the initial frame is calculated and the frame with the minimum mutual information is selected as the frame that represents an expression with the maximum arousal [Eq. (3.10)].

$$\mathbf{F}_{\text{intensity}} = \arg\min_{\mathbf{F}_j}\{I(\mathbf{F}_{\text{initial}}; \mathbf{F}_j)\} \tag{3.10}$$

where $\mathbf{F}_{\text{initial}}$ is an initial frame which includes a normal face image without any expression, $\mathbf{F}_j$ denotes other frames with different expressions and $I(\mathbf{F}_{\text{initial}}; \mathbf{F}_j)$ represents the mutual information between initial frame and other frames. The facial detection component is only used for sequence of images.

3.4 Face Component Detection

In this thesis, the part of the facial image which covers the entire face is used for the FER system. This section explains how to find the face regions. For face region location, face model [64] is used to extract the eyes and mouth from the face image. The location of eyes and mouth and the sample of eyes

and mouths images are shown in Figure 3.7 and Figure 3.8.

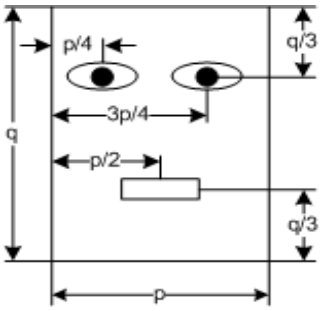
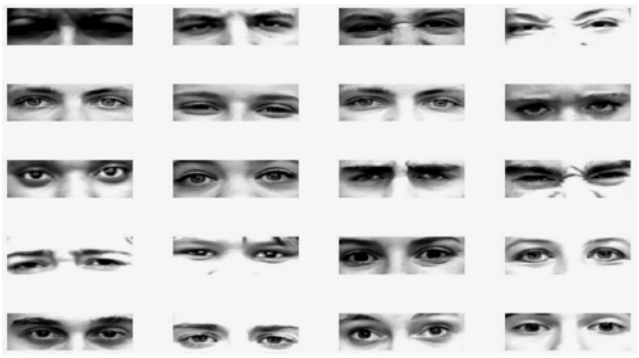


Figure 3.7: Face template used for eyes and mouth detection. [6]



(a)

(b)

Figure 3.8: (a) Eye images generated by face model. (b) Mouth images generated by face model. [7]

[6] Face model in [64].

[7] Original images from Cohn-Kanade database [13].

3.5 Gray-scale Facial Image Normalization

In the last step of image pre-processing, the face-only images are normalized using Eq. (2.1) to eliminate the illumination effect [19]. Furthermore, the face images are scaled to a normalized size for different experiment. Figure 3.9 shows an example of images after pre-processing stage.

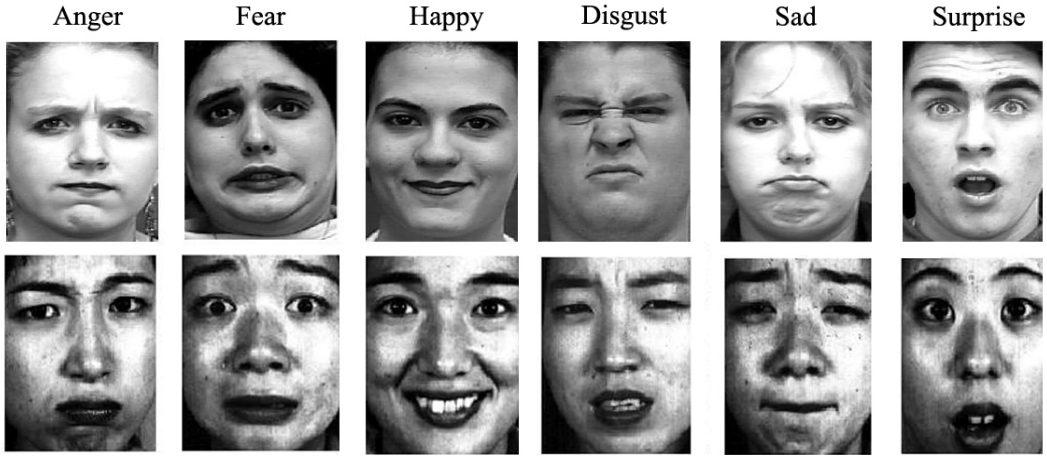


Figure 3.9: Face images after pre-processing step.

3.6 Colour Facial Image Pre-processing

Generally, the image pre-processing procedure is a vital stage in pattern recognition and computer vision [19]. For colour images, the pre-processing step attempt to reduce the presence of illumination and to generate tensor representation of colour image based on multi-linear image analysis.

3.6.1 Multilinear Analysis

Images are generated from the interaction of multiple factors related to scene structure, illumination and imaging. Facial images are determined by facial geometry (person, expression), the pose of the head relative to the camera, the lighting conditions and the camera employed. Linear methods can not present the multi-factor image ensembles completely. To overcome this issue,

the nonlinear methods have been performed based on multi-linear algebra [65]. Multilinear algebra involves the natural generalization of matrices. Whereas matrices are linear operators defined over a vector space, these generalizations, referred to as tensors, define multi-linear operators over a set of vector spaces. Hence, the algebra of higher-order tensors, subsumes linear algebra, matrices, vectors and scalars as a special case. Multilinear algebra serves as a unifying mathematical framework suitable for addressing a variety of challenging problems in image science and visual computing. The multi-linear algebraic framework can be applied to the synthesis, analysis and recognition of images. Within this mathematical framework, the image ensemble of interest is represented as a higher-order tensor, which must be decomposed in order to separate and parsimoniously represent the constituent factors. A tensor is a higher-order generalization of a matrix, vector and scalar. Basically, tensors can be considered as a multidimensional array of numbers, which are known as "components". The entries of such an array are symbolically denoted by the name of the tensor with indices giving the position in the array. The total number of indices is equal to the dimension of the array and is called the order or the rank of the tensor. The components of an order n tensor $\mathcal{T}$ would be denoted $\mathcal{T} \in \mathbb{R}^{\prod_1^n N_n}$, i.e. the different dimensions of the array are called modes (n). Just like the components of a vector change when the basis of the vector space is changed, the components of a tensor also change under such a transformation. The tensor can be seen as generalizations of vectors (first order tensors) and matrices (second order tensors). It is common to unfold a tensor along any of its $n' \leq n$ dimensions to obtain a mode-n matrix version of the tensor. For tensor order $n = 3$ and $\mathcal{T} \in \mathbb{R}^{\prod_1^3 N_n}$, the three possible unfolding matrices are as follow:

$$\mathcal{T} \in \mathbb{R}^{\prod_1^3 N_n} \rightarrow T^{(n'=1)} \in \mathbb{R}^{N_1 \times (N_2 \times N_3)} \tag{3.11}$$

$$\mathcal{T} \in \mathbb{R}^{\prod_1^3 N_n} \rightarrow T^{(n'=2)} \in \mathbb{R}^{N_2 \times (N_1 \times N_3)} \tag{3.12}$$

$$\mathcal{T} \in \mathbb{R}^{\prod_1^3 N_n} \rightarrow T^{(n'=3)} \in \mathbb{R}^{N_3 \times (N_1 \times N_2)} \tag{3.13}$$

3.6.2 Colour Image Analysis

A colour image is an image which includes colour information for each pixel. For each pixel, it is almost sufficient to provide three colour channels. These colour channels carry three components of primary colour in a digital image that make up a full colour image. There are several colour channels in image processing. The RGB image has three channels: red, green and blue. RGB channels roughly follow the colour receptors in the human eye and are used in computer displays and image scanners [19]. The RGB colour is an additive colour in which red, green and blue colours are added together in various ways to reproduce a broad array of colours. In this study, the RGB channels are used to make the 2-D matrix of tensor for each image. A well-known image intensity normalization method is used to remove the effect of illumination and lighting in the image. Given an input image of $N_1 \times N_2$ pixels represented in the *RGB* colour space, $\mathbf{x}[n_1, n_2, n_3]$, the normalized values, $\mathbf{x}_{\text{norm}}[n_1, n_2, n_3]$, are defined by:

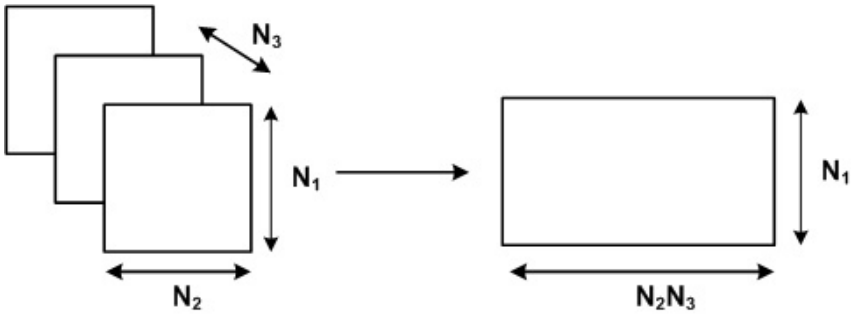
$$\mathbf{x}_{\text{norm}}[n_1, n_2, n_3] = \frac{\mathbf{x}[n_1, n_2, n_3]}{\sum_{n_3=1}^{3} \mathbf{x}[n_1, n_2, n_3]} \tag{3.14}$$

where $\mathbf{x}_{\text{norm}}[n_1, n_2, n_3]$ for $n_3 = 1, 2, 3$ corresponding to red, green and blue (or R, G and B) components of the image, $\mathbf{x}[n_1, n_2, n_3]$, respectively, $1 \leq n_1 \leq N_1$, and $1 \leq n_2 \leq N_2$. It is obvious that

$$\sum_{n_3=1}^{3} \mathbf{x}_{\text{norm}}[n_1, n_2, n_3] = 1 \tag{3.15}$$

After normalization, the image is unfolded to 2-D matrix tensor [65, 66]. A tensor is a higher-order generalization of a vector (first order tensor) and a

matrix (second order tensor). Tensors are multi-linear mappings over a set of vector spaces. Given a colour image $\mathbf{x}[n_1, n_2, n_3]$, with three colour channels ($n_3 = 3$), the unfolded of this image can be in three different dimensions. In this study, The image $\mathbf{x}_{N_1 \times N_2 \times N_3}$ is unfolded to $\mathbf{x}_{N_1 \times N_2 N_3}$ which is called horizontal unfolding. Also the image can be unfolded to $\mathbf{x}_{N_1 N_3 \times N_2}$ which is called vertical unfolding. Figure 3.10 shows the procedure of unfolding of colour facial image. The features are extracted from these unfolded facial images.



(a)

(b)

Figure 3.10: Horizontal unfolding of facial expression image.

3.7 Image Noise

Image noise is a random, usually unwanted, variation in brightness or colour information in an image [19]. It is also an important issue in pattern recognition and computer vision. Image transmission noise may be caused by various sources, such as car ignition systems, industrial machines in the vicinity of the receiver, switching transients in power lines, lightning in the atmosphere and

various unprotected switches. This type of transmission noise is often modelled as the impulse noise [67, 68]. The impulse noise can also be introduced into images during acquisition of the images. For example, the impulse noise may be introduced during fingerprint acquisition in real-life border security check. The two most common impulse noise types are fixed-value impulse noise (also known as the salt-and-pepper noise) and random-value impulse noise [67, 68, 69]. The following subsection describes the impulsive noise which is used for facial expression recognition.

3.7.1 Salt-and-pepper Noise

The fixed value impulsive noise is sometimes called salt-and-pepper noise or spike noise. The salt-and-pepper impulse noise can be very well dealt with by the rank-order statistics values of an image than by many other linear statistics values. An image containing salt-and-pepper noise will have dark pixels in bright regions and bright pixels in dark regions. This type of noise can be caused by dead pixels, analogue-to-digital converter errors, bit errors in transmission. Salt-and-pepper noise is a model adopted frequently to simulate impulsive noise in synthetic images. Let $\mathbf{x}[n_1, n_2, n_3]$ represent the *RGB* (Red, Green, Blue) pixel values of a colour image, where $0 \leq n_1 \leq N_1$, $0 \leq n_2 \leq N_2$, $1 \leq n_3 \leq 3$ and N_1 and N_2 are the hight and the width of the image respectively. There are two approaches reported in the literature to model the impulse noise for colour image. The impulse noise corruption of colour images in the *RGB* space is expressed by one of these multivariate models [67, 68].

$$\mathbf{x}_{sp}[n_1, n_2] = \begin{cases} \mathbf{s}[n_1, n_2] & \text{with probability } 1 - p_I \\ \mathbf{n}_T[n_1, n_2] & \text{with probability } p_I \end{cases} \tag{3.16}$$

$$\mathbf{x}_{sp}[n_1, n_2] = \begin{cases} \mathbf{s}[n_1, n_2] & \text{with probability } (1-p)^3 \\ \mathbf{n}_t[n_1, n_2] & \text{with probability } 1-(1-p)^3 \end{cases} \tag{3.17}$$

where $\mathbf{s}[n_1, n_2]$ and $\mathbf{x}_{sp}[n_1, n_2]$ represent the original and the observed pixel values respectively and the value of $\mathbf{n}_T[n_1, n_2]$ or $\mathbf{n}_t[n_1, n_2]$ is generated by substituting at least one colour component of the pixel $s[n_1, n_2]$ by a distinct value d in both Eq. (3.16) and Eq. (3.17). In Eq. (3.16), p_I is the impulse noise ratio. If at least one of the three components of the pixel is corrupted by the impulse noise, its remaining noise-free components will have a 50% probability to be corrupted [69]. The second approach Eq. (3.17) is a more generalized impulse noise model of colour images, where $p = p_r = p_g = p_b$ is the impulse noise ratio for each channel of a corrupted colour image, assuming that the image is corrupted by the impulse noise in a channel independent manner. In both Eq. (3.16) and Eq. (3.17), if the component value (d) of $\mathbf{n}_T[n_1, n_2]$ or $\mathbf{n}_t[n_1, n_2]$ equals the maximum or the minimum value of the digital image (i.e., 0 or 255 for an 8-bit channel of the 24-bit colour image in the RGB space), the impulse noise is referred to as the salt-and-pepper impulse [68]. Each pixel of the image may be corrupted by either the pepper or the salt impulse with unequal probabilities. The gray-scale image, $\mathbf{x}[n_1, n_2]$, is used for experiment. Figure 3.11 shows an example of images with different salt-and-pepper noises.

3.8 Image Rotation

The rotation operator performs a geometric transform which maps the position $[n_1, n_2]$ of a picture element in an input image onto a position $[n_1', n_2']$ in an output image by rotating it through a user-specified angle θ. In most implementations, output locations $[n_1', n_2']$ which are outside the boundary of the image are ignored. Rotation is most commonly used to improve the visual appearance of an image [70], although it can be useful as a preprocessor in applications where directional operators are involved. Rotation is a special case

Figure 3.11: Face images with salt-and-pepper noise. [8]

of affine transformation. The rotation operator performs a transformation of the form:

$$\begin{bmatrix} n'_1 \\ n'_2 \end{bmatrix} = \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix} \begin{bmatrix} n_1 - n_{1_0} \\ n_2 - n_{2_0} \end{bmatrix} + \begin{bmatrix} n_{1_0} \\ n_{2_0} \end{bmatrix} \tag{3.18}$$

where $[n_{1_0}, n_{2_0}]$ are the coordinates of the centre of rotation (in the input image) and θ is the angle of rotation with clockwise rotations having positive angles. Note here that image coordinates are used, so the n_2 axis goes downward. Similar rotation formula can be defined for when the n_2 axis goes upward. Even more than the translate operator, the rotation operation produces output locations $[n'_1, n'_2]$ which do not fit within the boundaries of the image (as defined by the dimensions of the original input image). In such cases, destination elements which have been mapped outside the image are ignored by most implementations. Pixel locations out of which an image has been rotated are usually filled in with black pixels. Figure 3.12 is shown the image with different

orientations.

Figure 3.12: Face images with different orientations. [8]

3.9 Summary

In this chapter, the overview architecture of a facial expression recognition system is described along with associated image pre-processing operations. Section 3.2 presents both the Viola-Jones face detection algorithm and the novel morphological face detection method. The advantage of morphological method is that it is less complex in terms of implementation and time processing than other methods. A novel facial detection module is described in Section 3.3 for sequence images. It is employed to FER system to detect the image frames with certain expression and used that image for further processing. In Section 3.6, a novel colour image pre-processing technique is introduced for colour facial expression images. The detail will be presented in Chapter 7.

[8] Original image from Cohn-Kanade database [13].

The impulsive noise modelling and image rotation is discussed in Sections 3.7 and 3.8. The image pre-processing module is designed to prepare the image for the other operations such as feature extraction. The detailed tasks depend on what kind of input is needed by feature extraction. Next chapter will discuss about feature extraction techniques.

Chapter 4

Facial Feature Extraction Methods

4.1 Introduction

To achieve better robustness and accuracy, before the normalized facial images are sent into classifier, they are projected into some feature spaces in which, hopefully, the faces are more separable [44, 58, 64]. In FER system, the feature extraction module is used to perform this projection [12, 71]. Feature extraction is an essential pre-processing step to pattern recognition and machine learning problems [20]. Feature extraction involves reducing the amount of resources required to describe a large set of data accurately. When performing analysis of complex data, one of the major problems stems from the number of variables involved. Analysis with a large number of variables generally requires a large amount of memory and computation power or a classification algorithm which overfits the training sample and generalizes poorly to new samples [35, 44, 58]. Feature extraction is a general term for methods of constructing combinations of the variables to get around these problems while still describing the data with sufficient accuracy.

The feature extraction process converts pixel data into a higher-level representation of shape, motion, colour, texture and spatial configuration of the face or its components. The extracted representation is used for subsequent classification. Feature extraction generally reduces the dimensionality of the input space. The reduction procedure should (ideally) retain essential informa-

tion possessing high discrimination power and high stability [22]. This chapter reviews several algorithms of feature extraction including Gabor filters, Log-Gabor filters, contourlet transform, local binary pattern operators and Zernike moments. In addition, it describes a number of novel feature extraction methods including HLACLF method, hybrid LGFCT method and hybrid face region method for FER systems.

4.2 Gabor Wavelet Filters

More recently, methods based on multi-resolution or multi-channel analysis such as wavelet transform and Gabor filters have gained a lot of attention for feature extraction and related applications [18, 72]. The wavelet transform decomposes the given image into only three directional components, i.e., horizontal, diagonal/antidiagonal and vertical detail sub bands in the direction of $0°, 45°/135°$ and $90°$, respectively, apart from the approximation sub-band. This limits the application of the wavelet transform for feature extraction from image in all directions. Feature extraction using Gabor functions is motivated by the fact that these filters can be considered as orientation and scale tunable detectors [73]. Basically, Gabor filters are a group of wavelets, with each wavelet capturing energy at a specific frequency and at a specific orientation or direction. There are several approaches to feature extraction, using banks of Gabor filters with different scale and orientation [12, 72, 73, 74]. The Gabor filters provide optimal Heisenberg joint resolution in space and spatial-frequency and it has been shown that Gabor filters exhibit spatial responses similar to receptive field profiles in mammalian vision [73]. Investigators have since successfully employed Gabor filters in a wide range of image-processing applications, including texture segmentation, document analysis, image coding, image and video picture quality assessment, target detection, fractal dimension measurement, edge detection, line characterization [72, 75, 76].

Gabor filters can be applied to images to extract features aligned at partic-

ular angles (orientations). Gabor filters possess optimal localization properties in both spatial and frequency domains and they have been successfully used in many pattern recognition applications [74, 77, 78, 79]. The most important parameters of a Gabor filter are its orientation and frequency. Certain features that share similar orientation or frequency can be selected and used to differentiate between different facial expressions depicted in images. A Gabor filter is a function obtained by modulating the amplitude of a sinusoid with a Gaussian function. Gabor filters are thought to mimic the functions of simple cells in the visual cortex. The various two dimensional receptive-field profiles encountered in populations of simple cells in the visual cortex are well described by an optimal family of two-dimensional (2-D) Gabor filters [12]. An elementary 1-D Gabor signal can be represented by:

$$g(t) = \frac{1}{\sqrt{2\pi}\sigma} \exp[-\frac{1}{2}(\frac{t-t_0}{\sigma})^2 + j\,\omega_t\,t] \tag{4.1}$$

which represents a modulation product of a sinusoidal wave of frequency ω_t and a Gaussian envelope of duration σ occurring at epoch t_0. The odd and even part of the 1-D Gabor filter can be derived from Eq. (4.1):

$$g(t) = g_{\text{even}}(t) + j g_{\text{odd}}(t) \tag{4.2}$$

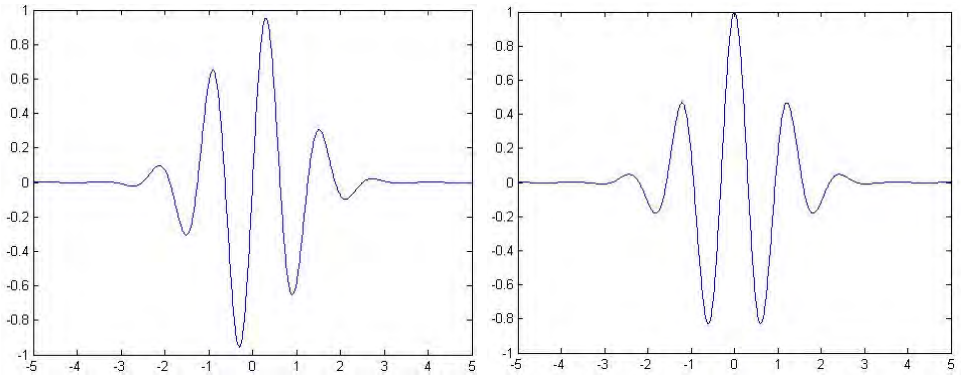
where

$$g_{\text{odd}}(t) = \frac{1}{\sqrt{2\pi}\sigma} \exp[-\frac{1}{2}(\frac{t-t_0}{\sigma})^2] \sin(\omega_t t) \tag{4.3}$$

$$g_{\text{even}}(t) = \frac{1}{\sqrt{2\pi}\sigma} \exp[-\frac{1}{2}(\frac{t-t_0}{\sigma})^2] \cos(\omega_t t) \tag{4.4}$$

Figure 4.1 shows the odd part and the even part of the 1-D Gabor filter based on Eq. (4.3) and Eq. (4.4).

A two-dimensional Gabor filter is expressed as a Gaussian modulated sinusoid in the spatial domain and as a shifted Gaussian in the frequency domain. Recent studies on modelling of visual cortical cells [74] suggest a tuned band-



(a) (b)

Figure 4.1: One-dimensional Gabor filter (a) odd part (b) even part. ($t_0 = 0$, $\sigma = 1$ and $\omega_t = 5$.)

pass filter bank structure. These filters are found to have Gaussian transfer functions in the frequency domain. The Inverse Fourier Transform of these transfer functions has characteristics closely resembling Gabor filters. A well designed Gabor filter bank can capture the relevant frequency spectrum in all directions. In the spatial domain, a Gabor filter is a complex exponential modulated by a Gaussian function [72]. An example of the Gabor filter bank feature images is shown in Figure 4.2.

A 2-D Gabor filter can be represented by the following equation:

$$\mathbf{g}_{n_w,n_o}[n_1,n_2] = \frac{1}{2\pi\sigma_{n_1}\sigma_{n_2}} \exp[-\frac{1}{2}(\frac{n_1'^2}{\sigma_{n_1}^2} + \frac{n_2'^2}{\sigma_{n_2}^2})] \times \exp[j\frac{2\pi {n_1}'}{\lambda}] \tag{4.5}$$

where

$$\begin{bmatrix} n_1' \\ n_2' \end{bmatrix} = \begin{bmatrix} cos\theta & sin\theta \\ -sin\theta & cos\theta \end{bmatrix} \begin{bmatrix} n_1 \\ n_2 \end{bmatrix} \tag{4.6}$$

knowing that $[n_1, n_2]$ is the pixel position in the spatial domain, λ is the wavelength (a reciprocal of frequency) in pixels, θ is the orientation of a Gabor filter and $\sigma_{n_1}, \sigma_{n_2}$ are the standard deviation along the n_1 and n_2 directions respectively.

In most cases, a Gabor filter bank with five frequencies and eight orientations is used to extract the features for a face representation [12]. The five wavelengths (reciprocals of frequencies) $\lambda_{n_w=1,\cdots,5}$ and the eight orientation

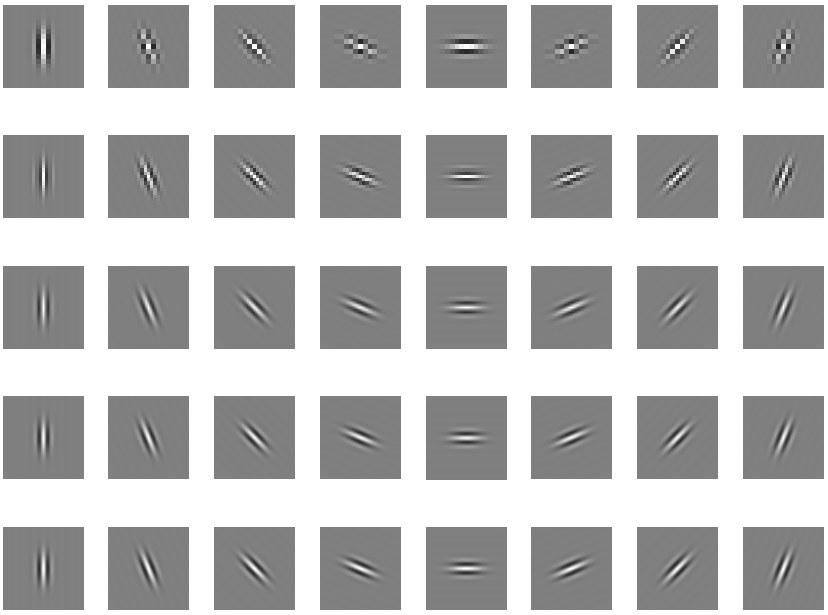


Figure 4.2: Bank of Gabor filters with 5 frequencies and 8 orientations.

angles $\theta_{n_o=1,\cdots,8}$ are calculated as:

$$\lambda_{n_w} = \lambda_{\max} \times 2^{\frac{1}{2}\times(n_w-1)} \tag{4.7}$$

$$\theta_{n_o} = \frac{\pi}{8} \times (n_o - 1) \tag{4.8}$$

where $\lambda_{\max} = 4$. For a given input image, $\mathbf{x}[n_1, n_2]$, the Gabor filter output $\mathbf{x}^G_{n_w,n_o}[n_1, n_2]$ is obtained by the convolution operation as follows:

$$\mathbf{x}^G_{n_w,n_o}[n_1, n_2] = \mathbf{g}_{n_w,n_o}[n_1, n_2] * \mathbf{x}[n_1, n_2] \tag{4.9}$$

and

$$\mathbf{g}_{n_w,n_o}[n_1, n_2] * \mathbf{x}[n_1, n_2] = \sum_{p=0}^{M-1}\sum_{q=0}^{N-1} \mathbf{g}_{n_w,n_o}[p, q]\mathbf{x}[n_1 - p, n_2 - q] \tag{4.10}$$

where the symbol "$*$" denotes the 2-D convolution and M and N are the

height and width of the Gabor filter mask. The convolution operation can be performed separately for the real and imaginary part as:

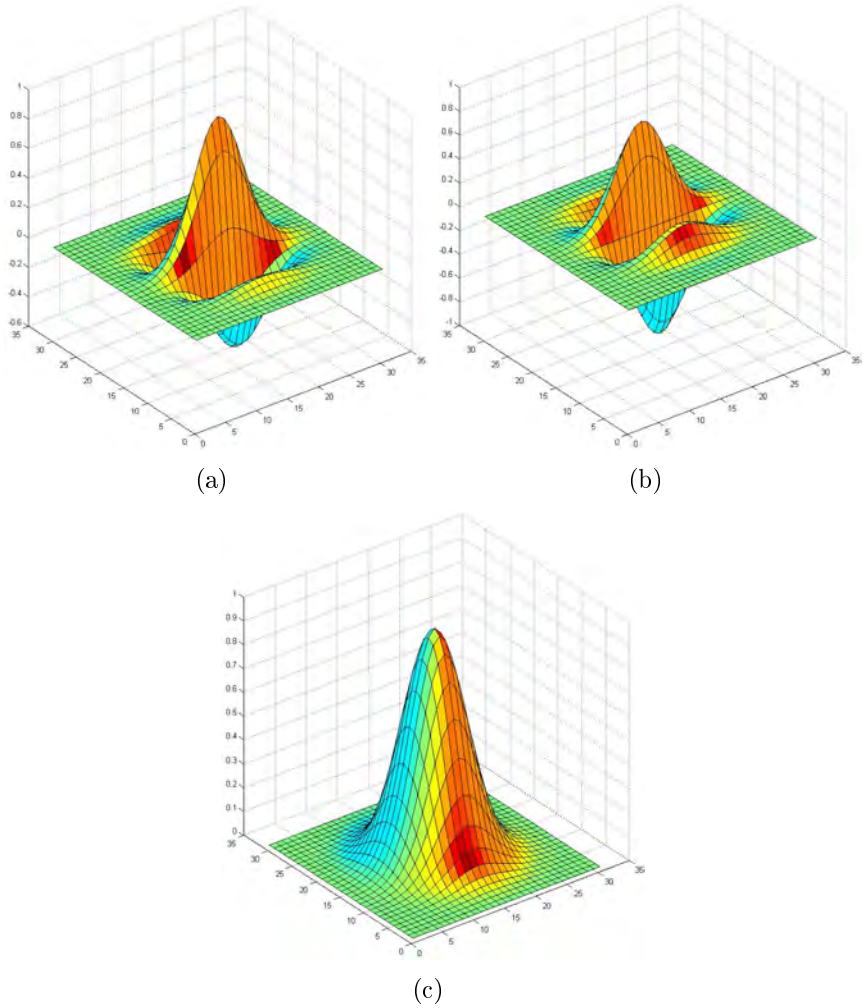
$$Re\{\mathbf{x}^G_{n_w,n_o}[n_1,n_2]\} = \mathbf{x}[n_1,n_2] * Re\{\mathbf{g}_{n_w,n_o}[n_1,n_2]\} \tag{4.11}$$

$$Im\{\mathbf{x}^G_{n_w,n_o}[n_1,n_2]\} = \mathbf{x}[n_1,n_2] * Im\{\mathbf{g}_{n_w,n_o}[n_1,n_2]\} \tag{4.12}$$

This is followed by the amplitude calculation:

$$|\mathbf{x}^G_{n_w,n_o}[n_1,n_2]| = \sqrt{Re\{\mathbf{x}^G_{n_w,n_o}[n_1,n_2]\}^2 + Im\{\mathbf{x}^G_{n_w,n_o}[n_1,n_2]\}^2} \tag{4.13}$$

An example of the Gabor filter bank feature images is shown in Figure 4.3.



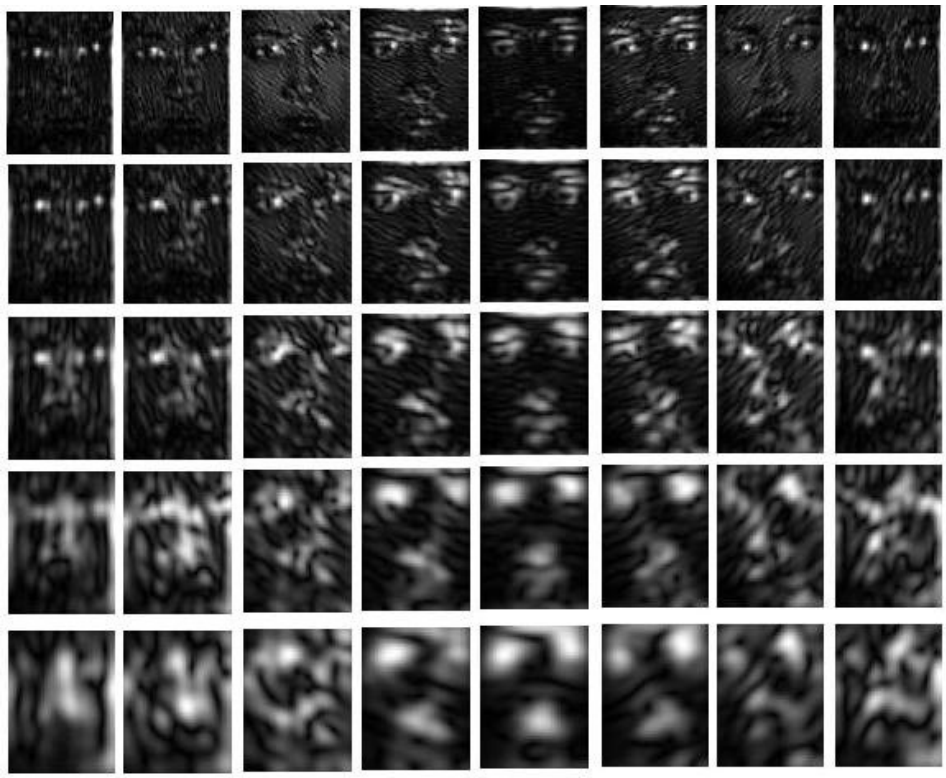
(a) (b)

(c)

Figure 4.3: (a) Real part of Gabor filter (b) Imaginary part of Gabor filter (c) magnitude of Gabor filter.

In practice, the time for performing Gabor feature extraction is very long and the dimension of the Gabor feature vector is prohibitively large. To reduce the size of feature vectors, down sampling can be performed without losing important information. In this study, it is proposed to average the Gabor filter bank features (AGF) [77], which means reducing the 40 feature images to a single average feature image [Eq. (4.14)].

$$\mathbf{x}^{\mathrm{AGF}}[n_1, n_2] = \frac{\sum_{n_w=1}^{5} \sum_{n_o=1}^{8} |\mathbf{x}_{n_w,n_o}^{G}[n_1, n_2]|}{8 \times 5} \tag{4.14}$$



(a)

(b)

Figure 4.4: (a) 40 Gabor filter bank feature images calculated for the face image from JAFFE database. (b) An averaged feature generated by the Gabor filter bank.

4.3 Logarithmic Gabor Filters

Gabor filters are commonly recognized as one of the best choices for obtaining localized frequency information. However, they suffer from two major limitations. The maximum bandwidth of a Gabor filter is limited to approximately one octave and Gabor filters are not optimal if one is seeking broad spectral information with maximal spatial localization. As an alternative to the Gabor filters, the Log-Gabor filters were proposed by Field [80]. Log-Gabor filters can be constructed with arbitrary bandwidth, which can be optimized to produce a filter with minimal spatial extent. The Log-Gabor filters (LGF) in the frequency domain can be defined in polar coordinates by the transfer function $H(f,\theta)$ constructed as a product $H(f,\theta) = H_f \times H_\theta$ of the radial component H_f controlling the bandwidth that the filter responds to and the angular component H_θ, controlling the spatial orientation that the filter responds to. Figure 4.5 illustrates the even-symmetric component and odd-symmetric component of Log-Gabor filter in time domain.

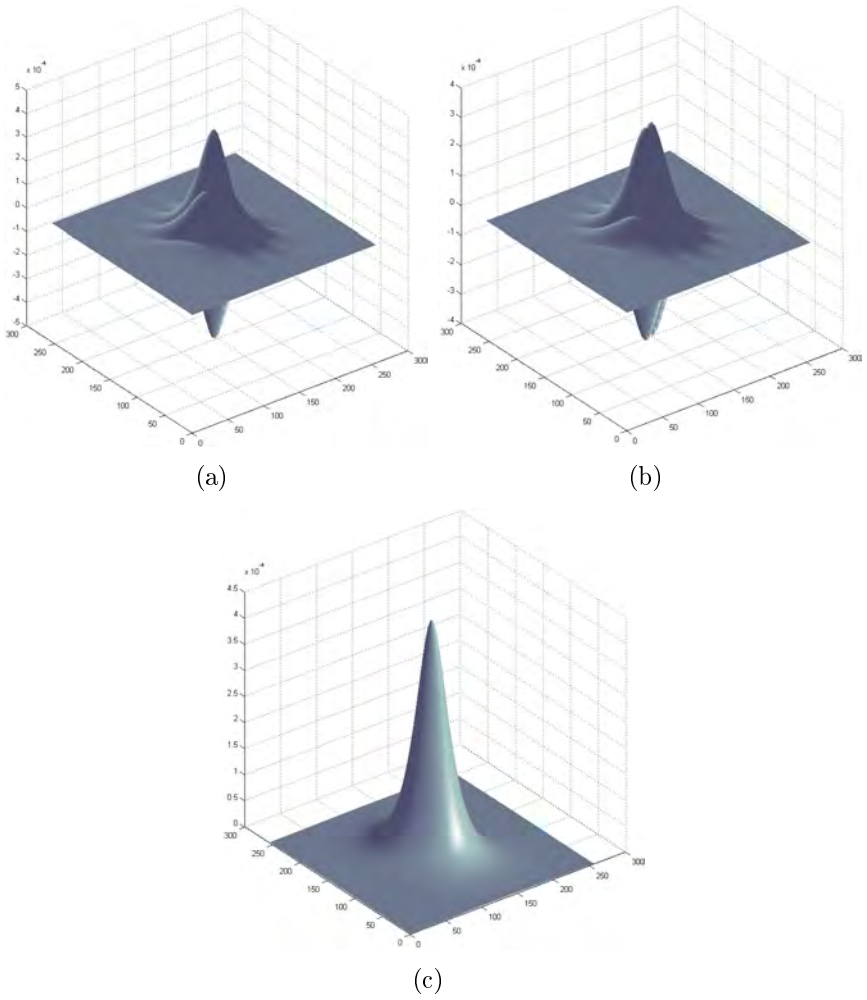
The 2-D Log-Gabor filters can be represented in a polar form as follows:

$$H(f,\theta) = \exp\{\frac{-[\ln(\frac{f}{f_0})]^2}{2[\ln(\frac{\sigma_f}{f_0})]^2}\} \exp\{\frac{-(\theta-\theta_0)^2}{2\sigma_\theta^2}\} \tag{4.15}$$

where f_0 is the filter's centre frequency and θ_0 the filter's direction. The constant σ_f defines the radial bandwidth B in octaves and the constant σ_θ, defines the angular bandwidth $\Delta\Omega$ in radians:

$$B = 2\sqrt{\frac{2}{\ln 2}} \times |\ln(\frac{\sigma_f}{f_0})|; \quad \Delta\Omega = 2\sigma_\theta\sqrt{\frac{2}{\ln 2}} \tag{4.16}$$

In the study described here, the ratio σ_f/f_0 is kept constant for varying f_0, B is set to one octave and the angular bandwidth is set to $\Delta\Omega = \pi/4$ radians. This left only σ_f to be determined for a varying value of f_0. Five scales and eight orientations are implemented to extract features from face images. This led to 40 filter transfer functions, $\{H_1, H_2, \cdots, H_{40}\}$, represent-



(a) (b)

(c)

Figure 4.5: An example of Log-Gabor filter (a) Even-symmetric component (b) Odd-symmetric component (c)Magnitude of Log-Gabor filter

ing different scales and orientations. The image filtering is performed in the frequency domain making the process faster compared with the space domain convolution. After the 2-D Discrete Fourier Transform (DFT), the image arrays $\mathbf{x}[n_1, n_2]$ are transformed into the spectral domain matrix representation, $\mathbf{X}$, which is multiplied by the Log-Gabor transfer functions $\{H_1, H_2, \cdots, H_{40}\}$, producing 40 spectral representations for each image. These filtered images in the transform domain are then transformed back to the spatial domain via the 2-D inverse DFT. This process resulted in prohibitively large number of the feature arrays.

An example of LGF features is shown in Figure 4.6. For large training and

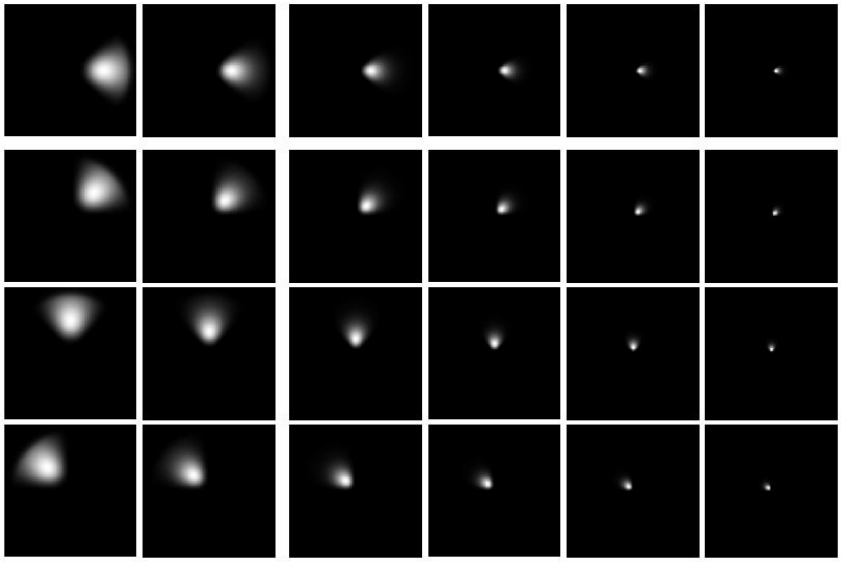


Figure 4.6: Sample of Log-Gabor filters with 6 frequencies and 4 orientations in frequency domain.

testing sets, the computations are highly impractical [81]. In order to improve the computational efficiency, it is critical to reduce the features dimensions. This is achieved through data selection and data reduction processes.

4.4 Contourlet Transform

The Contourlet Transform is a directional transform, which is capable of capturing contours and fine details in images [82]. The contourlet expansion is composed of basis function oriented at various directions in multiple scales, with flexible aspect ratios. With this rich set of basis functions, the contourlet transform effectively captures smooth contours that are the dominant feature in natural images. It is well known that many signal processing tasks, e.g., compression, denoising, feature extraction and enhancement, benefit tremendously from having a parsimonious representation of the signal at hand. The first stage transforms the original image into a Laplacian pyramid (LP) having $N_l + 1$ scale levels. The second stage is a decomposition of each LP scale level into D sub-bands through a directional filter bank (DFB) structure using quincunx filters. The Laplacian pyramid decomposes images into sub-bands

and then the directional filter banks analyze each detail image as illustrated in Figure 4.7.

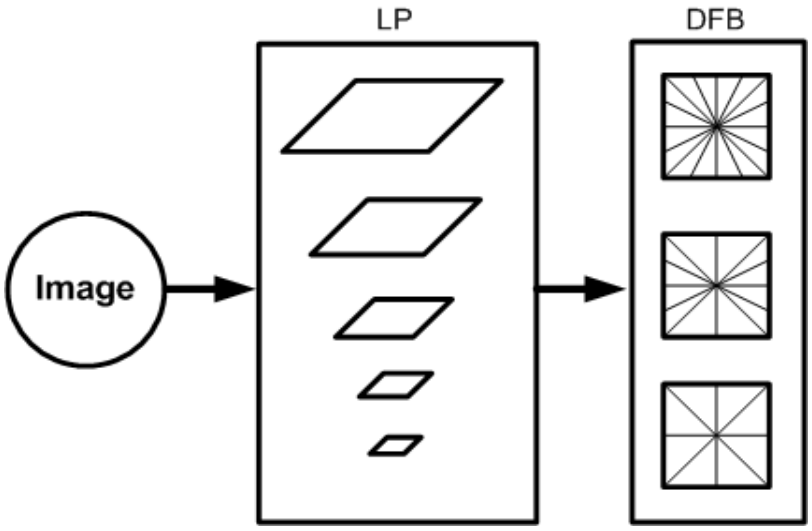


Figure 4.7: Contourlet block diagram.

The pyramidal directional filter bank (PDFB) [83] was proposed by Do and Vetterli, which overcomes the block-based approach of curvelet transform by a directional filter bank, applied on the whole scale also known as contourlet transform (CT). The grouping of wavelet coefficients suggests that one can obtain a sparse image expansion by first applying a multi-scale transform and then applying a local directional transform to gather the nearby basis functions at the same scale into linear structures. In essence, first a wavelet-like transform is used for edge (points) detection and then a local directional transform for contour segments detection. With this insight, one can construct a double filter bank structure [see Figure 4.8(a)] where at first the Laplacian pyramid (LP) is used to capture the point discontinuities and followed by a directional filter bank (DFB) to link point discontinuities into linear structures [82]. The overall result is an image expansion with basis images as contour segments and thus it is named the contourlet transform. The combination of this double filter bank is named pyramidal directional filter bank (PDFB). Figure 4.8(a) shows the block diagram of the CT. First a standard multi-scale decomposition into octave bands is computed, where the low pass channel is sub-sampled while

the high pass is not. Then a directional decomposition with a DFB is applied to each high pass channel.

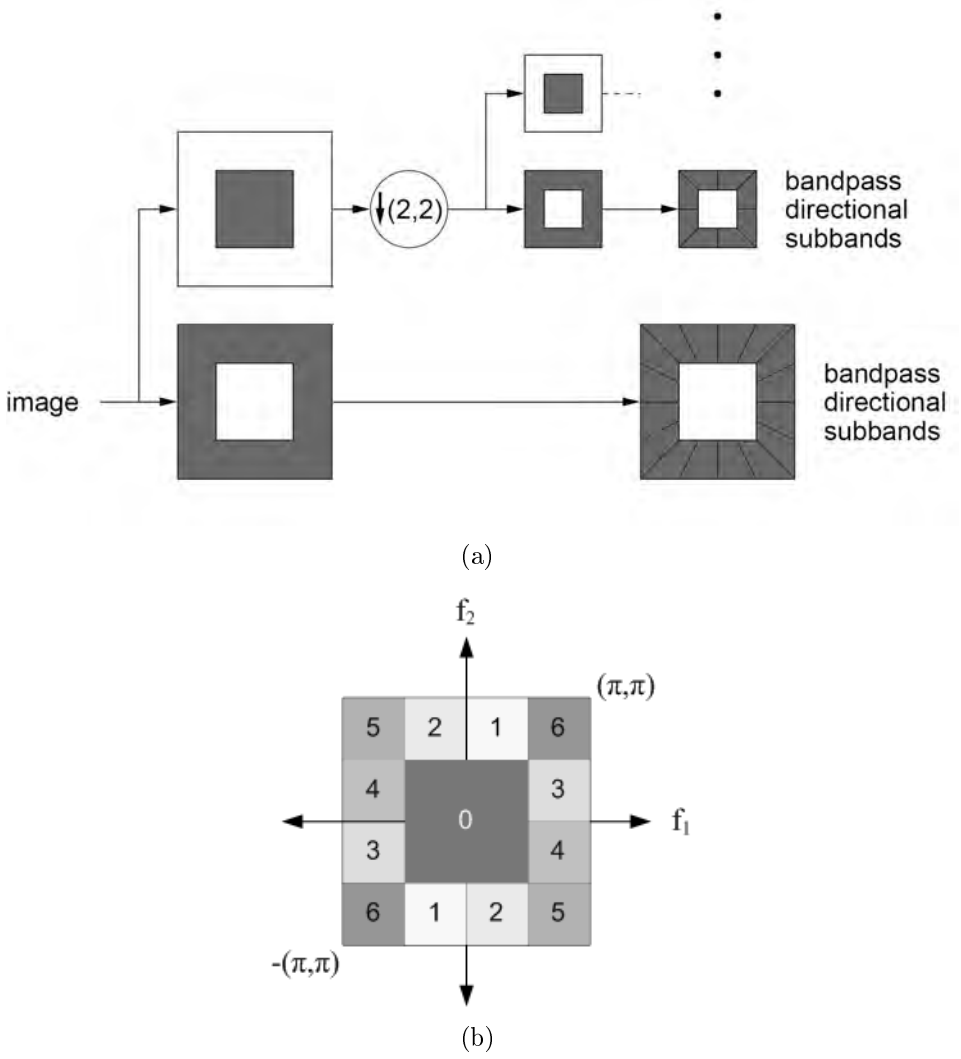


Figure 4.8: (a) Contourlet filter bank. (b) 2-D frequency spectrum division of contourlet.

Figure 4.9 shows the support shapes for contourlets implemented by a PDFB that satisfies the anisotropic scaling relation. From the upper line to the lower line, four reduces the scale while the number of directions is doubled. PDFB allows for different number of directions at each scale/resolution to nearly achieve critical sampling.

As DFB is designed to capture high frequency components (representing directionality), the LP part of the PDFB permits sub-band decomposition

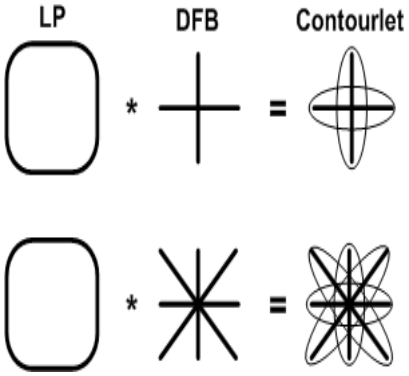


Figure 4.9: Contourlet construction.

to avoid leaking of low frequencies into several directional sub-bands, thus directional information can be captured efficiently. In general, the contourlet construction allows for any number of DFB decomposition levels (n_{b_j}) to be applied at each LP level j. For the contourlet transform to satisfy the anisotropy scaling relation, one simply needs to impose that in the PDFB, the number of directions is doubled at every other finer scale of the pyramid. In the frequency domain, the contourlet transform provides a multi-scale and directional decomposition. The main idea of contourlet is to find some directional extension to further divide each high pass sub-band into two directions. Figure 4.8(b) shows the desired frequency partitioning in contourlet transform which contains of six directional sub-bands roughly oriented at 15°, 45°, 75°, 105°, 135° and 165°. The number of contourlet images are obtained by [83]:

$$N_{co} = \sum_{j=1}^{N_l} 2^{n_{b_j}} \tag{4.17}$$

where n_{b_j} is the number of sub-bands of the DFB, N_l is the number of scales of the LP.

4.4.1 Laplacian Pyramid

One way of achieving a multi-scale decomposition is to use a Laplacian pyramid introduced by Burt and Adelson [84]. The LP decomposition at each level

generates a down sampled low pass version of the original and the difference between the original and the prediction, resulting in a bandpass image as shown in Figure 4.10(b). In this figure, 'H' and 'G' are called analysis and synthesis filters and 'M' is the sampling matrix. The process can be iterated on the coarse version. In Figure 4.10(a), the outputs are a coarse approximation 'a' and a difference 'b' between the original signal and the prediction. The process can be iterated by decomposing the coarse version repeatedly. The original image is convolved with a Gaussian kernel [83]. The resulting image is a low pass filtered version of the original image. The Laplacian is then computed as the difference between the original image and the low pass filtered image. This process is continued to obtain a set of band-pass filtered images (since each one is the difference between two levels of the Gaussian pyramid). Thus the Laplacian pyramid is a set of band pass filters. A sequence of images are obtained by repeating these steps several times. If these images are stacked one above another, the result is a tapering pyramid data structure, as shown in Figure 4.11 and hence the name. The Laplacian pyramid can thus be used to represent images as a series of band-pass filtered images, each sampled at successively sparser densities. It is frequently used in image processing and pattern recognition tasks because of its ease of computation. A drawback of the LP is the implicit over-sampling. However, in contrast to the critically sampled wavelet scheme, the LP has the distinguishing feature that each pyramid level generates only one bandpass image (even for multi-dimensional cases), which does not have scrambled frequencies. This frequency scrambling happens in the wavelet filter bank when a high pass channel, after down-sampling, is folded back into the low frequency band and thus its spectrum is reflected. In the LP, this effect is avoided by down-sampling the low pass channel only.

4.4.2 Directional Filter Bank

The 2-D directional filter bank was introduced by Bamberger and Smith in 1992 [85]. The directional filter bank is a critically sampled filter bank that can

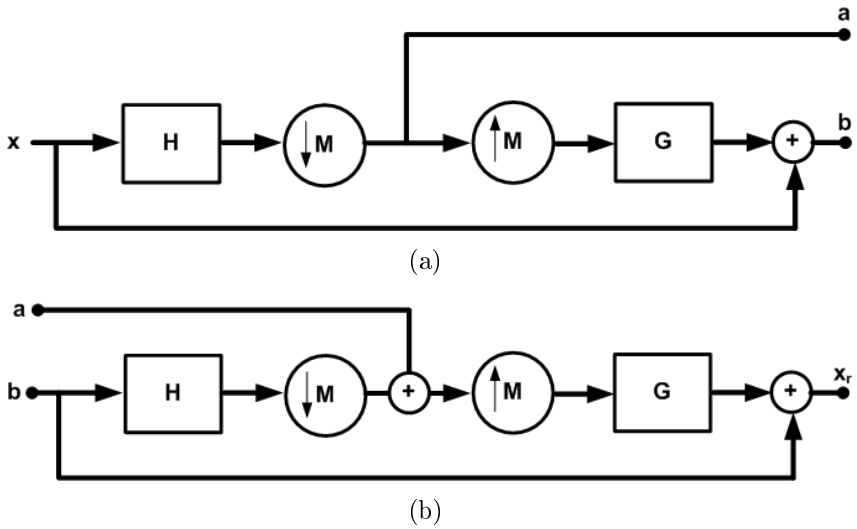


Figure 4.10: Laplacian pyramid scheme (a) analysis and (b) reconstruction.

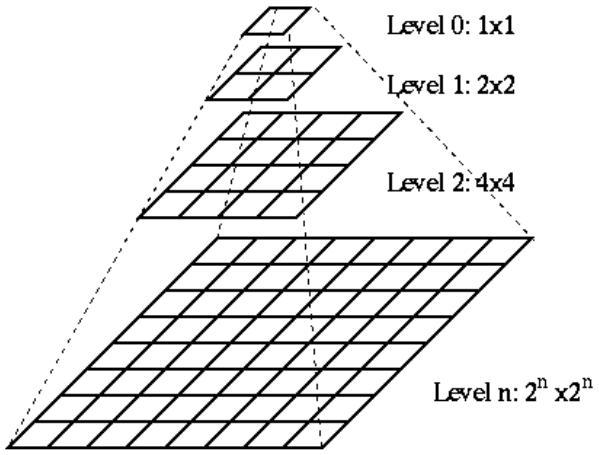


Figure 4.11: Laplacian pyramid structure

decompose images into any power of two's number of directions. The DFB is efficiently implemented via a l-level tree structured decomposition that leads to 2^l sub-bands with wedge-shaped frequency partition as shown in Figure 4.8(b). The original construction of the DFB involves modulating the input signal and using diamond-shaped filters. Furthermore, to obtain the desired frequency partition, an involved tree expanding rule has to be followed. As a result, the frequency regions for the resulting sub-bands do not follow a simple ordering as shown in Figure 4.8(b) based on the channel indices. The DFB is designed to capture the high frequency components (representing directionality) of images.

Therefore, low frequency components are handled poorly by the DFB. In fact, with the frequency partition shown in Figure 4.8(b), low frequencies would leak into several directional sub-bands, hence DFB does not provide a sparse representation for images. To improve the situation, low frequencies should be removed before the DFB. This provides another reason to combine the DFB with a multiresolution scheme. Therefore, the LP permits further sub-band decomposition to be applied on its bandpass images. Those bandpass images can be fed into a DFB so that directional information can be captured efficiently. The scheme can be iterated repeatedly on the coarse image. The end result is a double iterated filter bank structure, named pyramidal directional filter bank (PDFB), which decomposes images into directional sub-bands at multiple scales. The scheme is flexible since it allows for a different number of directions at each scale.

Figure 4.12 shows the face image decomposed using the contourlet transform. It can be seen that contourlet transform provides successive refinements at both spatial and directional resolution. Each image is decomposed into a low pass sub-band and several bandpass directional sub-bands. It can be seen that only contourlets that match with both location and direction of image contours produce significant coefficients. Thus, the contourlet transform effectively explores the fact, that the edges in images are localized in both location and direction. One can decompose each scale into any arbitrary power of two's number of directions and different scales can be decomposed into different numbers of directions. This feature makes contourlets a unique transform that can achieve a high level of flexibility in decomposition while being close to critically sampled. Other multi-scale directional transforms either have a fixed number of directions or are significantly over complete.

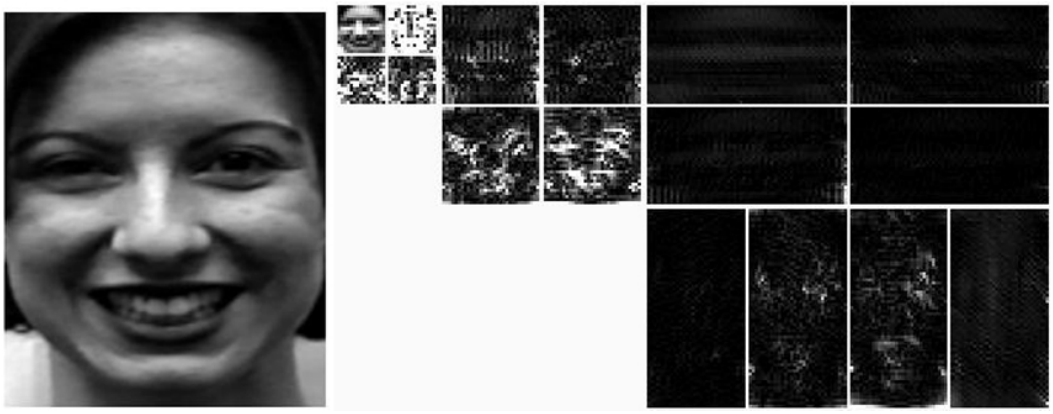


Figure 4.12: The contourlet coefficients of face image from Cohn-Kanade database

4.5 Local Binary Pattern Operator

The local binary pattern (LBP) is a non-parametric operator which describes the local spatial structure of an image. The original LBP operator was introduced by Ojala [86] showed its high discriminative power for texture classification. At a given pixel position (n_1, n_2), LBP is defined as an ordered set of binary comparisons of pixel intensities between the centre pixels, (n_{1_c}, n_{2_c}) and its surrounding pixels, (n_{1_p}, n_{2_p}), for $p = 1, \cdots P$. Figure 4.13 shows an example of LBP for different surrounding pixels. The LBP operator $LBP_{P,R}$ produces 2^P different output values, corresponding to the 2^P different binary patterns that can be formed by the P pixels in the neighbour set. The LBP is obtained by first concatenating these binary numbers and then converting the sequence into the decimal number. Using circular neighbourhoods and linearly interpolating the pixel values allows the choice of any radius, R and number of pixels in the neighbourhood, P, to form an operator. The decimal form of the resulting LBP code can be defined by

$$LBP_{P,R}[n_{1_c}, n_{2_c}] = \sum_{n_{1_p}, n_{2_p} \in \mathbf{P}_n} I_b(\mathbf{x}[n_{1_p}, n_{2_p}] - \mathbf{x}[n_{1_c}, n_{2_c}]) 2^{(p-1)} \qquad (4.18)$$

and

$$\mathbf{P}_n = \{(n_{1_p}, n_{2_p}) | \sqrt{(n_{1_p} - n_{1_c})^2 + (n_{2_p} - n_{2_c})^2} = R, 1 \leq p \leq P\} \tag{4.19}$$

where $\mathbf{x}[n_{1_c}, n_{2_c}]$ denotes the gray value of the centre pixels, $\mathbf{x}[n_{1_p}, n_{2_p}]$ represents the gray values of the surrounding pixels and I_b is the binary function that expressed by:

$$I_b(\mathbf{x}[n_{1_p}, n_{2_p}] - \mathbf{x}[n_{1_c}, n_{2_c}]) = \begin{cases} 1, & \mathbf{x}[n_{1_p}, n_{2_p}] - \mathbf{x}[n_{1_c}, n_{2_c}] \geq 0 \\ 0, & \mathbf{x}[n_{1_p}, n_{2_p}] - \mathbf{x}[n_{1_c}, n_{2_c}] < 0 \end{cases} \tag{4.20}$$

It has been shown that certain bins contain more information than others [86]. Therefore, it is possible to use only a subset of the 2^P LBP to describe the texture of images. These fundamental patterns called uniform patterns [87]. A Local Binary Pattern is called uniform if it contains at most two bitwise transitions from 0 to 1 or vice versa when the binary string is considered circular. For example, 00000000, 001110000 and 11100001 are uniform patterns. It is observed that uniform patterns account for nearly 90% of all patterns in the $(8, 1)$ neighbourhood and for about 70% in the $(16, 2)$ neighbourhood in texture images [see Figure 4.13]. Accumulating the patterns which have more than 2 transitions into a single bin yields an LBP operator, denoted $LBP^{u2}_{P,R}$ with less than 2^P bins. For example, the number of labels for a neighbourhood of 8 pixels is 256 for the standard LBP but 59 for LBP^{u2}.

$LBP^{u2}_{P,R}$ means using the operator in a neighbourhood of P sampling points on a circle of radius R. Superscript $u2$ stands for using uniform patterns and labelling all remaining patterns with a single label. In this work, $LBP^{u2}_{8,2}$ is applied to extract LBP code for each pixel of face images, generating LBP features. All feature values are quantified into 59 bins according to uniform strategy. The LBP code contains information about the distribution of the local micro-patterns, such as edges, spots and flat areas over the whole image.

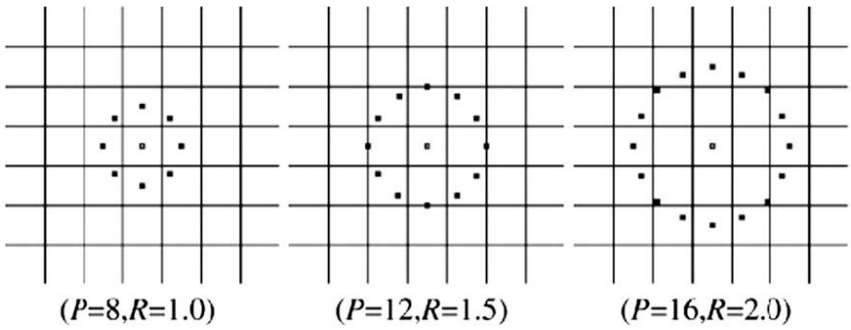
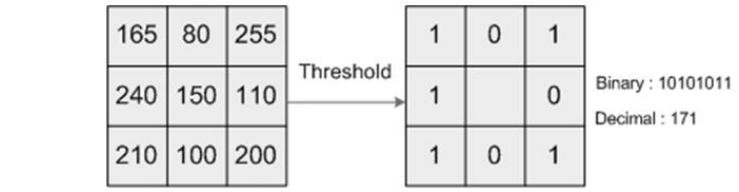


Figure 4.13: Three examples of the extended LBP: the circular $(8, 1)$ neighbourhood, the circular $(12, 1.5)$ neighbourhood and the circular $(16, 2)$ neighbourhood, respectively.

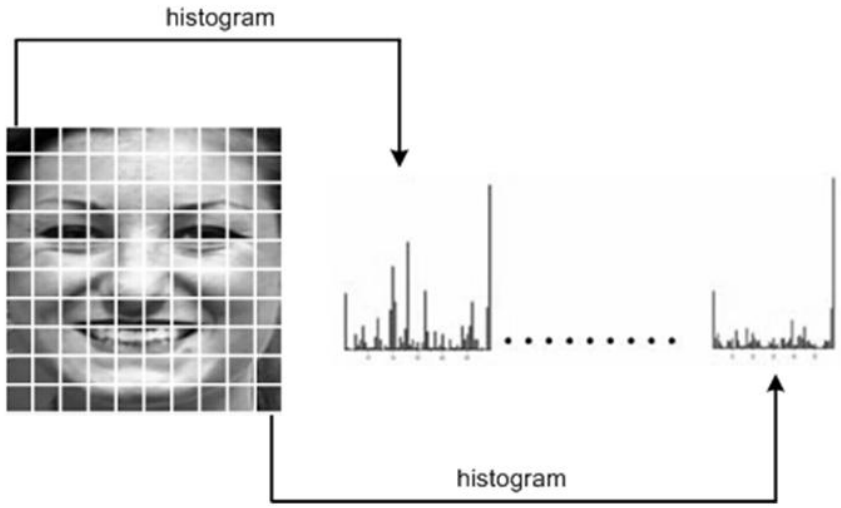
Face images can be seen as a composition of micropatterns which can be effectively described by the LBP features. However, an LBP histogram computed over the whole face image encodes only the occurrences of the micropatterns without any indication about their locations. Considering shape information of faces, face images are divided into small regions $\{\mathbf{x}_{r_0}, \mathbf{x}_{r_1}, \cdots, \mathbf{x}_{r_k}\}$ to extract LBP features. A spatial histogram, which concatenates the histograms of all the sub-regions, is employed to represent the face [see Figure 4.14(b)]. The spatial histogram encodes both the appearance and the spatial relations of facial regions. The spatial histogram (H_{sb}) of the face image is represented as:

$$H_{sb} = \{H_{LBP}(\mathbf{x}_{r_i}) | i = 1, \cdots, k\} \tag{4.21}$$

where $H_{LBP}(\mathbf{x}_{r_i})$ denotes the histogram of the LBP patterns extracted from the sub-region $\mathbf{x}_{r_i}$. The extracted feature histogram represents the local texture and global shape of face images. Some parameters can be optimized for better feature extraction. One is the LBP operator and the other is the number of regions divided. The 59-bin $LBP^{u2}_{8,2}$ operator is selected and the face images are divided into 100 regions, giving a good trade-off between recognition performance and feature vector length [87]. To generate the feature vector, the face images are divided into 100 (10×10) regions as shown in Figure 4.14(b) and represented by the LBP histograms [88, 89]. The feature vector, $\mathbf{x}^f_{\text{LBP}}$, for



(a)



(b)

Figure 4.14: (a) Illustration of the basic LBP operator.(b) A facial image is divided into 100 small regions from which LBP histograms are extracted and concatenated into a single histogram.

facial expression image is given by

$$\mathbf{x}_{\mathrm{LBP}}^{f} = \{H_{LBP}(\mathbf{x}_{r_i})|i = 1, \cdots, 100\} \tag{4.22}$$

4.6 Higher Order Spectra

Signals such as speech, image are generated mechanically by systems with non-linear dynamics and, as a result, obviously most of them are non-Gaussian. If the prediction and coding quality of such signals are to be improved, more of the information available in the signal must be used and the signal higher-order statistics (HOS) must be exploited. The scientific field of statistics provides many tools to handle random signals. In signal processing, first and second order statistics have gained significant importance [90]. However, many signals,

especially when it comes to nonlinearities, cannot be examined properly by second order statistical methods [91]. For this reason higher-order statistical methods have been developed during the 20^{th} century. Higher-order statistics techniques have been applied to real signal processing problems and expand into different fields such as economics, speech, seismic data processing, plasma physics, optics and images. HOS measures are extensions of second-order measures (such as the autocorrelation function and power spectrum) to higher orders. The second order measures work fine if the signal has a Gaussian (normal) probability density function (pdf), but as mentioned above, many real-life signals are non-Gaussian. In this thesis, higher-order local autocorrelation (HLAC) features are used for feature extraction. The HLAC is introduced in the following subsection.

4.6.1 Higher-Order Local Autocorrelation

The HLAC features, an extension of autocorrelation features (second-order statistics), are based on HOS. The features are generated using higher-order local autocorrelation. The d^{th}-order autocorrelation functions, extensions of autocorrelation functions, are defined as [86]:

$$x_{hlac}(a_1, a_2, ..., a_d) = \int x(z)x(z + a_1) \cdots x(z + a_d)dz \tag{4.23}$$

where $x(z)$ denotes the intensity at the observing pixel z and $a_1, a_2, \cdots, a_d$ are d displacements. HLAC features are primitive image features based on Eq. (4.23) [92]. Their orders and displacements are arbitrary. However, higher-order features with a large displacement region become extremely numerous. Hence, the original HLAC features are restricted up to the second-order (three-point relations) and within a 3×3 displacement region. They are represented by 25 mask patterns with 0, 1 and 2 displacements [25 mask patterns in Figure 4.15(a)]. Each mask pattern is scanned over the entire image and for each possible position, the product of the pixels marked in white is computed.

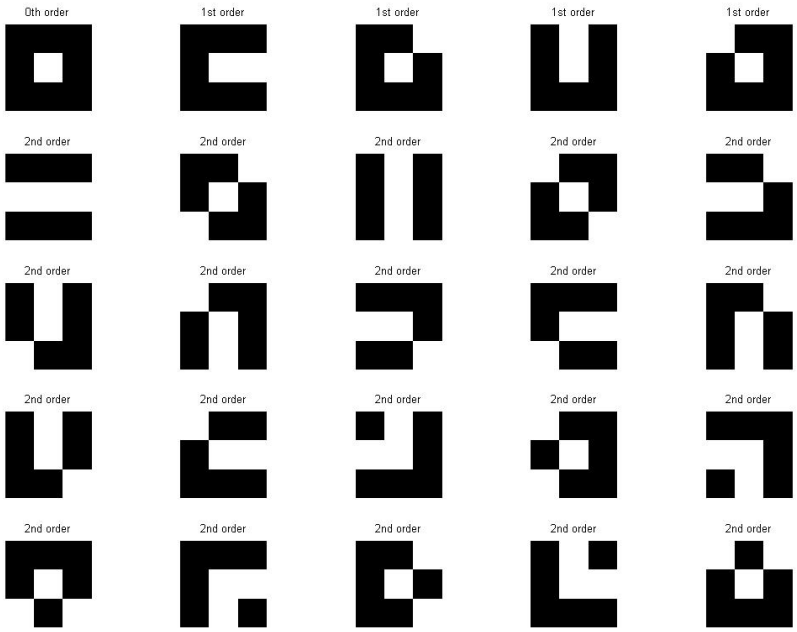
All the products corresponding to a mask are then summed so as to provide one feature. This operation is performed using 25 different mask pattern to create the feature vector for each facial image. Each feature value represents the power spectrum of the mask pattern, which corresponds to a basis functions of frequency analysis [93]. As a rough comparison with a Fourier transform, the mask size corresponds to the frequency component and the distribution of the displacements corresponds to the direction component. Since the HLAC features use the information of two-dimensional distributions as well as the directions, they analyze the image more closely.

Furthermore, large mask patterns are used to support large displacement regions [Figure 4.15(b)] and extract the features of low resolutions or low frequencies.

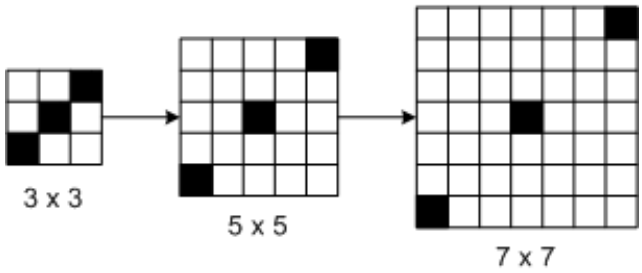
4.7 Statistical Moments

Moment functions are applicable to many different aspects of image processing, ranging from invariant pattern recognition and image encoding to pose estimation [58, 94]. When applied to images, they describe the image content (or distribution) with respect to its axes. They are designed to capture both global and detailed geometric information about the image. In this study, they are used to characterize a gray level image so as to extract properties that have analogies in statistics or mechanics. Generally, these features are invariant under image translation, scale normalization and rotation only when they are computed from the original non-distorted analogue two-dimensional image. In practice, one observes the digitized, quantized and often noisy version of the image and the invariance properties are satisfied only approximately. Regular moments have by far been the most popular type of moments. They are defined for digital image as [94, 95]:

$$m_{pq} = \sum_{n_1=1}^{N_1} \sum_{n_2=1}^{N_2} n_1^p n_2^q \mathbf{x}[n_1, n_2] \tag{4.24}$$



(a)



(b)

Figure 4.15: (a) 25 mask patterns of the HLAC features (3x3). (b) An extension of HLAC features.

where m_{pq} is the $(p+q)^{th}$ order moment of the digital image function $\mathbf{x}[n_1, n_2]$ and N_1 and N_2 are the height and the width of the image in pixels.

Seven nonlinear functions defined on regular moments which are translation, scale and rotation invariant are introduced by Hu [96]. These seven are, therefore, moment invariant. The definition of regular moments has the form of projection of $\mathbf{x}[n_1, n_2]$ function onto the monomial $n_1^p n_2^q$. Unfortunately, the basis set $n_1^p n_2^q$ is not orthogonal. Consequently, the recovery of image from these moments is quite difficult and computationally expensive. Moreover, it implies that the information contents of m_{pq} have a certain degree of re-

dundancy. To overcome the problems associated with the regular moments, an orthogonal moment based on the theory of orthogonal polynomials has suggested by Teague [94]. The polynomials, including Legender, Zernike and pseudo-Zernike, have been used to generate moment-based features which are invariant to location, size and rotation [97].

Zernike moments (ZM) used in this study are a class of such orthogonal moments. The reason for selecting them from among the other orthogonal moments is that they possess a useful rotation invariance property. Rotating the image does not change the magnitudes of its Zernike moments. Hence, they could be used as rotation invariant features for image representation. These features could easily be constructed to an arbitrary high order. Hence, Zernike moments play a vital role in feature extraction of digital images. These Zernike moments have proved to be better in terms of their feature representation capability, rotation invariance, fast computation, multi-level representation for describing the shapes of patterns and low noise sensitivity [98, 95]. In the following subsection, the Zernike moments are described with its application for extracting the features from face image.

4.7.1 Zernike Moments

The advantages of considering orthogonal moments are that they are shift, rotation and scale invariant and very robust in the presence of noise. The invariant properties of moments are utilized as pattern sensitive features in classification and recognition applications [7, 98, 99, 100, 101]. Zernike moments are useful tools in pattern recognition and image analysis due to their orthogonality and rotation invariance property. The kernel of Zernike moments is a set of orthogonal Zernike polynomials defined over the polar coordinate space inside a unit circle. The Zernike polynomial expression $Z_{o,m}(r, \theta)$ is a complex polynomial expression representing a complete orthogonal system in a unit circle. It is defined as [98]:

$$Z_{o,m}(r,\theta) = R_{o,m}(r)e^{jm\theta}, \quad 0 \leq r \leq 1 \quad , 0 \leq \theta \leq 2\pi \tag{4.25}$$

where o denotes the order, m represents the number of iterations which satisfies the conditions that $o-|m|$ is even and $|m| \leq o$, $r = \sqrt{n_1^2 + n_2^2}$ is distance from the centre and $\theta = tan^{-1}(\frac{n_2}{n_1})$ is the angle of deviation with respect to the n_1 axis. $R_{o,m}$ is referred to as the real valued radial polynomials which are given by:

$$R_{o,m}(r) = \sum_{s=0}^{\frac{o-|m|}{2}} (-1)^s \frac{(o-s)!}{s!(\frac{o+|m|}{2}-s)!(\frac{o-|m|}{2}-s)!} r^{o-2s} \tag{4.26}$$

based on above definitions, the complex Zernike moments with order o and m iterations of a function $x(r,\theta)$ are defined as:

$$S_{o,m} = \frac{o+1}{\pi} \int_0^{2\pi} \int_0^1 x(r,\theta) Z^*_{o,m}(r,\theta) r dr d\theta \tag{4.27}$$

where the asterisk denotes complex conjugation. The discrete approximation of the continuous Zernike integral based on Eq. (4.27) for image function $\mathbf{x}[n_1, n_2]$ with spatial dimension $N_1 \times N_2$ written as follows:

$$x^Z_{o,m} = \frac{o+1}{\pi} \sum_{n_1 \in C_r} \sum_{n_2 \in C_r} \mathbf{x}[n_1, n_2] \; Z^*_{o,m}[n_1, n_2] \tag{4.28}$$

where $C_r = \{[n_1, n_2] | n_1^2 + n_2^2 \leq 1\}$. Figure 4.16 shows an example of feature extraction from face image.

Figure 4.16: Example of ZM for feature extraction with different orders and repetitions.

The list of the first 10-order Zernike moments is given in Table 4.1. The

feature vector, $\mathbf{x}^f_{\text{Zernike}}$ is given by

$$\mathbf{x}^f_{\text{Zernike}} = \{x^Z_{o,m} | \, 0 \leq o \leq 10, \, 0 \leq m \leq 10\} \tag{4.29}$$

Table 4.1: The first 10-order Zernike moments.

Order	Dimensionality	Zernike moments
0	1	$x^Z_{0,0}$
1	2	$x^Z_{1,1}$
2	4	$x^Z_{2,0}$, $x^Z_{2,2}$
3	6	$x^Z_{3,1}$, $x^Z_{3,3}$
4	9	$x^Z_{4,0}$, $x^Z_{4,2}$, $x^Z_{4,4}$
5	12	$x^Z_{5,1}$, $x^Z_{5,3}$, $x^Z_{5,5}$
6	16	$x^Z_{6,0}$, $x^Z_{6,2}$, $x^Z_{6,4}$, $x^Z_{6,6}$
7	20	$x^Z_{7,1}$, $x^Z_{7,3}$, $x^Z_{7,5}$, $x^Z_{7,7}$
8	25	$x^Z_{8,0}$, $x^Z_{8,2}$, $x^Z_{8,4}$, $x^Z_{8,6}$, $x^Z_{8,8}$
9	30	$x^Z_{9,1}$, $x^Z_{9,3}$, $x^Z_{9,5}$, $x^Z_{9,7}$, $x^Z_{9,9}$
10	36	$x^Z_{10,0}$, $x^Z_{10,2}$, $x^Z_{10,4}$, $x^Z_{10,6}$, $x^Z_{10,8}$, $x^Z_{10,10}$

4.8 HLACLF Methodology

The HLAC features [102] are primitive image features based on Eq. (4.23) and they are too complex to extract the features from images. The original HLAC features are represented by mask pattern with different displacements. In this section, a novel higher-order local autocorrelation like features (HLACLF) is proposed for FER system. The implementation of HLACLF method is quite simple and it has better feature representation than other feature extraction method for facial expression recognition. The HLACLF function is defined as:

$$\mathbf{x}_{\text{hlaclf}} = \sum_{n_2=W}^{N_2-W} \sum_{n_1=W}^{N_1-W} \mathbf{x}[n_1 \pm a_i, n_2 \pm a_j]; \quad i, j \in \{1, \cdots, d\} \tag{4.30}$$

where $W = \lceil w+1/2 \rceil$, $w \times w$ denotes the window size ($w \in \{3, 5, 7\}$), $\mathbf{x}$ is a face image, $[n_1, n_2]$ are pixel Cartesian coordinates, $[a_i, a_j]$ are 2-D displacement and $N_1 \times N_2$ is the size of image. Using Eq. 4.30, each mask pattern is scanned

over the image from left to right and for each possible position, the product of the pixels marked in white is computed. All the products corresponding to a mask are then summed so as to produce a feature vector. An example is given in Figure 4.17.

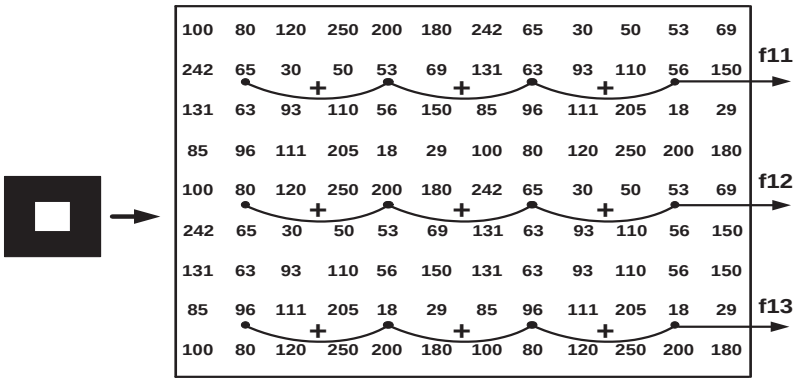


Figure 4.17: Feature vector constructed based on 0^{th}-order HLACLF.

This operation is performed using different mask patterns to create the feature vector for each facial image. Since the HLACLF features use the information of two-dimensional distributions as well as the directions, they analyze an image more closely. Furthermore, large mask patterns are used to support large displacement regions [Figure 4.15(b)] and extract the features of low resolutions. Therefore, different sizes of masks are used to construct multi-resolution features.

4.9 Hybrid LGFCT Method

The hybrid feature extraction method consists of two phases [103]. The first part is to extract required information from the magnitudes of Log-Gabor filters (LGF) and stored in a matrix form. Then, the multiresolution features for different LP levels are calculated using contourlet transform (CT). The feature vector $\mathbf{x}$ is created by concatenating both LGF coefficients $\mathbf{x}_{\text{LGF}}$ and CT coefficients $\mathbf{x}_{\text{CT}}$ for every image in database. The block diagram of LGFCT method is shown in Figure 4.18.

$$\mathbf{x} = \mathbf{x}_{\mathrm{LGF}} \bigcup \mathbf{x}_{\mathrm{CT}} \tag{4.31}$$

where $\bigcup$ denotes the join between two feature vectors.

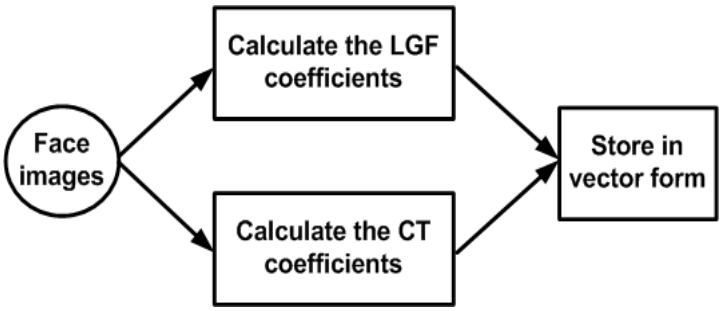


Figure 4.18: The block diagram of LGFCT method to extract the features.

4.10 Hybrid Face Region Method

The hybrid face recognition method (HFR) extracts features from the part of and whole face image using Log-Gabor filters. A problem that is frequently encountered in FER task is partial occlusions. Occlusions can introduce errors into the predicted expression or result in an incorrect expression being transferred to a virtual head. One type of partial occlusion is a temporary occlusion caused by a part of the face being obscured momentarily by an object or as a result of a person moving their head so that not all features of the face can be captured by a camera. Another type of occlusion is a systematic occlusion, which can be caused by a person wearing something such as a head-mounted display, which causes the features of the upper half of the face to be invisible. These types of occlusions are potentially more damaging since they result in whole features of relevance to judging facial expression being obscured. To create the features vector based on HFR method, the features are extracted from eyes and mouth (Chapter 3, Section 3.4) and concatenate them with whole face features [Eq. (4.32)].

$$\mathbf{x}_{\mathrm{HFR}} = \mathbf{x}_{\mathrm{Eyes}} \bigcup \mathbf{x}_{\mathrm{Mouth}} \bigcup \mathbf{x}_{\mathrm{Face}} \tag{4.32}$$

where $\mathbf{x}_{\text{Eyes}}, \mathbf{x}_{\text{Mouth}}$ and $\mathbf{x}_{\text{Face}}$ are the feature vector for eyes image, mouth images and whole face image respectively. This process results in prohibitively large number of the feature arrays. For large training and testing sets, the computations are highly impractical. In order to improve the computational efficiency, it is critical to reduce the features dimensions. This is achieved through feature selection process.

4.11 Summary

In this chapter, several state-of-the-arts feature extraction algorithms including Gabor filters, Log-Gabor filters, contourlet transform, LBP operators and Zernike moments are investigated. All methods are employed to FER system using different databases. The ZM is presented in Section 4.7. This method is employed on facial images with certain noise and rotation. The ZM is robust in terms of recognition rate for images with different noises and rotations. The complete results will be given in Chapter 6. A novel HLACLF technique is presented in Section 4.8. The HLACLF locally extracts the features based on the correlation between the pixels. Section 4.9 introduces another novel LGFCT feature extraction method. The LGFCT combines both the LGF and CT coefficients for each facial image. The advantage of this method is that it extracts all the features in different scales and orientations, however, it is more complex than other feature extraction method. A novel HFR based on face regions is presented in Section 4.10. This method extracts the features from two parts of the face, i.e., eyes and mouth and combines them with the features of the whole face. The performance assessment results of aforementioned new methods and techniques will be shown in Chapter 6. Since the extracted features need further processing, it is not possible to decide at this stage which particular method is more suitable to solve the FER problem. Furthermore, most of extracted features are of high dimensions and they are not suitable for subsequent process by classifier. Therefore, the feature selection module

needs to be used before classification. Next chapter will discuss several feature selection methods.

Chapter 5

Feature Selection

5.1 Introduction

This chapter will introduce the feature selection module. By reducing the number of features, one can both reduce overfitting of learning methods and increase the computation speed of the prediction process [20]. As mentioned before, feature selection methods can be classified into two types, i.e., filters and wrappers [51, 104]. The first kind is classifier agnostic, as they are not dedicated to a specific type of classification method. On the contrary, the wrappers rely on the performance of one type of classifier to evaluate the quality of a set of features. The feature selection stage has an effect on both the computational complexity and the quality of the classification results. It is essential that the information contained in the selected features is sufficient to correctly determine the input class. Too many features may unnecessarily increase the complexity of the training and classification tasks, whereas a poor, or insufficient, selection of features may have a detrimental effect on the classification results. Feature selection methods that are adequate for simple distributions of patterns belonging to different classes fail in classification tasks with more complex distributions and overlapping boundaries. Methods based on correlation assume linear dependencies between data and cannot handle arbitrary relations between the pattern coordinates and the different classes. The main advantages of feature selection are as follow [20]:

- Dimension reduction to reduce the computational cost.
- Reduction of noise to improve the classification accuracy.
- More interpretable features or characteristics that can help identify and monitor the function types.

This thesis examines different feature selection algorithms (FSA) and both filters and wrappers methods are implemented for feature selection [20, 51].

5.2 Feature Selection Algorithm

A feature selection algorithm (FSA) is a computational solution that is motivated by a certain definition of relevance. However, the relevance of a feature as seen from the inductive learning perspective may have several definitions depending on the objective that is looked for. An irrelevant feature is not useful for induction, but not all relevant features are necessarily useful for induction. Several fundamental algorithms found in the literature are used to assess their performance in controlled scenario. To this end, a measure to evaluate FSAs is proposed that takes into account the particularities of relevance, irrelevance and redundance on the sample data set. This measure computes the degree of matching between the output given by a FSA and the known optimal solution. The particularities to be evaluated in FSA algorithm are as follow [105, 106]:

- Relevance : Different families of problems are generated by varying the number of relevant features. These are features that, by construction, have an influence on the output and whose role cannot be assumed by the rest (i.e., there is no redundance).
- Irrelevance : Irrelevant features are defined as those features not having any influence on the output and whose values are generated at random for each example. For a problem with relevant features, different numbers of irrelevant features are added to the corresponding data sets. Thus providing with several subproblems for each choice of relevant features).
- Redundance: In these experiments, redundance exists whenever a fea-

ture can take the role of another (perhaps the simplest way to model redundance). This is obtained by choosing a relevant feature randomly and replicating it in the data set. For a problem with relevant features, different numbers of redundant features are added in a way analogous to the generation of irrelevant features.

The characterization of FSAs is considered in several state-of-the-art literatures [105, 106, 107, 108]. In view of them, it is possible to describe this characterization as a search problem in the hypothesis space as follows:

- Search Organization: General strategy with which the space of hypothesis is explored. This strategy is in relation to the portion of hypothesis explored with respect to their total number.
- Generation of Successors: Mechanism by which possible variants (successor candidates) of the current hypothesis are proposed.
- Evaluation Measure: Function by which successor candidates are evaluated, allowing to compare different hypothesis to guide the search process.

5.2.1 General Feature Selection Schemes

The relationship between a FSA and the inducer chosen to evaluate the usefulness of the feature selection process can take three main forms: embedded, filter and wrapper.

- Embedded approach : The inducer has its own FSA (either explicit or implicit). The methods to induce logical conjunctions provide an example of this embedding. Other traditional machine learning tools such as decision trees or artificial neural networks are included in this scheme [109].
- Filter approach : If the feature selection process takes place before the induction step, the former can be seen as a filter of non-useful features prior to induction. In a general sense, it can be considered as a particular case of the embedded scheme in which feature selection is used as

a pre-processing. The filter schemes are independent of the induction algorithm.

- Wrapper approach : In this scheme, the relationship is taken the other way around, i.e., it is the FSA that uses the learning algorithm as a subroutine [110]. The general argument in favour of this scheme is to neutralize the bias of both the FSA and the learning algorithm that will be used later on to assess the goodness of the solution. The main disadvantage of the approach is the computational burden that comes from calling the induction algorithm to evaluate each subset of considered features.

5.3 Filter approach

The earliest approaches to feature selection within machine learning were filter methods. Filter methods use statistical properties of the variables to filter out poorly informative variables. This is done before applying any classification algorithm. For instance, singular value decomposition (SVD) or independent component analysis (ICA) remain popular methods to limit the dimension of signals, but these two methods do not always yield relevant selection of variables [111]. Superposition of several efficient filters has been proposed to remove irrelevant and redundant features and the use of a combinatorial feature selection algorithm has provided results achieving significant reduction of dimensions preserving good accuracy of classification algorithms on real life problems of image processing [112]. A mixture model with an information gain criterion and a Markov Blanket Filtering method have proposed to reach very low dimensions [113]. In the following subsections, two different feature selection methods are described based on filter approach using mutual information [53, 21].

5.3.1 Mutual Information

Information theory provides theoretical tools to quantify the uncertainty of random quantities, or how much information is shared by a few of them [53]. The most fundamental concept in information theory is the entropy of a random variable, which quantifies the uncertainty of the variable. The conditional entropy quantifies the remaining uncertainty of two variables when one of them is known. If these variables are independent, the conditional entropy is equal to the entropy itself.

Mutual information is considered as a good indicator of relevance between two random variables [53]. Efforts to adopt mutual information in feature selection problem resulted in a series of research publications [53, 21]. Because the computation of mutual information between continuous variables is a very difficult job requiring probability density functions (pdf) and involving integration of those functions, mutual information feature selection (MIFS) [53, 114] and its variants used histograms in approximating the PDFs to avoid these complexities. Thus, the performance can be degraded as a result of large errors in estimating the mutual information. In addition, MIFS methods have another limitation in that these methods do not provide a direct measure to judge whether to add additional features or not. More direct calculation of mutual information is attempted using the quadratic mutual information in the feature transformation field, but the relationship between Shannon's mutual information and the quadratic mutual information is not clear so far [53].

The mutual information represents a measure of information found commonly in two discrete random variables say $\mathbf{V_t}$ and $\mathbf{V_s}$ and it is given as [53]:

$$I(\mathbf{V_t};\mathbf{V_s}) = \sum_{\mathbf{v_t}\in\mathbf{V_t}} \sum_{\mathbf{v_s}\in\mathbf{V_s}} p(\mathbf{v_t},\mathbf{v_s}) \log \frac{p(\mathbf{v_t},\mathbf{v_s})}{p(\mathbf{v_t})p(\mathbf{v_s})} \tag{5.1}$$

and

$$\mathbf{v_t} = [v_t^1, v_t^2, \cdots, v_t^N]^T, \quad \mathbf{v_s} = [v_s^1, v_s^2, \cdots, v_s^N]^T \tag{5.2}$$

where $\mathbf{v_t}$ and $\mathbf{v_s}$ are instances of the discrete random variable $\mathbf{V_t}$ and $\mathbf{V_s}$, respectively. $p(\mathbf{v_t}, \mathbf{v_s})$ is the joint probability distribution function (PDF) of $\mathbf{V_t}$ and $\mathbf{V_s}$, $p(\mathbf{v_t})$ and $p(\mathbf{v_s})$ are the marginal PDFs of $\mathbf{V_t}$ and $\mathbf{V_s}$, respectively, $1 \leq t \leq N_f$, $1 \leq s \leq N_f$ and N_f is the input dimensionality which equals the number of features in the dataset. The MI can be also expressed in terms of the entropy:

$$I(\mathbf{V_t}; \mathbf{V_s}) = H(\mathbf{V_t}) - H(\mathbf{V_t}|\mathbf{V_s}) \tag{5.3}$$

$$H(\mathbf{V_t}) = -\sum_{\mathbf{v_t} \in \mathbf{V_t}} p(\mathbf{v_t}) \log p(\mathbf{v_t}); \tag{5.4}$$

$$H(\mathbf{V_t}|\mathbf{V_s}) = -\sum_{\mathbf{v_t} \in \mathbf{V_t}} \sum_{\mathbf{v_s} \in \mathbf{V_s}} p(\mathbf{v_t}, \mathbf{v_s}) \log\ p(\mathbf{v_s}|\mathbf{v_t}) \tag{5.5}$$

where $H(\mathbf{V_t})$ is the entropy of a random variable $\mathbf{V_t}$ and $H(\mathbf{V_t}|\mathbf{V_s})$ is the conditional entropy. Before the feature selection started, each feature vector $\mathbf{v}[n] = [v_1, v_2, \cdots, v_n]^T$ is converted to a binary-valued vector using the following formula:

$$\mathbf{v}[n] = \begin{cases} 1 & v_n \geq \tau \\ 0 & v_n < \tau \end{cases} \tag{5.6}$$

In Eq. (5.6), v_n is the n^{th} coordinate of the feature vector f and τ is an arbitrary threshold, which is calculated as a median value of the feature vector coordinates.

Given an initial feature space $\mathcal{F}$ with N_F feature vectors and a set of classes, $C = \{c_1, c_2, \cdots, c_k\}$, the aim of MI feature selection technique is to find the subset $\mathcal{S} \subset \mathcal{F}$ with $N_S << N_F$ feature vectors that maximize the mutual information $I(C; \mathbf{S})$ between the selected feature set S and the class set. The mutual information feature selection (MIFS) algorithm, proposed by Battiti in [53], is applied to perform the feature selection. In this approach, starting from an empty set, the best available feature vectors are added, one by one to the selected feature set, until the size of the set reaches the desired

value of N_S. The algorithm proceeds in the following steps:

Step 1 - Initialization.

- Set $\mathcal{F} \longleftarrow$ "initial set of N_F features", $\mathcal{S} \longleftarrow$ "empty set".

Step 2 - Computation of the MI with the output class.

- $\forall \mathbf{V}_i \in \mathcal{F}$, $1 \leq i \leq N_F$, compute $I(\mathbf{V}_i; C)$ by

$$I(\mathbf{V}_i; C) = H(C) - H(C|\mathbf{V}_i) \tag{5.7}$$

where

$$H(C) = -\sum_{k=1}^{N_c} p(c_k) \log(p(c_k)) \tag{5.8}$$

$$H(C|\mathbf{V_i}) = -\sum_{k=1}^{N_c} \sum_{v_i \in \mathbf{V_i}} p(c_k, \mathbf{v_i}) \log(p(c_k|\mathbf{v_i})) \tag{5.9}$$

where $H(C)$ is the entropy of C, $H(C|\mathbf{V_i})$ is the conditional entropy of C on $\mathbf{V_i}$ and N_c is the number of classes (six expressions, $N_c = 6$).

Step 3 - Selection of the first feature vector.

- Set the iteration number $j \longleftarrow 1$ and find the feature vector $\mathbf{V}_d$ such that:

$$\mathbf{V}_d = \arg\max_{\mathbf{V}_i} \{I(\mathbf{V}_i; C)\} \tag{5.10}$$

- Set $\mathbf{S}_j \longleftarrow \mathbf{V}_d$, $\mathcal{S} \longleftarrow \{\mathbf{S}_j\}$ and $\mathcal{F} \longleftarrow (\mathcal{F} - \mathcal{S})$

Step 4 - Selection feature vectors.

- Set $j \longleftarrow j+1$ and repeat until the size of $\mathcal{S}$ is equal to the desired value of N_S.
- $\forall \mathbf{V}_i \in \mathcal{F}$, compute the joint MI between variables, $I(C; \mathbf{V}_i|\mathbf{S}_j)$, where

$$I(C; \mathbf{V}_i|\mathbf{S}_j) = I(C; \mathbf{V}_i) - \beta \sum_{i,j} I(\mathbf{V}_i; \mathbf{S}_j) \tag{5.11}$$

- Find the feature vector such that

$$\mathbf{V}_d = \arg\max_{\mathbf{V}_i} [I(C; \mathbf{V}_i|\mathbf{S}_j)] \tag{5.12}$$

- Set $\mathbf{S}_j \longleftarrow \mathbf{V}_d$, $\mathcal{S} \longleftarrow \{\mathbf{S}_1, \cdots, \mathbf{S}_j\}$ and $\mathcal{F} \longleftarrow (\mathcal{F} - \mathcal{S})$,

Step 5 - Output the set $\mathcal{S}$ containing the selected features.

The first term $I(C; \mathbf{V}_i)$ in Eq. (5.11), represents the mutual information between the feature vector $\mathbf{V}_i$ and the class labels C. The second term $\sum_{i,j} I(\mathbf{V_i}; \mathbf{S_j})$ represents the sum of mutual information between the selected feature vector $\mathbf{V}_i$ and each of the feature vectors that belong to the selected sub-set of feature vectors $\mathbf{S}_j$. In other words, the feature selection formula in Eq. (5.11) indicates that the sub-set $\mathcal{S}$ of feature vectors is selected through a simultaneous maximization of the mutual information between the selected feature vectors in $\mathcal{S}$ and the class labels C and minimization of the mutual information between the selected feature vectors within $\mathcal{S}$. As a result, an optimal sub-set $\mathcal{S} \subset \mathcal{F}$ of mutually independent and highly representative feature vectors is obtained. The amount of independence between selected feature vectors is controlled by the constant value β.

5.3.2 MRMR Criteria

So far, the number of features retained in the feature set is set by human intuition with trial-and-error, although there are studies on selecting features based on certain assumptions on data distributions [21]. A deficiency of this simple ranking approach is that the features could be correlated among themselves [21], i.e., if the feature is ranked high for the classification task, other features highly correlated with that feature are also likely to be selected by the filter method. It is frequently observed that simply combining a very effective feature with another very effective feature often does not form a better feature set. One reason is that these two features could be highly correlated. This raises the issue of "redundancy" of feature set. The fundamental problem with redundancy is that the feature set is not a comprehensive representation of the characteristics of the target phenotypes. There are two aspects of this problem [21]:

1. Efficiency: If a set of features contains quite a number of mutually

highly correlated features (i.e., 200), the true independent or representative features are most likely much fewer (i.e., 50). The other highly correlated features can ignored without effectively reducing the performance of the prediction, which implies that 150 features in the set are essentially wasted.

2. Broadness: Because the features are selected according to their discriminative powers, they are not maximally representative of the original space covered by the entire dataset.

The feature set may represent one or several dominant characteristics of the target phenotypes, but these could still be narrow regions of the relevant space. Thus, the generalization ability of the feature set could be limited. Based on these observations, it's proposed to expand the representative power of the feature set by requiring that features are maximally dissimilar to each other, for example, their mutual Euclidean distances are maximized, or their pair-wise correlations are minimized. These minimum redundancy criteria are supplemented by the usual maximum relevance criteria such as maximal mutual information with the target phenotypes. Therefore, this approach is called the minimum redundancy-maximum relevance (MRMR) approach. The benefits of this approach can be realized in two ways:

1. With the same number of features, the MRMR feature set is to be more representative of the target phenotypes, therefore leading to better generalization property.
2. Using smaller MRMR feature set to effectively cover the same space as a larger conventional feature set does.

In this subsection, the importance of minimum redundancy is pointed out in feature selection and it's provided a comprehensive study.

Mutual Information Quotient

A feature selection method based on the mutual information quotient (MIQ) [21] criterion is investigated. If a feature vector has expressions randomly or

uniformly distributed in different classes, its mutual information with these classes is zero. If a feature vector is strongly different from other features for different classes, it should have large mutual information. The idea of minimum redundancy is to select the feature vectors such that they are mutually maximally dissimilar. Minimal redundancy will make the feature set a better representation of the entire data set. Let $\mathcal{F}$ denotes the feature space, $\mathcal{S}$ denotes the desired subset of features , C denotes a set of classes $C = \{c_1, c_2, \cdots, c_k\}$ and $\mathbf{v_s}$ denotes the vector of N observations for that feature

$$\mathbf{v_s} = [v_s^1, v_s^2, \cdots, v_s^N]^T \tag{5.13}$$

where $\mathbf{v_s}$ is an instance of the discrete random variable $\mathbf{V_s}$. The minimum redundancy condition is given by:

$$R_{\min} = \arg\min_{V_s}\{f_{\text{redundancy}}\} \tag{5.14}$$

and

$$f_{\text{redundancy}} = \{\frac{1}{|\mathcal{S}|^2}\sum_{\mathbf{V_u},\mathbf{V_s}\in\mathcal{S}} I(\mathbf{V_u};\mathbf{V_s})\} \tag{5.15}$$

where $I(\mathbf{V_u};\mathbf{V_s})$ is the mutual information between $\mathbf{V_u}$ and $\mathbf{V_s}$, $|\mathcal{S}|$ is the number of features in $\mathcal{S}$ and $\bar{\mathcal{S}}$ is the complement feature subset of $\mathcal{S}$. To measure the level of discriminant powers of features when they are differentially expressed for different targeted classes, the mutual information $I(\mathbf{V_u};C)$ between targeted classes and features are used to quantify the relevance of $\mathbf{V}_u$ for the classification task. Therefore, the maximum relevance condition is to maximize the total relevance of all features in S is defined by :

$$R'_{\max} = \arg\max_{\mathbf{V_u}}\{f_{\text{relevance}}\} \tag{5.16}$$

and

$$f_{\text{relevance}} = \frac{1}{|\mathcal{S}|}\sum_{\mathbf{V_u}\in S} I(\mathbf{V_u};C) \tag{5.17}$$

The minimum redundancy, maximum relevance feature set is obtained by optimizing the conditions in Eqs. (5.14) and (5.16) simultaneously. Optimization of these two conditions requires combining them into a single criterion function as follows:

$$\mathbf{V_d} = \arg\max_{\mathbf{V_u}} \{ \frac{f_{\text{relevance}}}{f_{\text{redundancy}}} \} \tag{5.18}$$

In this algorithm, the first feature is selected according to Eq. (5.16), i.e., the feature with the highest $I(\mathbf{V_u}; C)$. The rest of the features are selected in an incremental way, i.e., earlier selected features remain in the feature set. Suppose M features are selected for the set $\mathcal{S}$, the additional features are selected from the set $\bar{\mathcal{S}}$. The MI between both features and class label are optimized based on following two conditions:

$$\max_{\mathbf{V_t} \in \bar{\mathcal{S}}} I(\mathbf{V_t}; C), \quad \min_{\mathbf{V_t} \in \bar{\mathcal{S}}} \frac{1}{|\mathcal{S}|} \sum_{\mathbf{V_s} \in \mathcal{S}} I(\mathbf{V_t}; \mathbf{V_s}) \tag{5.19}$$

Based on Eq.(5.19), the features ($\mathbf{V_d}$) for desired feature subset, $\mathcal{S}$, of the form $\langle \mathcal{S}; c \rangle$ where $\mathcal{S} \in \mathcal{F}$ and $c \in C$ are selected based on solution of following problem:

$$\mathbf{V_d} = \arg\max_{\mathbf{V_t}} \{ \frac{I(\mathbf{V_t}; C)}{\frac{1}{|\mathcal{S}|} \sum I(\mathbf{V_t}; \mathbf{V_s})} \}, \ \mathbf{V_t} \in \bar{\mathcal{S}}, \mathbf{V_s} \in \mathcal{S} \tag{5.20}$$

where $\bar{\mathcal{S}}$ is the complement feature subset of $\mathcal{S}$, $|\mathcal{S}|$ is the number of features in subset S and $I(\mathbf{V_t}; \mathbf{V_s})$ is the MI between the candidate feature ($\mathbf{V_t}$) and the selected feature ($\mathbf{V_s}$). Based on Eq. (5.20), the MI between selected feature and intra-class features is maximized whereas the MI between the selected feature and inter-class features are minimized respectively. These features are used for expression classification.

5.4 Wrapper approach

The second approach (i.e., wrapper methods) is computationally demanding, but often is more accurate. A wrapper algorithm explores the space of features

subsets to optimize the induction algorithm that uses the subset for classification. These methods based on penalization face a combinatorial challenge when the set of variables has no specific order and when the search must be done over its subsets since many problems related to feature extraction have been shown to be NP-hard [115]. Therefore, automatic feature space construction and variable selection from a large set has become an active research area. For instance, tree-structured classifiers have been built successively considering statistical properties such as correlations or empirical probabilities in order to achieve good discriminant properties [37, 116]. In a recent work, the mutual information was used to recursively select features and to obtain performance as good as that obtained with a boosting algorithm [32] with fewer variables. Another recursive selection method was constructed to optimize generalization ability with a gradient descent algorithm on the margin of Support Vector classifiers. Another effective approach is the Automatic Relevance Determination (ARD) which introduces a learning hierarchical prior over weights in a Bayesian Network, whose weights connected to irrelevant features are automatically penalized reducing their influence near zero. Finally, a hybrid wrapper and filter approach was reported in [104] to reach highly accurate and selective results that considered an empirical loss function as a shape value and perform an iterative ranking method combined with backward elimination and forward selection.

5.4.1 Optimization Algorithm

Genetic algorithms are adaptive search techniques based on the principles of natural selection in biology [117, 118]. They employ a population of competing solutions evolved over time to converge to an optimal solution. Effectively, the solution space is searched in parallel, which helps in avoiding local optima. For feature selection, a solution is typically a fixed length binary string representing a feature subset. The value of each position in the string represents the presence or absence of a particular feature. The algorithm is an iterative

process where each successive generation is produced by applying genetic operators such as crossover and mutation to the members of the current generation. Mutation changes some of the values (thus adding or deleting features) in a subset randomly. Crossover combines different features from a pair of subsets into a new subset. The application of genetic operators to population members is determined by their fitness (how good a feature subset is with respect to an evaluation strategy). Better feature subsets have a greater chance of being selected to form a new subset through crossover or mutation. In this manner, good subsets are evolved over time. Algorithm 2 shows a simple genetic search strategy.

Algorithm 2 Simple genetic search strategy

1. Begin by randomly generating an initial population P.
2. Calculate $e(x)$ for each member $x \in P$.
3. Define a probability distribution p over the members of P.
4. Select two population members x and y with respect to p.
5. Apply crossover to x and y to produce new population members x' and y'.
6. Apply mutation to x' and y'.
7. Insert x' and y' into P' (the next generation).
8. If $|P'| < |P|$, goto 4.
9. Let $P \longleftarrow P'$.
10. If there are more generations to process, goto 2.
11. Return $x \in P$ for which $e(x)$ is highest.

Initialization of the population is commonly done by seeding the population with random values. The fitness value is proportional to the performance measurement of the function being optimized. The calculation of fitness values is conceptually simple. It can, however, be quite complex to implement in a way that optimizes the efficiency of the GA's search of the problem space. It is this fitness that guides the search of the problem space.

After fitness calculation, the next step is reproduction. Reproduction com-

prises forming a new population, usually with the same total number of chromosomes, by selecting from members of the current population using a stochastic process that is weighted by each of their fitness values. The higher the fitness, the more likely it is that the chromosome will be selected for the new generation. One commonly used way is a "roulette wheel" procedure that assigns a portion of a roulette wheel to each population member where the size of the portion is proportional to the fitness value. This procedure is often combined with the elitist strategy, which ensures that the chromosome with the highest fitness is always copied into the next generation. The next operation is called crossover. To many evolutionary computation practitioners, crossover is what distinguishes a GA from other evolutionary computation paradigms. Crossover is the process of exchanging portions of the strings of two "parent" chromosomes. An overall probability is assigned to the crossover process, which is the probability that is given two parents, the crossover process will occur. The final operation in the typical GA procedure is mutation. Mutation consists of changing an element's value at random, often with a constant probability for each element in the population. The probability of mutation can vary widely according to the application and the preference of the person exercising the GA [119].

Genetic Algorithm For Feature Selection

The GA is a stochastic global search method that mimics the metaphor of natural biological evolution [120]. These algorithms are general purpose optimization algorithms with a probabilistic component that provide a means to search poorly understood, irregular spaces. GA's work with a population of points rather than a single point. Each "point" is a vector in hyperspace representing one potential (or candidate) solution to the optimization problem. A population is, thus, just an ensemble or set of hyperspace vectors. Each vector is called a chromosome in the population. The number of elements in each vector (chromosome) depends on the number of parameters in the op-

timization problem and the way to represent the problem. How to represent the problem as a string of elements is one of the critical factors in successfully applying a GA (or other evolutionary algorithm) to a problem [120]. A global optimization algorithm based on genetic algorithm is adapted and applied in the process of the optimal feature selection for the classification of facial expressions. The block diagram of the wrapper approach is illustrated in Figure 5.1.

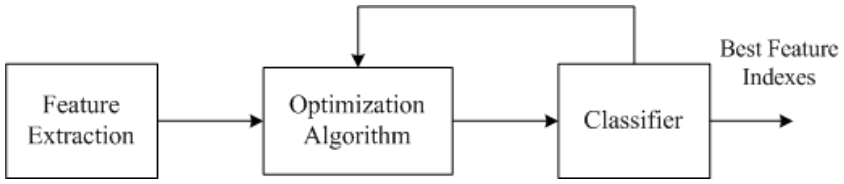


Figure 5.1: Optimal feature selection using the wrapper approach.

The objective function used in the selection process is the classification error produced by classifier. The classification error is iteratively reduced. At each iteration, the optimization procedure is selecting a sub-set of features that are generating smaller classification error compared with the previous iteration. The range of the feature space scanned during the search is decreasing exponentially with the iteration number. At the beginning of the algorithm, a wide range of the feature space is scanned and the algorithm allows selection of features that decrease the objective function value as well as features that give a small increase of the objective value. As the algorithm is coming closer to the final solution, the probability of accepting solutions leading to higher values of the objective function is decreasing rapidly to zero. At the same time, the search range is increasingly confided to a very small space around the final solution. The feature selection diagram is shown in Figure 5.2 based on optimization algorithm.

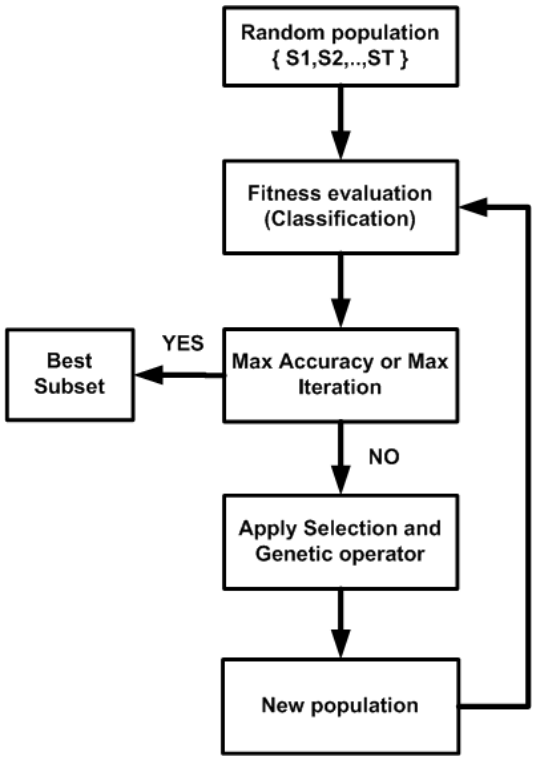


Figure 5.2: Feature selection diagram.

5.5 Summary

In this chapter, the filter and the wrapper feature selection approaches are investigated. The main contribution of this chapter is to employ these methods for FER system. The MIFS and the MIQ algorithms are described in Sections 5.3.1 and 5.3.2. The most discriminative features are selected based on maximizing the MI between intra-class features and selected features, and minimizing the MI between inter-class features and selected features. For wrapper approach, the genetic algorithm is presented in Section 5.4. The features are selected based on GA optimization algorithm. The wrapper approach is more complex than filter approach in terms of implementation and processing time. To evaluate the performance of each feature selection method, the selected features for FER system will be classified using several classifiers. Next chapter describes the classification process and gives the complete experimental results for Facial expression recognition.

Chapter 6

Facial Expression Classification

6.1 Introduction

This chapter deals with the implementation of the classification module. In mathematics, a classification theorem answers the classification problem "What are the objects of a given type, up to some equivalence?". It gives a non-redundant enumeration: each object is equivalent to exactly one class. In pattern recognition, after optimal feature subset is selected, a classifier can be designed using various approaches. Generally, there are three different approaches for classification issues. The first approach is the simplest and the most intuitive approach which is based on the concept of similarity (i.e. template matching). The second one is a probabilistic approach. It includes methods based on Bayes decision rule, the maximum likelihood or density estimator, including KNN, Parzen window classifier, branch-and-bound methods (BnB) and Naive Bayesian classifier (NB). The third approach is to construct decision boundaries directly by optimizing certain error criterion (i.e., Fisher's linear discriminant, MLP, decision tree and SVM). In the following Sections, several classifiers are described that are used for experiments.

6.1.1 NB Classifier

The NB classifier is a probabilistic method that has been shown to be effective in many classification problems [44, 121]. It assumes that the presence (or

lack) of a particular feature of a class is unrelated to the presence (or lack) of any other feature. Let C denotes a set of classes, $C = \{c_1, c_2, \cdots, c_k\}$, and $\mathbf{x}^f = [x_1^f, \cdots, x_m^f]^T$ represent the feature vector, the classification decision is made by:

$$C_{\text{new}} = \arg\max_k \{p(c_k) \prod_{i=1}^{m} p(x_i^f | c_k)\} \tag{6.1}$$

and

$$p(c_k) = \frac{N_s}{N_T} \tag{6.2}$$

where N_s is the number of sample in each class c_k, N_T is the total number of samples, $p(x_i^f | c_k)$ are conditional tables (or conditional density) learned in training using examples and m is the length of feature vector. Despite the independence assumption, NB has been shown to have very good classification performance for many real data sets on par with many more sophisticated classifiers. The proof of the Eq. (6.1) can be found in Appendix B.

6.1.2 Multi-Class LDA Classifier

The linear classifier based on discriminant analysis is used to classify the six different expressions. A natural extension of Fisher Linear Discriminant (FLD) that deals with more than two classes exists [44], which uses multiple discriminant analysis (Appendix C). The projection is from a high dimensional space to a low dimensional space and the transformation sought is the one that maximizes the ratio of intra-class scatter ($\mathbf{S_b}$) to the inter-class ($\mathbf{S_w}$) scatter. The maximization is performed among several competing classes. The $\mathbf{S_b}$ can be viewed as the mean distance between all different classes and $\mathbf{S_w}$ can be regarded as the average class-specific covariance. The intra-class ($\mathbf{S_b}$) and inter-class ($\mathbf{S_w}$) matrices for feature vector ($\mathbf{x}^f$) are given by:

$$\mathbf{S_b} = \sum_{i=1}^{N_c} m_i (\mathbf{x}_{\mu_i}^f - \mathbf{x}_{\mu}^f)(\mathbf{x}_{\mu_i}^f - \mathbf{x}_{\mu}^f)^T \tag{6.3}$$

$$\mathbf{S_w} = \sum_{i=1}^{N_c} \sum_{\mathbf{x}^f \in c_i} (\mathbf{x}^f - \mathbf{x}^f_{\mu_i})(\mathbf{x}^f - \mathbf{x}^f_{\mu_i})^T \tag{6.4}$$

where N_c is the number of classes (i.e., for six expressions, $N_c = 6$), m_i is the number of training samples for each class, c_i is the class label, $\mathbf{x}^f_{\mu_i}$ is the mean vector for each class samples (m_i) and $\mathbf{x}^f_{\mu}$ is total mean vector over all training samples (m) defined by:

$$\mathbf{x}^f_{\mu_i} = \frac{1}{m_i} \sum_{\mathbf{x}^f \in c_i} \mathbf{x}^f \tag{6.5}$$

$$\mathbf{x}^f_{\mu} = \frac{1}{m} \sum_{i=1}^{N_c} m_i \mathbf{x}^f_{\mu_i} \tag{6.6}$$

After obtaining $\mathbf{S_w}$ and $\mathbf{S_b}$, based on Fisher's criterion the linear transformation, $\mathbf{W}_{\text{LDA}}$, can be obtained by solving the generalized eigenvalue (λ) problem:

$$\mathbf{W}^T_{\text{LDA}} \mathbf{S_b} = \lambda \mathbf{W}^T_{\text{LDA}} \mathbf{S_w} \tag{6.7}$$

Once the transformation $\mathbf{W}_{\text{LDA}}$ is given, the classification can be performed in the transformed space based on predefined distance measure such as the Euclidean distance, $||\bullet||$. The new instance, $\mathbf{x}^f_n$, is classified to:

$$c_{\text{n}} = \arg\min_i ||\mathbf{W}_{\text{LDA}} \mathbf{x}^f_{\text{n}} - \mathbf{W}_{\text{LDA}} \mathbf{x}^f_{\mu_i}|| \tag{6.8}$$

where c_{n} denotes the predicted class-label for $\mathbf{x}^f_n$ and $\mathbf{x}^f_{\mu_i}$ is the centroid of i^{th} class.

6.2 Structural Similarity Classifier

6.2.1 Background

In this section, the novel classifier based on the structural similarity metric is presented. The structural similarity (SSIM) index is a method for measuring

the similarity between two images [122]. The SSIM index is a full reference metric, in other words, it measures image quality based on an initial uncompressed or distortion-free image as reference. The SSIM is designed to improve on traditional methods such as the peak signal-to-noise ratio (PSNR) and the mean squared error (MSE), which have proved to be inconsistent with human visual perception. The SSIM was proposed for image quality assessment [122]. The motivation behind the structural similarity approach for measuring image quality is that the Human Visual System (HVS) is not designed for detecting imperfections and "errors" in images. Instead, the HVS has evolved so that it can perform visual pattern recognition in order to be able to extract the structure or connectedness of natural images. Therefore, a useful perceptual quality metric would emphasize the structure of scenes over the lighting effects. The structural similarity approach is mostly insensitive to the distortions that lighting changes may create, i.e., changes in the mean and contrast of an image. However, SSIM is sensitive to distortions that break down natural spatial correlation of an image, such as blur, block compression artifacts and noise. The SSIM index (SM) between two image patches $\mathbf{x}$ and $\mathbf{y}$ is expressed as [122]:

$$SM(\mathbf{x},\mathbf{y}) = \frac{(2\mu_{\mathbf{x}}\mu_{\mathbf{y}} + K_1)(2\sigma_{\mathbf{xy}} + K_2)}{(\mu_{\mathbf{x}}^2 + \mu_{\mathbf{y}}^2 + K_1)(\sigma_{\mathbf{x}}^2 + \sigma_{\mathbf{y}}^2 + K_2)} \tag{6.9}$$

where K_1 and K_2 are two small positive constants, $\mu_{\mathbf{x}}, \mu_{\mathbf{y}}$ are the means of $\mathbf{x}$ and $\mathbf{y}$ and $\sigma_{\mathbf{x}}, \sigma_{\mathbf{y}}$ are the variances of $\mathbf{x}$ and $\mathbf{y}$, respectively. It can be shown that the maximum SSIM index value is 1 when $\mathbf{x} = \mathbf{y}$ [122]. Recent research showed that it is straightforward to implement a structural similarity metric in other domains [123, 124]. For instance, complex wavelet structural similarity (CWSSIM) and contourlet transform structural similarity (CTSSIM) were recently used for pattern recognition [8, 124]. In the following subsection, the GFSSIM methodology is described and the experimental results are presented as well.

6.2.2 GFSSIM Methodology

In the proposed GFSSIM algorithm, the complex Gabor coefficients are adopted for classification. Since, the Gabor filters are bandpass filters [18], they have no response at zero frequency and the mean of the Gabor coefficients is zero ($\mu_{\mathbf{x}^G} = \mu_{\mathbf{y}^G} = 0$). Accordingly, the terms $(2\mu_{\mathbf{x}^G}\mu_{\mathbf{y}^G} + K_1)$ and $(\mu^2_{\mathbf{x}^G} + \mu^2_{\mathbf{y}^G} + K_1)$ in Eq. (6.9) will be cancelled out. Furthermore, the variance of $\mathbf{x}^G$, $\sigma^2_{\mathbf{x}^G}$, the variance of $\mathbf{y}^G$, $\sigma^2_{\mathbf{y}^G}$ and the covariance between $\mathbf{x}^G$ and $\mathbf{y}^G$, $\sigma_{\mathbf{x}^G\mathbf{y}^G}$, for GFSSIM can be calculated as:

$$\sigma^2_{\mathbf{x}^G} = \frac{1}{M}\sum_{i=1}^{M}(\mathbf{x}_i^G - \mu_{\mathbf{x}^G})^2 \xrightarrow{\mu_{\mathbf{x}^G}=0} \sigma^2_{\mathbf{x}^G} = \frac{1}{M}\sum_{i=1}^{M}(\mathbf{x}_i^G)^2 \tag{6.10}$$

$$\sigma^2_{\mathbf{y}^G} = \frac{1}{M}\sum_{i=1}^{M}(\mathbf{y}_i^G - \mu_{\mathbf{y}^G})^2 \xrightarrow{\mu_{\mathbf{y}^G}=0} \sigma^2_{\mathbf{y}^G} = \frac{1}{M}\sum_{i=1}^{M}(\mathbf{y}_i^G)^2 \tag{6.11}$$

$$\sigma_{\mathbf{xy}} = \frac{1}{M}\sum_{i=1}^{M}(\mathbf{x}_i^G - \mu_{\mathbf{x}^G})(\mathbf{y}_i^G - \mu_{\mathbf{y}^G}) \xrightarrow[\mu_{\mathbf{y}^G}=0]{\mu_{\mathbf{x}^G}=0} \sigma_{\mathbf{xy}} = \frac{1}{M}\sum_{i=1}^{M}\mathbf{x}_i^G\mathbf{y}_i^G \tag{6.12}$$

Now using Eqs. (6.9), (6.10), (6.11) and (6.12), the GFSSIM index (GS) between two given Gabor coefficients $\mathbf{x}^G$ and $\mathbf{y}^G$ (that correspond to image patches $\mathbf{x}$ and $\mathbf{y}$) can be obtained as:

$$GS(\mathbf{x}^G, \mathbf{y}^G) = \frac{2|\sum_{i=1}^{M}\mathbf{x}_i^G\mathbf{y}_i^{G^*}| + K}{\sum_{i=1}^{M}|\mathbf{x}_i^G|^2 + \sum_{i=1}^{M}|\mathbf{y}_i^G|^2 + K} \tag{6.13}$$

where K is a small positive number ($K < 0.1$) and * denotes the complex conjugate. The GFSSIM algorithm for recognizing facial expression images is demonstrated in Algorithm 3. In this algorithm, the GFSSIM between training Gabor features, $\mathbf{x}^G_{c_j}$, where j is the number of samples in each class and c is the class label of each sample and test Gabor features, $\mathcal{X}_{\text{Test}}$, are calculated. Then, a bank of GFSSIM indices (G_{c_j}) are derived, which can be processed to identify the expression. The final step is to find the maximum mean from GFSSIM

indices of six expressions. The index of the maximum mean determines the class c of the test image ($\mathbf{x}_k^G$).

Algorithm 3 GFSSIM algorithm for FER system

Create data set of Gabor coefficients for each expression set ($\mathcal{X}$).
$\mathcal{X} = \{\mathbf{x}_{c_i}^G | 1 \leq i \leq N, 1 \leq c \leq 6, (i,c) \in \mathbb{N}\}$.
Create training set of Gabor coefficients for each expression set ($\mathcal{X}_{\text{Train}}$).
$\mathcal{X}_{\text{Train}} = \{\mathbf{x}_{c_j}^G | 1 \leq j \leq M, 1 \leq c \leq 6\}$.

Create testing set of Gabor coefficients ($\mathcal{X}_{\text{Test}}$).
$\mathcal{X}_{\text{Test}} = \{\mathbf{x}_k^G | k \neq j\}$.

for $c = 1$ to 6 **do**
  for $j = 1$ to M **do**
    Calculate the GFSSIM (G_{c_j}) between each training sample and testing sample.

    $G_{c_j} = GS(\mathbf{x}_{c_j}^G, \mathbf{x}_k^G)$.

    Put G_{c_j} in D_{c_j}
  end for
end for

for $c = 1$ to 6 **do**
  Find the mean of GFSSIM for training samples of each class label
  $\mu_c = \frac{1}{M} \sum_{j \in c}(D_{c_j})$
end for

The minimum of (μ_c) is the class label (c) of test sample.
$c = \arg\min_c(\mu_c)$

6.2.3 Experimental Results

For each image, the face was detected based on Viola-Jones method and scaled to different sizes (16×16, 32×32, 64×64). Thereafter, the features were extracted using 40 Gabor filters and an optimum subset $\mathcal{S}$ was selected based on the MIQ algorithm [21]. This subset features, with a size of 100 (which was the best rate for classification), were used for classification. In this experiment, different values of SNR were used and the noise type was assumed as the salt-and-pepper [7].

For comparison, the SVM classifier with the RBF kernel was applied to the

same setup of the experiments [56]. The average recognition rates for different resolutions are illustrated in Figure 6.1 with different SNRs.

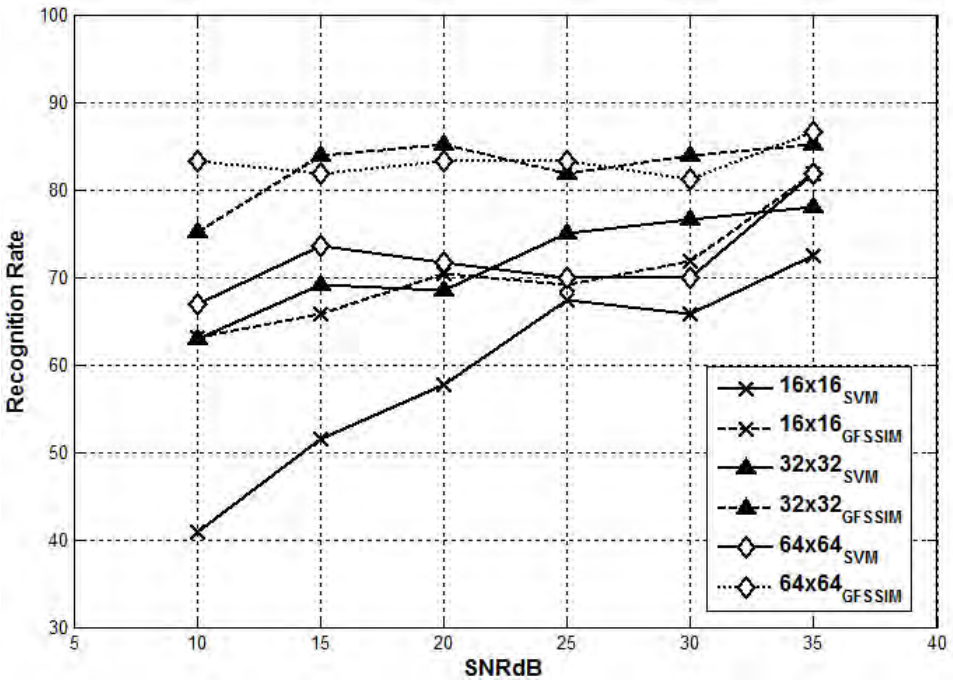


Figure 6.1: Percentage of average recognition rate for different SNRs for mixed Cohn-Kanade and JAFFE database.

The recognition rate for each individual expression is shown in Table 6.1 and Table 6.2. In addition, the average processing times was calculated for SVM and GFSSIM algorithm with CPU speed 2.4 GHz and 2GB RAM. The average processing time to recognize the expression of one sample image is 0.20 second for SVM classifier, whereas the average time for GFSSIM algorithm is about an order-of-magnitude less, specifically, 0.025 second. Furthermore, the proposed algorithm is compared with different state-of-the-art methods including: Gabor filters (8 orientations and 5 scales), LBP operator (dividing the images into 10×10 pixel regions) and HLAC with SVM classifier [9, 12]. The recognition rates are illustrated in Table 6.3 for facial images (128×128 pixels) from Cohn-Kanade database.

As a result, one may conclude that GFSSIM algorithm has remarkable improvement in facial expression recognition in low SNRs. However, for low reso-

Table 6.1: Expression recognition rate based on SVM classifier for different resolutions with SNR = $35dB$ from mixed Cohn-Kanade and JAFFE database.

Resolutions	16×16	32×32	64×64
Anger	70.1%	75.0%	78.2%
Disgust	63.0%	65.0%	74.2%
Fear	70.2%	78.9%	84.0%
Happy	76.6%	83.0%	85.3%
Sad	73.4%	77.2%	74.1%
Surprise	86.1%	88.4%	95.4%

Table 6.2: Expression recognition rate based on GFSSIM classifier for different resolutions with SNR = $35dB$ from mixed Cohn-Kanade and JAFFE database.

Resolutions	16×16	32×32	64×64
Anger	83.4%	87.5%	87.9%
Disgust	73.1%	76.0%	78.7%
Fear	74.7%	76.2%	80.2%
Happy	91.3%	95.8%	96.3%
Sad	77.5%	80.3%	81.0%
Surprise	91.3%	95.4%	96.0%

lution images with higher SNRs (>15 dB), the SVM classifier recognition rate shows some improvements compared with the proposed GFSSIM algorithm. Fortunately, the GFSSIM algorithm possesses lower complexity in addition to shorter processing time compared to the SVM. These advantages make the GFSSIM algorithm much more cost effective and more reliable in transmission of images (video sequences) over noisy communication channels.

Table 6.3: The recognition rate of different state-of-the-art methods for Cohn-Kanade database images (128×128).

Expression	Gabor	LBP	HLAC	GFSSIM
Anger	83.3%	82.2%	86.7%	88.1%
Disgust	89.8%	87.6%	90.6%	91.2%
Fear	85.8%	86.5%	85.2%	87.6%
Happy	96.3%	94.9%	96.2%	96.9%
Sad	85.2%	81.3%	82.5%	81.2%
Surprise	100%	98.1%	98.9%	98.2%
Average	**90%**	**88.4%**	**90%**	**90.5%**

6.3 Expression Recognition Results

Facial expression databases are required to compare the computational efficiency of FER system. In this study, JAFFE and Cohn-Kanade databases [18, 13] are used to train and test the facial expression recognition system. Each test was performed three times using randomly selected testing and training sets and an average result was calculated. The subjects represented in the training set were not included in the testing set of images, thus ensuring a person-independent classification of facial expressions. An automatic face detector [25] and facial detection [11] were used and the faces were also scaled. The tested images were classified using several feature extraction and classifiers. Furthermore, the difficult problem of recognizing an expression in different resolutions was considered.

6.3.1 Gabor Filter Experiments

The 40 Gabor filters are used for feature extraction. Several feature selection methods are used for comparative evaluation. The NB classifier is performed for both JAFFE database (215 images) and Cohn-Kanade database (388 images). Each test is performed 3 times using randomly selected testing and training sets and an average result is calculated. The subjects represented in the training set are not in the testing set. The accuracy results are demonstrated in Table 6.4. From the table, it can be seen that the best performance is given by using the MIQ method, whereas the PCA data reduction has the worst performance for facial expression recognition.

Table 6.4: Percentage of recognition rate for Gabor filter based on different feature selection methods (128×128 pixels).

(a) Cohn-Kanade Database

	AGF*	PCA	MIFS	MIQ	GA
Anger	65.2	64.9	68.2	75.2	69.2
Disgust	64.1	60.2	77.8	78.1	76.1
Fear	66.2	57.8	75.1	79.3	70.1
Happy	82.1	86.3	95.2	96.5	94.4
Neutral	72.1	60.2	73.4	87.1	76.3
Sad	68.3	58.2	75.1	83.2	75.4
Surprise	90.2	89.2	90.3	96.2	93.2
Average	**72.6**	**68.1**	**79.3**	**85.1**	**79.2**

(b) JAFFE Database

	AGF*	PCA	MIFS	MIQ	GA
Anger	75.1	74.6	78.5	81.3	79.1
Disgust	74.5	73.9	77.8	82.3	78.3
Fear	70.8	69.5	75.1	80.5	77.9
Happy	70.7	68.8	95.2	98.6	96.2
Neutral	77.0	70.7	78.2	87.1	80.3
Sad	75.4	71.4	76.8	83.2	77.8
Surprise	96.2	92.3	96.2	98.3	95.9
Average	**77.1**	**74.5**	**82.5**	**87.3**	**83.6**

AGF*: Average Gabor Features

6.3.2 Log-Gabor Filter Experiments

In second experiment, The Log-Gabor filters are used to extract the features from facial expression images and different feature reduction/selection methods are employed. Table 6.5 shows the recognition rate using Log-Gabor filters for feature extraction. 24 Log-Gabor filters are used for feature extraction and NB classifier is used for expression recognition. As a result, Log-Gabor filters have better accuracy than Gabor filters and therefore, they are more effective for recognition expression.

Table 6.5: Percentage of recognition rate for Log-Gabor filter based on different feature selection methods (128×128 pixels).

(a) Cohn-Kanade Database

	ALGF*	PCA	MIFS	MIQ	GA
Anger	66.4	66.1	69.4	76.4	70.4
Disgust	65.3	61.4	79.0	79.3	77.3
Fear	67.4	59.0	76.3	80.5	71.3
Happy	83.3	87.6	96.5	97.8	95.7
Neutral	73.3	61.4	74.6	88.4	77.5
Sad	69.5	59.4	76.3	84.4	76.6
Surprise	91.5	90.5	91.6	97.5	94.5
Average	**73.8**	**69.3**	**80.5**	**86.3**	**80.5**

(b) JAFFE Database

	ALGF*	PCA	MIFS	MIQ	GA
Anger	75.4	74.9	78.8	81.6	79.4
Disgust	74.8	74.2	78.1	82.6	78.6
Fear	71.1	69.8	75.4	80.8	78.2
Happy	71.0	69.1	95.6	99.0	96.6
Neutral	77.3	71.0	78.5	87.4	80.6
Sad	75.7	71.7	77.1	83.5	78.1
Surprise	96.6	92.7	96.6	98.7	96.3
Average	**77.4**	**74.8**	**82.9**	**87.7**	**84.0**

ALGF: Average Log-Gabor Filters

6.3.3 LBP Operator Experiments

In third approach, the features were generated based on LBP operator. The features are selected based on MIQ method. The results for different feature selection methods are shown in Table 6.6. NB classifier is used for classification process.

Table 6.6: Percentage of recognition rate for LBP operator based on different feature selection methods (128×128).

(a) Cohn-Kanade Database

	MIFS	MIQ	GA
Anger	67.0	73.9	68.0
Disgust	76.5	76.8	74.8
Fear	73.8	77.9	68.9
Happy	93.6	94.8	92.8
Neutral	72.1	85.6	75.0
Sad	73.8	81.8	74.1
Surprise	88.8	94.6	91.6
Average	**77.9**	**83.6**	**77.9**

(b) JAFFE Database

	MIFS	MIQ	GA
Anger	75.9	78.6	76.5
Disgust	75.2	79.6	75.7
Fear	72.6	77.8	75.3
Happy	92.1	95.3	93.0
Neutral	75.6	84.2	77.7
Sad	74.3	80.5	75.2
Surprise	93.0	95.1	92.7
Average	**79.8**	**84.4**	**80.9**

6.3.4 HLAC Experiments

For the forth experiment, the features are extracted using HLAC features. Then, the most discriminative features are selected using both mutual information and optimization algorithm. In this experiment, the NB classifier is employed to recognize the expressions. The classification results are shown in Table 6.7.

Table 6.7: Percentage of recognition rate for HLAC features based on different feature selection methods (128×128).

(a) Cohn-Kanade Database

	MIFS	MIQ	GA
Anger	71.5	78.5	72.5
Disgust	81.1	81.4	79.4
Fear	78.4	82.6	73.4
Happy	98.6	99.9	97.8
Neutral	76.7	90.5	79.6
Sad	78.4	86.5	78.7
Surprise	93.7	99.6	96.6
Average	**82.6**	**88.4**	**82.6**

(b) JAFFE Database

	MIFS	MIQ	GA
Anger	81.2	84.0	81.8
Disgust	80.5	85.0	81.0
Fear	77.8	83.2	80.6
Happy	97.8	98.2	97.9
Neutral	80.9	89.8	83.0
Sad	79.5	85.9	80.5
Surprise	98.6	99.1	98.7
Average	**85.2**	**89.3**	**86.2**

6.3.5 HLACLF Experiments

In the last experiment, the features are extracted using HLACLF method and Multi-class LDA classifier is employed for classification. The results illustrate

in Table 6.8 for different feature selection methods. The highest accuracy is 91.8% for Cohn-Kanade database and 93.3% for JAFFE database. It's concluded that the HLACLF and the MIQ algorithm is effective for expression recognition.

Table 6.8: Percentage of recognition rate for HLACLF features based on different feature selection methods.

(a) Cohn-Kanade Database

	MIFS	MIQ	GA
Anger	78.7	86.2	83.4
Disgust	89.2	87.3	89.9
Fear	86.3	86.1	84.4
Happy	98.7	99.7	98.8
Neutral	79.6	93.8	82.6
Sad	81.3	89.8	81.6
Surprise	97.1	99.8	98.9
Average	**87.3**	**91.8**	**88.5**

(b) JAFFE Database

	MIFS	MIQ	GA
Anger	86.1	89.1	87.5
Disgust	85.3	90.1	86.7
Fear	82.5	88.2	86.3
Happy	98.8	99.2	98.9
Neutral	81.7	95.2	83.9
Sad	80.3	91.1	81.3
Surprise	99.6	99.9	99.7
Average	**87.8**	**93.3**	**89.2**

The average classification results are shown in Figure 6.2 for different resolutions from 16×16 to 128×128. As a result, the recognition rate is reduced by changing the scale of the image. The holistic method (Gabor, Log-Gabor filters) have better performance than other method for low resolution images. The local methods like HLAC and LBP have competitive performance in high resolution images. The HLACLF has the highest rate for high resolution images.

Table 6.9 shows the CPU execution times corresponding to different stages of the classification process. The times are given (CPU speed 2.4 GHz and 2GB RAM) for different feature extraction methods.

Table 6.9: Comparison of the CPU execution times (sec).

	Gabor	Log-Gabor	LBP	HLAC	HLACLF
Pre-Processing	1.05				
Feature Extraction	15.5	10.7	0.17	6.7	1.1
Classification	0.56				
Entire Process	17.1	12.31	1.78	8.31	2.71

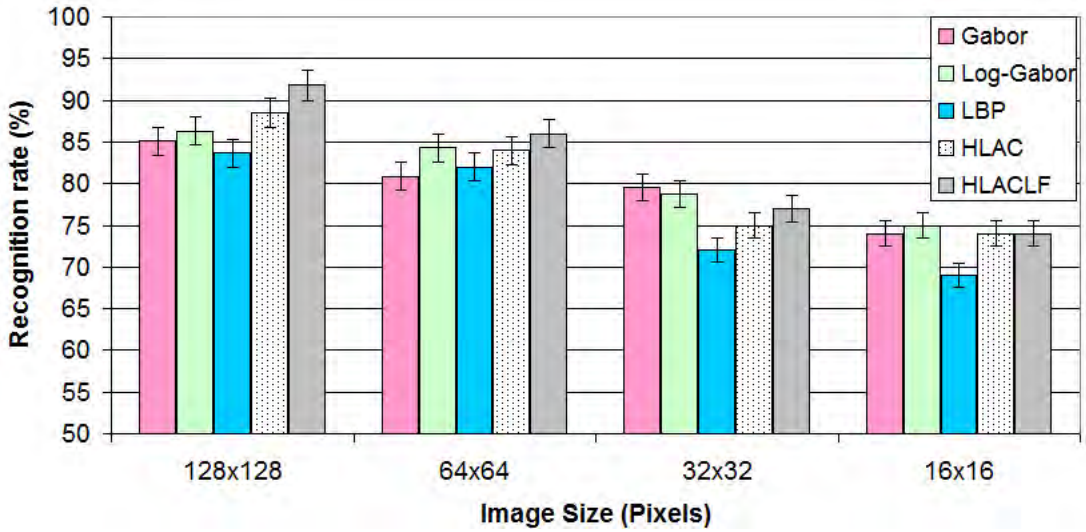


Figure 6.2: Percentage of correct recognition rate for different image resolutions.

6.3.6 Zernike Moments Experiment

Another experiment is using Zernike moments (ZM) as feature extractor. As discussed in Section 4.7, ZM features are robust in presence of the noise and also rotation invariant. Table 6.10 illustrates the recognition rate for different rotations of facial expressions. Due to rotation, each Zernike moment acquires a phase shift, thus the magnitude of a rotated image remains identical to that before rotation. For this study, the LDA classifier is used to recognize the seven expressions. As a result, The maximum rate is ZM order 10 which is 73.2% and after that the recognition rate is reduced by increasing the orders. Therefore, the ZM order 10 is selected for conducting other experiments.

Table 6.10: Percentage of classification of Zernike moments for different orientations and different orders (Cohn-Kanade database).

Order	0^o	30^o	45^o	60^o	90^o	120^o	135^o	150^o	180^o
3	32.4	32.8	31.8	32.4	32.4	31.3	31.8	32.4	32.4
4	42.5	43.6	46.4	44.1	42.5	44.7	46.4	44.1	42.5
5	50.3	52	52.5	51.4	50.3	53.6	52.5	51.4	50.3
6	65.4	65.4	64.3	64.3	65.4	64.3	64.3	64.3	65.4
7	63.7	62.6	61.5	62	63.7	63.7	61.5	62	63.7
8	65.4	63.7	63.1	62.6	65.4	62.7	63.1	62.7	65.4
9	69.3	68.7	66.5	67.6	69.3	67	66.5	67.6	69.3
10	73.2	67	72.1	65.9	73.2	68.2	72.1	66	73.2

Table 6.11: Percentage confusion Table based on 10^{th}-order Zernike moments.

(a) Cohn-Kanade Database

	Anger	Disgust	Fear	Happy	Neutral	Sad	Surprise
Anger	69.2	15.4	0	0	0	15.4	0
Disgust	22.2	70	0	0	0	7.8	0
Fear	0	8.3	83.3	0	0	8.3	0
Happy	5.8	1.9	19.2	71	0	2.1	0
Neutral	1.1	6.9	9.5	0	73.1	9.5	0
Sad	17.9	9.1	0	0	0	73	0
Surprise	4.2	8.3	4.2	2.1	0	8.3	72.9
Average	**73.2**						

(b) JAFFE Database

	Anger	Disgust	Fear	Happy	Neutral	Sad	Surprise
Anger	70.2	15.2	1.2	0	0	13.4	0
Disgust	9.2	78.1	0	0	5.2	7.5	0
Fear	8.2	1.3	77.3	0	7.8	5.4	0
Happy	5.0	3.9	4.1	80.2	5.6	1.2	0
Neutral	4.8	6.2	0	0	79.2	9.8	0
Sad	3.2	7.0	5.8	0	6.5	77.5	0
Surprise	1.0	6.9	0	2.1	4.1	0	85.9
Average	**78.4**						

6.3.7 LGFCT Method Experiments

The tested images are classified using proposed LGFCT feature extraction as described in Section 4.9 of Chapter 4, and LDA classifier. Table 6.12 shows

the recognition rate for seven facial expressions. For comparison, other feature extraction methods, the CT and the LGF are applied to the same classification problems. On average, the LGF gives correct classifications to 86.3% and 87.7% of cases, the CT to 84.1% and 85.2% of cases, whereas the LGFCT method gives the overall correct classification rates of 89.6% and 90.3%. The percentage of correct classifications varied across different facial expressions.

Table 6.12: Percentage of correct classifications based on Hybrid (LGFCT) method.

(a) Cohn-Kanade Database

	Anger	Disgust	Fear	Happy	Neutral	Sad	Surprise
Anger	79.5	4.7	0	0	3.2	12.6	0
Disgust	13.9	86.1	0	0	0	0	0
Fear	3.4	0	87.3	5.1	1.7	2.5	0
Happy	3.0	0	0	97.0	0	0	0
Neutral	0	0	7.2	0	89.8	3.0	0
Sad	7.6	0	0	0	0	92.4	0
Surprise	0	0	0	2.1	2.7	0	95.2
Average	**89.6**						

(b) JAFFE Database

	Anger	Disgust	Fear	Happy	Neutral	Sad	Surprise
Anger	86.5	0.5	4.2	0	3.3	5.5	0
Disgust	2.8	87.1	5.1	0	3.9	1.1	0
Fear	0	3.4	93.5	0	3.1	0	0
Happy	0	2.0	0	95.8	2.2	0	0
Neutral	0	0	0	5.3	91.2	3.5	0
Sad	0	1.9	2.1	0	7.9	88.1	0
Surprise	0	5.4	0	4.5	0	0	90.1
Average	**90.3**						

6.3.8 HFR Method Experiments

The tested images are classified using the Log-Gabor filters for feature extraction and the NB classifier. The features are extracted for different scales and orientations and are tested using NB classifier to choose the best scale and orientation for the Log-Gabor filters. Figure 6.3 illustrates the recognition rate for different four scales and eight orientations using Cohn-Kanade database.

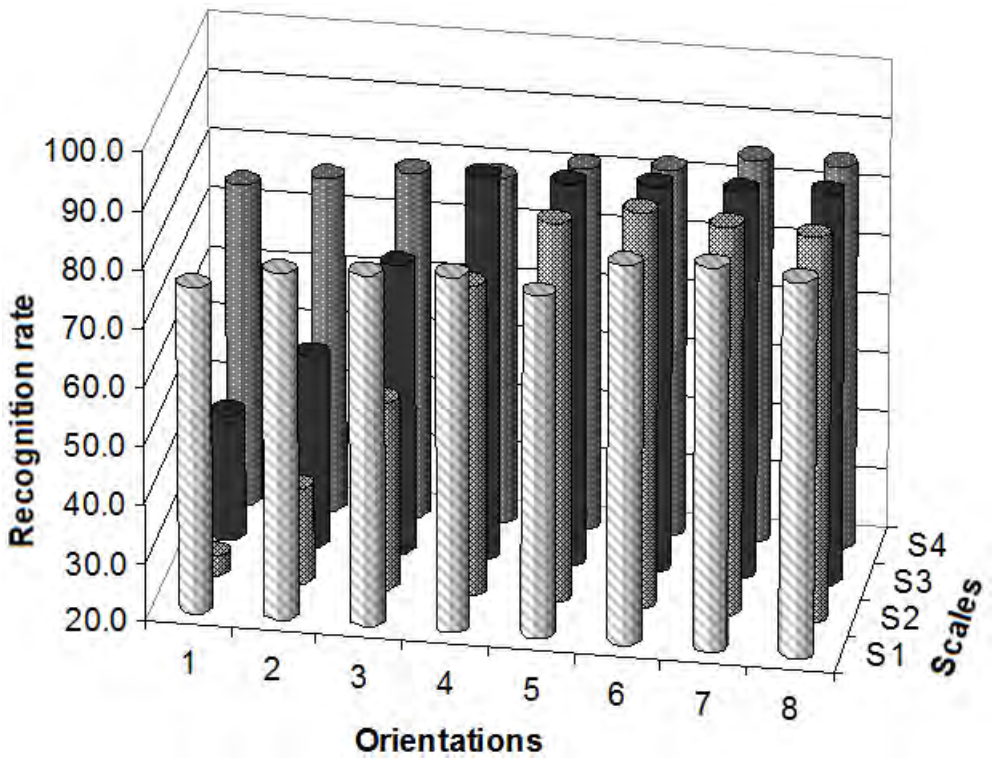


Figure 6.3: Percentage of recognition rate for 4 scales and 8 orientations based on Log-Gabor filters.

As a result, the Log-Gabor filters with 3 scales and 6 orientations are chosen to conduct the experiments. The hybrid face region (HFR) method which is described in Section 4.10 of Chapter 4, has been used to extract the features from face images. The MIFS and the NB classifier are used for feature selection and classification. The recognition rate for different parts of the face using the HFR method is shown in Figure 6.4. The accuracy results are shown in Table 6.13 for both Cohn-Kanade and JAFFE databases based on HFR method using Log-Gabor filter and mutual information. The recognition rate is increased from 87% and 89% based on whole face image to 91.5% and 93.9% based on the HFR method for Cohn-Kanade and JAFFE databases, respectively. Generally, the accuracy is improved nearly 5% for both Cohn-Kanade and JAFFE databases.

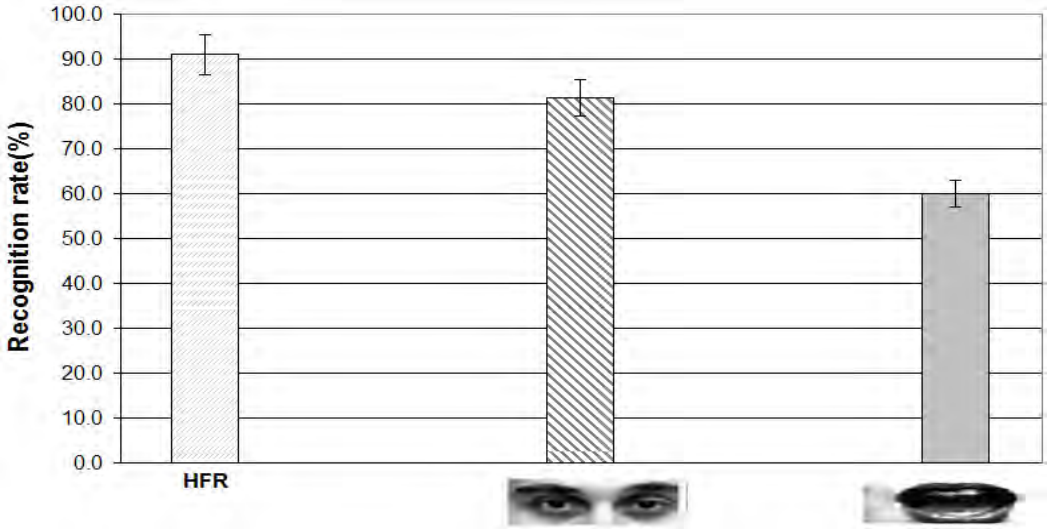


Figure 6.4: The average recognition rate for face components (mouth and eyes) and HFR method.

Table 6.13: Percentage of correct classification using HFR Method.

(a) Cohn-Kanade Database

	Anger	Disgust	Fear	Happy	Neutral	Sad	Surprise
Anger	82.0	9.5	0	0	0	8.5	0
Disgust	8.6	85.1	6.3	0	0	0	0
Fear	0	0	95.5	4.5	0	0	0
Happy	0	0	3.0	97.0	0	0	0
Neutral	0	0	4.5	0	90.3	5.2	0
Sad	8.0	0	0	0	0	92.0	0
Surprise	0	0.8	0	0	0	0	99.2
Average	**91.5**						

(b) JAFFE Database

	Anger	Disgust	Fear	Happy	Neutral	Sad	Surprise
Anger	93.8	0	2.0	0	0	4.2	0
Disgust	0	92.5	3.5	4.0	0	0	0
Fear	2.1	0	95.8	2.1	0	0	0
Happy	0	0	2.5	95.5	2.0	0	0
Neutral	0	0	1.4	0	93.3	5.3	0
Sad	6.4	0	5.2	0	0	88.4	0
Surprise	1.5	0	0	0	0	0	98.5
Average	**93.9**						

6.4 Summary

This chapter presents the experimental results of different approaches for FER systems. Section 6.1 describes the two popular classifiers including the NB classifier and the LDA classifier. Furthermore, a novel SSIM classifier is described in Section 6.2. The experiments conducted on gray-scale facial expression images from Cohn-Kanade and JAFFE databases and the results are presented in Section 6.3. The Log-Gabor filters has better results than Gabor filters in terms of recognition rate for FER system. The performance of LBP operator is the same as that of Gabor filters for high resolutions images. Since the LBP extracts the features locally and for each block, it uses 8 neighbour pixels to extract the features from facial expression image, the recognition rate is less than other methods for low resolution images. The HLAC-Like technique has the best performance in terms of recognition rate. For facial images with certain impulsive noise, a novel GFSSIM technique has remarkable advantages in expression recognition for images with different noise in terms of recognition rate compared with the SVM classifier. Furthermore, the performance of the GFSSIM method outperforms the ZM method for expression recognition. However, the ZM is rotation invariant and it gives best performance in terms of recognition rate for images with different orientations. The results for LGFCT method are presented in Section 6.3.7 and show that the LGFCT has improved the performance of FER system in comparison with the LGF or the CT features. However, the complexity of the FER system is increased using LGFCT method. For the HFR method, the experimental results show that the area of eyes has more information than that of the mouth for facial expressions. Furthermore, the performance of the FER system is improved when combining the whole face and the partial face features together.

This chapter only considers gray-scale facial images for FER system. Next chapter describes the effect of colour information on facial expression recognition and explains a novel framework for colour facial expressions.

Chapter 7

Colour FER System

7.1 Introduction

The current state-of-the-art techniques for facial expression classification mostly focus on gray-scale image features [12], while rarely study the colour image features [6, 62, 71]. A combination of colour feature data may lead to more robust classification results. Recent research reveals that colour information improves face recognition and image retrieval performance [125, 126, 127, 128]. It was first reported in [125] that embedding colour information improves recognition rate when compared with the same scheme using only the luminance information. It was shown in [126] that colour components helped improve face retrieval performance. Liu and Liu proposed a new colour space for face recognition [127]. Young, Man and Plataniotis demonstrated that facial colour cue significantly improved recognition performance using low-resolution face images [128].

Although the researchers showed the improvement of performance by embedding the colour components, the effect of colour information on recognition performance dependent on changes of the light source (e.g., from indoor illumination to outdoor daylight), often making recognition impossible. Therefore, the RGB colour space is not always the most convenient space in which to process colour information. This issue can be addressed using perceptually uniform colour systems [129]. In this chapter, the performance of facial expression

recognition (FER) is comparatively studied based on colour information and multi-linear image analysis in different colour spaces (RGB, YC_bC_r, $CIELab$, $CIELuv$). Furthermore, the performance of FER system in perceptual colour space under slight varying of illumination is investigated and introduced a tensor perceptual colour framework (TPCF). This chapter is organized as follows. Section 7.2 explains the normalization for colour images. Section 7.3 describes different colour spaces. Section 7.4 presents the tensor-based colour facial image and TPCF method. Section 7.5 examines several experiment results. Section 7.6 presents the discussion and Section 7.7 presents the final conclusions.

7.2 Colour Image Normalization

The purpose of colour normalization is to reduce the lighting effect because the normalization process is actually a brightness elimination process [19, 130]. The colour values of face images are normalized with respect to RGB values of the image. Given an input image of $N_1 \times N_2$ pixels represented in the RGB colour space, $\mathbf{x} = \{\mathbf{x}[n_1, n_2, n_3] | \ 1 \leq n_1 \leq N_1, 1 \leq n_2 \leq N_2, 1 \leq n_3 \leq 3\}$, the normalized values, $\mathbf{x}_{\text{norm}}[n_1, n_2, n_3]$, are defined by [130]:

$$\mathbf{x}_{\text{norm}}[n_1, n_2, n_3] = \frac{\mathbf{x}[n_1, n_2, n_3]}{\sum_{n_3=1}^{3} \mathbf{x}[n_1, n_2, n_3]} \tag{7.1}$$

where $\mathbf{x}_{\text{norm}}[n_1, n_2, n_3]$ for $n_3 = 1, 2, 3$ corresponding to red, green and blue (or R, G and B) components of the image, $\mathbf{x}$. It is obvious that

$$\sum_{n_3=1}^{3} \mathbf{x}_{\text{norm}}[n_1, n_2, n_3] = 1 \tag{7.2}$$

Figure 7.2 shows the original and normalized facial expression images in RGB colour space.

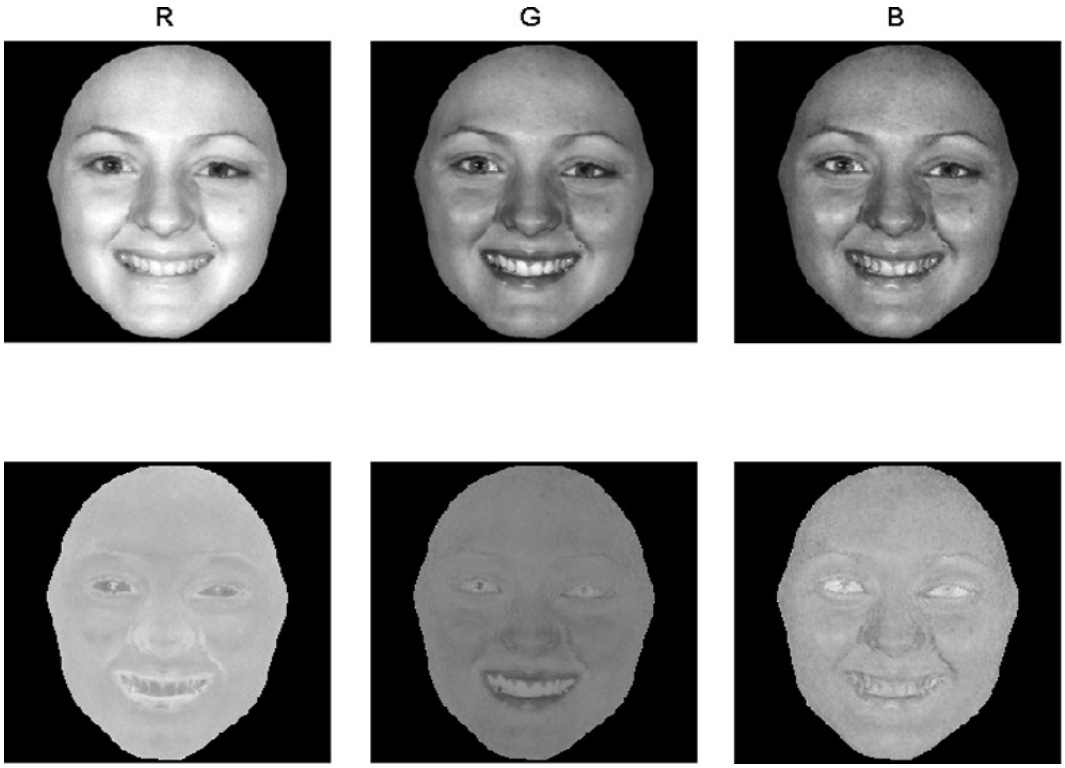


Figure 7.1: The facial expression images, top row, original colour components, bottom row, normalized colour components.

7.3 Colour Spaces

7.3.1 Background

Humans perceive colour as a result of light in the visible region of the spectrum which being projected upon the retina. A typical human eye will respond to wavelengths from about 390 to 750 nanometre (nm) [131] which corresponds to a band in the vicinity of $400-790$ THz in terms of frequency. A light-adapted eye generally has its maximum sensitivity at around 555 nm (540 THz), in the green region of the optical spectrum. Therefore, colour is the brain's reaction to a specific visual stimulus (i.e., an object reflecting light of a certain wavelength). Although colour can be precisely described by measuring its spectral power distribution (SPD), the intensity of the visible electromagnetic radiation at many discrete wavelengths, this leads to a large degree of redundancy.

The human retina has only three types of colour photoreceptor cone cells each of which responds to incident radiation with a somewhat different spectral response curve. These three broad spectral bands roughly correspond to what humans perceive as red, green and blue light. The rod is the fourth type of photoreceptor cell present in the retina. Rods are effective only at extremely low light levels. The signals from these colour sensitive cells (cones), together with those from the rods (sensitive to intensity only), combine in the brain to give "sensations" of different colours [132]. Three numerical components are necessary and sufficient to describe a colour given the appropriate use of spectral weighting functions since there are exactly three types of colour photoreceptors [133]. This is the concern of the science of colorimetry. In 1931, the International Commission on Illumination (CIE) adopted standard curves for a hypothetical standard observer [131, 134]. These curves specify how an SPD can be transformed into a set of three numbers that specifies a colour.

A colour space is a method by which it can specify, create and visualize colour. Colour may be defined by its attributes of brightness, hue and purity for humans. A computer may describe a colour using the amounts of red, green and blue phosphor emission required to match a colour. A printing press may use the reflectance and absorbance of cyan, magenta and yellow inks on the printing paper to generate a specific colour (black ink is also used for gray tones). Thus, a colour is usually specified using three co-ordinates, or parameters (except for printing although technically speaking black is not considered a colour but the absence of a colour). These parameters describe the position of the colour within the colour space being used. They do not tell us what the colour is, that depends on what colour space is being used. There are several image representation models in the colour spaces used for image processing [19]. The following subsections explain the details of colour conversions and perceptual colour spaces.

7.3.2 Colour Space Conversions

The *RGB* (red, green and blue) colour space is the fundamental colour space which are mostly used in image processing and pattern recognition systems. A colour in this space is represented by a triplet of values typically between zero and one and is usually scaled by 255 for an 8-bit representation. Each colour can be broken down into its relative intensity in the three primaries corresponding to the spectral response of one of the three types of cones present in the human eye: red, green and blue. Therefore, This colour space is used to generate other different colour formats. The YC_bC_r colour space is a digital and offset version of the YUV used by the PAL video standard [19]. The conversion functions between RGB and YC_bC_r is defined by:

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 65.481 & 128.553 & 24.966 \\ -37.774 & -74.159 & 111.934 \\ 111.958 & -93.751 & -18.207 \end{bmatrix} \begin{bmatrix} \mathbf{x}_{\text{norm}}[n_1, n_2, 1] \\ \mathbf{x}_{\text{norm}}[n_1, n_2, 2] \\ \mathbf{x}_{\text{norm}}[n_1, n_2, 3] \end{bmatrix} \tag{7.3}$$

$$\begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} + \begin{bmatrix} Y \\ U \\ V \end{bmatrix} \tag{7.4}$$

where $\mathbf{x}_{\text{norm}}[n_1, n_2, n_3]$, $1 \leq n_3 \leq 3$, was defined in Eq. (7.1) and the gray-scale image, $\mathbf{x}_{\text{Gray}}[n_1, n_2]$, is the Y component of YC_bC_r colour space.

7.3.3 Perceptual Colour Spaces

The colour representation in the RGB space is sensitive to the viewing direction, object surface orientation, illumination direction, illumination intensity, illumination colour and inter-reflection. Furthermore, the RGB colour space do not correspond to colour differences as perceived by humans [131]. To overcome these issues, the CIE proposed the perceptually uniform colour spaces. A system is perceptually uniform if a small perturbation to a component value is approximately equally perceptible across the range of that value [134]. In

1976, the CIELab and CIELuv colour spaces were recommended as standards by the CIE [131, 134, 135]. To convert from RGB to perceptual colour spaces ($CIELab$ or $CIELuv$), the RGB colour space is converted to XYZ colour space which is then converted to perceptual colour spaces. The component L is the same for both $CIELab$ and $CIELuv$ colour spaces. The conversion procedure is as follows:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.431 & 0.342 & 0.178 \\ 0.222 & 0.707 & 0.071 \\ 0.020 & 0.130 & 0.939 \end{bmatrix} \begin{bmatrix} \mathbf{x}_{\text{norm}}[n_1, n_2, 1] \\ \mathbf{x}_{\text{norm}}[n_1, n_2, 2] \\ \mathbf{x}_{\text{norm}}[n_1, n_2, 3] \end{bmatrix} \tag{7.5}$$

$$L = \begin{cases} 116 \times (\frac{Y}{Y_n})^{\frac{1}{3}} - 16 & \frac{Y}{Y_n} > 0.008856 \\ 903 \times (\frac{Y}{Y_n}) & \frac{Y}{Y_n} \leq 0.008856 \end{cases} \tag{7.6}$$

$$a = 500 \times (f(\frac{X}{X_n}) - f(\frac{Y}{Y_n})) \tag{7.7}$$

$$b = 200 \times (f(\frac{Y}{Y_n}) - f(\frac{Z}{Z_n})) \tag{7.8}$$

where X_n, Y_n and Z_n are reference white tristimulus value which are defined in CIE chromaticity diagram [19] and

$$f(t) = \begin{cases} t^{\frac{1}{3}} & t > 0.008856 \\ 7.787 \times t + \frac{16}{116} & t \leq 0.008856 \end{cases} \tag{7.9}$$

For u and v colour components, the conversion is defined by:

$$u = 13 \times L \times (u' - u'_n) \quad v = 13 \times L \times (v' - v'_n) \tag{7.10}$$

The equations for u' and v' are given below:

$$u' = \frac{4X}{X + 15Y + 3Z} \tag{7.11}$$

$$v' = \frac{9Y}{X + 15Y + 3Z} \tag{7.12}$$

The quantities u'_n and v'_n are the (u', v') chromaticity coordinates of a specified white object which defined by:

$$u'_n = \frac{4X_n}{X_n + 15Y_n + 3Z_n} \tag{7.13}$$

$$v'_n = \frac{9Y_n}{X_n + 15Y_n + 3Z_n} \tag{7.14}$$

7.4 Tensor-based Colour Image Methodology

7.4.1 Tensor Colour Images

Each colour image can be represented as a three-dimensional (3-D, i.e., horizontal, vertical and colour) data array. There is a technical challenge to proceed with applying a 2-D filtering process to a 3-D matrix which represents the colour image. It can either process a single channel of the colour image (e.g., luminance image) or perform the filtering operation on each colour channel individually. The latter approach is to employ the 2-D filters three times over three component images respectively. Instead of implementing the filter for each component of the colour image, a tensor of the colour image is generated and the filtering operation is directly applied to this tensor [6, 65, 66]. A tensor is a higher-order generalization of a vector (first order tensor) and a matrix (second order tensor). Tensors are multi-linear mappings over a set of vector spaces. A colour image represented by $\mathcal{T}$ is a tensor of order 3 and $\mathcal{T} \in \mathbb{R}^{\prod_1^3 N_n}$ where N_1 is the height of the image, N_2 is the width of the image and N_3 represents the number of colour channels. In this study, N_1 and N_2 vary from 16 to 64 and N_3 is 3. Tensor can be unfolded to n-mode mathematical objects. In this study, there are three modes for tensor $\mathcal{T}^{(\mathrm{n-mode})}$ which defined by:

$$\mathcal{T} \in \mathbb{R}^{\prod_1^3 N_n} \rightarrow \mathcal{T}^{(1)} \in \mathbb{R}^{N_1 \times (N_2 \times N_3)} \tag{7.15}$$

$$\mathcal{T} \in \mathbb{R}^{\prod_1^3 N_n} \rightarrow \mathcal{T}^{(2)} \in \mathbb{R}^{N_2 \times (N_1 \times N_3)} \tag{7.16}$$

$$\mathcal{T} \in \mathbb{R}^{\prod_1^3 N_n} \rightarrow \mathcal{T}^{(3)} \in \mathbb{R}^{N_3 \times (N_1 \times N_2)} \tag{7.17}$$

The 3-D colour image is unfolded to obtain 2-D tensors based on multi-linear analysis criteria [65], which are suitable for 2-D feature extraction filters. These tensors are used for feature extraction and classification. All modes are tested and the best one is unfolding mode 1 (Eq. (7.16)). In this study, the image $\mathbf{x}_{N_1 \times N_2 \times N_3}$ is unfolded to $\mathbf{x}_{N_1 \times N_2 N_3}$ which is called horizontal unfolding. Figure 7.2 shows the unfolding of colour images in different colour spaces.

7.4.2 PCA Statistical Analysis

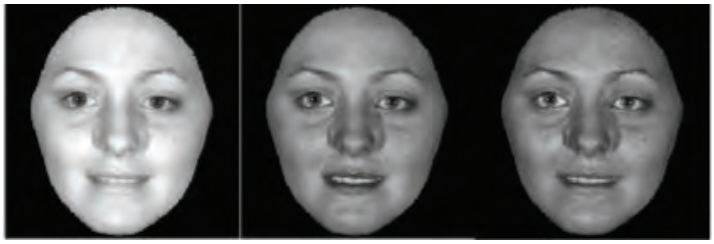
To demonstrate the role of colour information in expression recognition, the colour effect on the recognition performance is evaluated based on Fisher criterion [40, 58] which defined by:

$$J = \frac{trace(\mathbf{S_b})}{trace(\mathbf{S_w})} \tag{7.18}$$

where J represents a well-discriminative capability of the feature subspace for classification tasks, $\mathbf{S_b}$ is the between-class scatter matrix, $\mathbf{S_w}$ is within-class scatter matrix [44] and $trace(\mathbf{S_b})$, $trace(\mathbf{S_w})$ are defined to be the sum of the elements on the main diagonal of $\mathbf{S_w}$ and $\mathbf{S_b}$, respectively. The $\mathbf{S_b}$ and $\mathbf{S_w}$ matrices for feature vector ($\mathbf{x}$) are given by:

$$\mathbf{S_b} = \sum_{i=1}^{N_c} m_i (\mathbf{x}_{\mu_i} - \mathbf{x}_{\mu})(\mathbf{x}_{\mu_i} - \mathbf{x}_{\mu})^T \tag{7.19}$$

$$\mathbf{S_w} = \sum_{i=1}^{N_c} \sum_{\mathbf{x} \in c_i} (\mathbf{x} - \mathbf{x}_{\mu_i})(\mathbf{x} - \mathbf{x}_{\mu_i})^T \tag{7.20}$$



(a) RGB

(b) YCbCr

(c) CIELab

(d) CIELuv

Figure 7.2: Horizontal unfolding of facial expression image.

where N_c is the number of classes (i.e., for six expressions, $N_c = 6$), m_i is the number of training samples for each class, c_i is the class label, $\mathbf{x}_{\mu_i}$ is the mean vector for each class samples (m_i) and $\mathbf{x}_\mu$ is total mean vector over all training samples (m) defined by:

$$\mathbf{x}_{\mu_i} = \frac{1}{m_i} \sum_{\mathbf{x} \in c_i} \mathbf{x} \tag{7.21}$$

$$\mathbf{x}_{\mu} = \frac{1}{m} \sum_{i=1}^{N_c} m_i \mathbf{x}_{\mu_i} \tag{7.22}$$

The outputs of Eq. (7.19) and Eq. (7.20) can be displayed respectively as $k \times k$ matrices

$$\mathbf{S_w} = \begin{bmatrix} s_{w11} & s_{w12} & \cdots & s_{w1k} \\ s_{w21} & s_{w22} & \cdots & s_{w2k} \\ . & . & \cdots & . \\ s_{wk1} & s_{wk2} & \cdots & s_{wkk} \end{bmatrix} \tag{7.23}$$

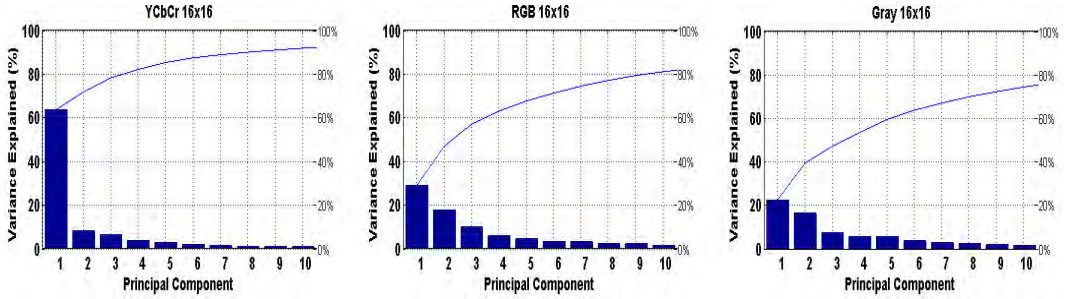
$$\mathbf{S_b} = \begin{bmatrix} s_{b11} & s_{b12} & \cdots & s_{b1k} \\ s_{b21} & s_{b22} & \cdots & s_{b2k} \\ . & . & \cdots & . \\ s_{bk1} & s_{bk2} & \cdots & s_{bkk} \end{bmatrix} \tag{7.24}$$

The trace of these matrices are given by:

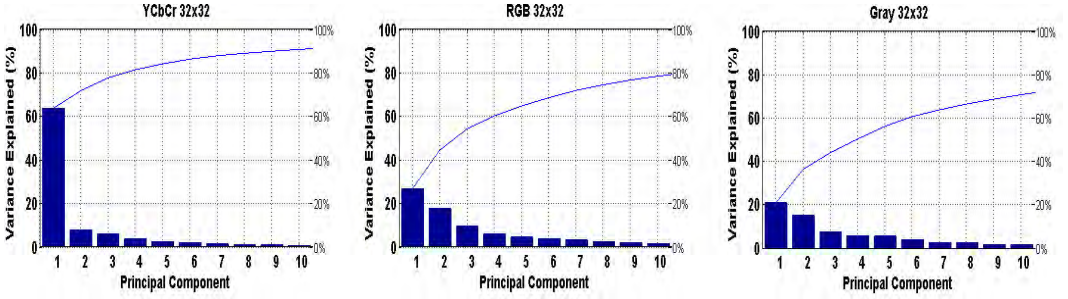
$$trace(\mathbf{S_w}) = \sum_{j=1}^{k} s_{wjj}, \quad trace(\mathbf{S_b}) = \sum_{j=1}^{k} s_{bjj}, \tag{7.25}$$

where s_{wjj} is the total summation of variance of sample variables in all classes and s_{bjj} is the total mean summation of variance of sample variables in all classes. Since a colour image has three different components whereas a gray-scale image has only one component, it can easily be proven that the output of filtered colour image has more power than gray-scale image. In other words, the variance of the features can explain the total variability in the standardized ratings, which might be a reasonable way to reduce the dimensions of features using this criterion. Based on the principal component analysis, the variance represented by the corresponding principal component is used to show that the total variance of the feature space is changed for different colour space images. The Pareto chart of the first ten PC's of the training set for 16×16, 32×32 and 64×64 resolution images are shown in Figure 7.3 for different colour spaces. As a result, the total variances are increased when the colour components of the image are considered. The graph also shows that the distribution of the

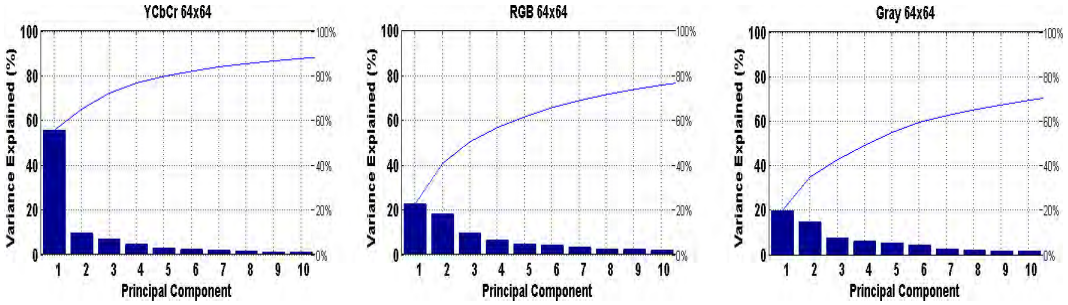
feature space can be expected as high as 90% in YC_bCr space compared with 75% when only luminance (gray-level) is considered.



(a) 16x16



(b) 32x32



(c) 64x64

Figure 7.3: Corresponding principal component variance in different colour spaces.

7.4.3 Facial Expression Images Under Illumination

The CIE (International Commission on Illumination) tristimulus system proposed alternative colour spaces in terms of its three coordinates relating usually a standard illuminance to a reference colour [129]. The CIELab is one of colorimetric colour spaces which separates a luminance variable 'L' from two perceptually uniform chromaticity variables ('a','b') [129]. The CIELab

is widely used in several image processing applications include: perceptual image quality assessment, face detection, skin detection, image segmentation [126, 129]. Despite the many advantages of such colour space, it has rarely been used in pattern recognition. In this study, the performance of the FER system in both CIELab and CIELuv colour spaces are investigated.

A facial colour image $\mathbf{x}[n_1, n_2, n_3]$ under illumination using the Lambertian model [70] is defined by:

$$\mathbf{x}[n_1, n_2, n_3] = \mathbf{R}[n_1, n_2, n_3] \times \mathbf{Iu}[n_1, n_2, n_3] \tag{7.26}$$

where $\mathbf{Iu}[n_1, n_2, n_3]$ and $\mathbf{R}[n_1, n_2, n_3]$ are respectively the illumination and the reflectance at a location $[n_1, n_2]$ in n_3 colour channel. For robust pattern recognition under various illumination conditions, the $\mathbf{R}[n_1, n_2, n_3]$ is the key feature due to its stability. Since, the multiplication means nonlinear relation between reflection and illumination, it is a problem to extract key facial features by solving Eq. (7.26) directly. Hence, a common assumption is that $\mathbf{Iu}[n_1, n_2, n_3]$ changes slowly and $\mathbf{R}[n_1, n_2, n_3]$ varies abruptly, that is, $\mathbf{Iu}[n_1, n_2, n_3]$ is regarded as the low frequency of the image $\mathbf{x}[n_1, n_2, n_3]$ and $\mathbf{R}[n_1, n_2, n_3]$ is the high frequency which can be regarded as "noise" in a noisy image. In [70], the logarithm operator is used to make the equation linear and then apply the low pass filter (LPF) or high pass filter (HPF) to separate the illumination from the image. Usually the homomorphic filter is used to separate the reflectance and illumination [19]. However, the threshold for filter is not constant due to variation of illumination. In addition, this filter is nonlinear filter and makes the computational complex of the system high. Therefore, it is computationally more costly to use this filter for FER system. An alternative is Retinex colour image enhancement algorithm, whose theory models the effect of varying intensity of light on colour perception of human vision [136]. Retinex theory addresses the colour invariance and the illumination-reflectance model. However, it has two weaknesses [136, 137], First, the images enhanced by Retinex algorithm often have "halos" and their details may be also blurred.

Second, restoration functions based on Retinex theory do not have rigourous mathematical or conclusive neurophysiological proof. The facial expression relies on the details of the facial image and it is essential for any FER system to have the full details of face images. Therefore, Retinex is not suitable to enhance the illumination from of the colour facial image. In this study, the perceptual colour spaces are investigated. The RGB colour images has been transformed to both $CIELab$ and $CIELuv$ colour spaces. The local normalization technique [138] is performed to reduce the effect of illumination for facial expression images.

To see the effect of illumination on images in different colour spaces, the illumination pattern is applied to original colour facial images [90]. Let $\mathbf{y}[n_1, n_2]$ denotes the illumination pattern, the image under illumination ($\hat{\mathbf{x}}[n_1, n_2, n_3]$) is given by

$$\hat{\mathbf{x}}[n_1, n_2, n_3] = \mathbf{x}[n_1, n_2, n_3]\mathbf{y}[n_1, n_2] \tag{7.27}$$

and

$$\mathbf{y} = \begin{bmatrix} T & 2T & \cdots & N_2T \\ T & 2T & \cdots & N_2T \\ . & . & . & . \\ . & . & . & . \\ . & . & . & . \\ T & 2T & \cdots & N_2T \end{bmatrix} \tag{7.28}$$

and

$$\mathbf{y}[n_1, n_2] = \begin{cases} \mathbf{y}[n_1, n_2] & \mathbf{y}[n_1, n_2] < 1 \\ 1 & \mathbf{y}[n_1, n_2] > 1 \end{cases} \tag{7.29}$$

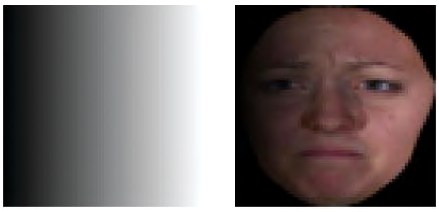
where $0 < T < 0.1$ is the intensity value which used to change the effect of illumination on the image and $\mathbf{y}[n_1, n_2]$ is the matrix element of $\mathbf{y}$. Figure 7.4 shows the illumination pattern and its impact on the images. One hundred

facial images are randomly chosen and transformed to different colour spaces (YC_bC_r, $CIELab$ and $CIELuv$). The structural similarity (SSIM) index [122] is adopted to evaluate the quality of the image under illumination variation. Table 7.1 presents the SSIM along with their illumination parameter values.

Table 7.1: Illumination value for different SSIM index.

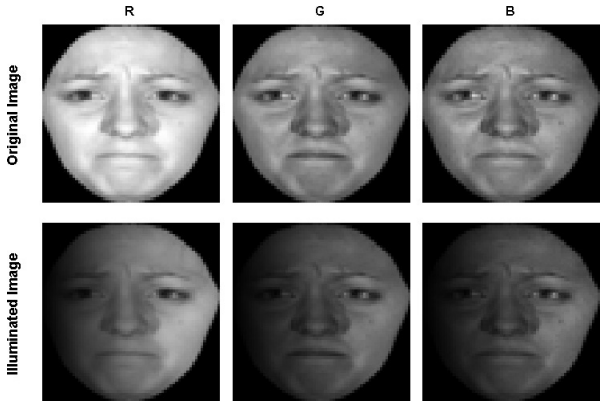
Illumination value (T)	SSIM Index
0.004-0.01	0.25-0.35
0.01-0.03	0.75-0.85
0.04-0.08	0.90-0.93
0.09-0.1	0.93-0.96

The range of T and thats of SSIM depend on the resolution of the image. Figure‘ 7.5 shows both original and images under lighting changes for different colour components. It is shown that the perceptual $CIELab$ and $CIELuv$ are more robust against illumination variation than other colour spaces in terms of the structure similarity criterion.

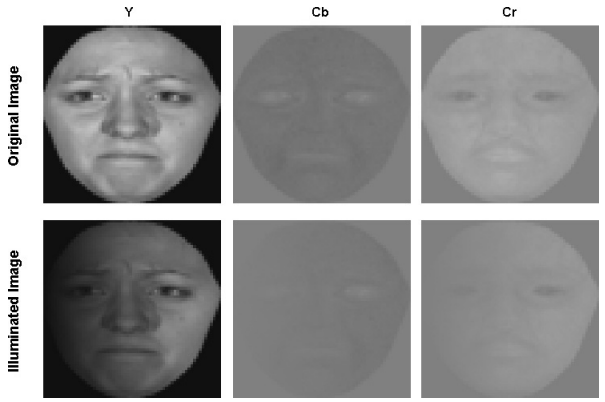


(a) (b)

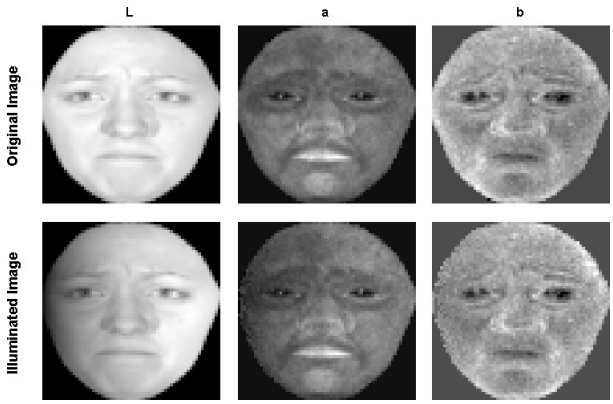
Figure 7.4: (a) Illumination pattern, (b) Image under illumination of (a).



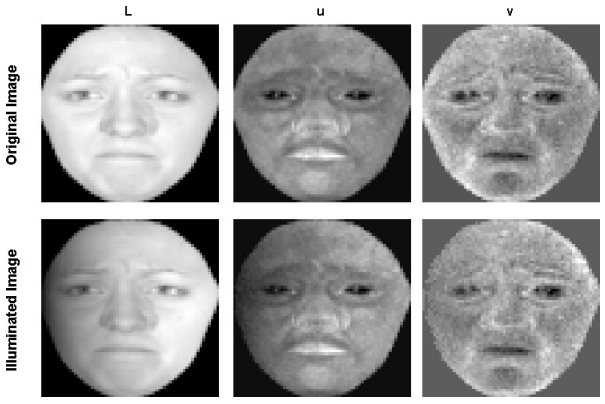
(a)



(b)



(c)



(d)

Figure 7.5: Facial expression images in different colour components Top row: original image, Bottom row: image under illumination variation.

7.5 Experimental Results

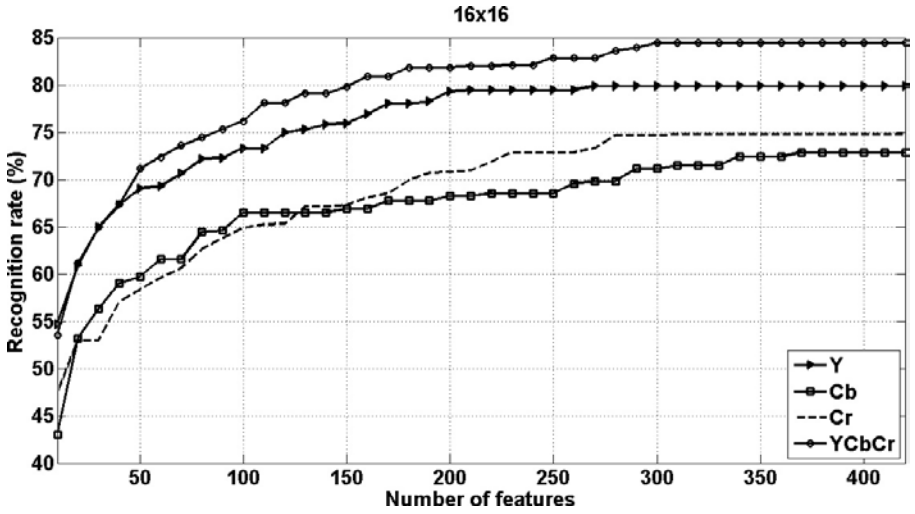
The performance of the TPCF in perceptual colour spaces is evaluated using frontal view facial images from Binghamton University 3-D Facial Expression Database (BU-3DFE) [62]. The database presently contains 100 subjects (56% female, 44% male), with age ranging from 18 years to 70 years old and a variety of ethnic/racial ancestries, including White, Black, East-Asian, Middle-east Asian, Indian and Hispanic Latino. The database has images which are not relevant to six main facial expressions. Therefore, these images are ignored and not used in testing and training set. Totally, 2400 facial expression images are selected from the database. The original resolution of the images is 512×512. However, the resolution of the images is normalized to different resolutions for several experiments. Fifty percent of the images are used for the training set and the rest for the testing set. The RGB facial images are transformed to other colour spaces (YC_bC_r, $CIELab$ and $CIELuv$). All the images are normalized using Eq. (7.1) and unfolded in horizontal mode (Eq. (7.15)). A bank of 24 Log-Gabor filters is used to extract the features from unfolded tensors representing colour facial images which are concatenated and employed for the MIQ algorithm to generate the feature vector. These features are classified by a multi-class LDA classifier. The results are shown in Figures 7.6-7.8 using images of 16×16, 32×32 and 64×64 resolutions. It can be seen from the results that the average recognition rate with standard deviation, $0 \leq \sigma < 2.5$, for red (R) and green (G) in RGB, luminance or Gray (Y) in YC_bC_r, lighting (L) in $CIELab$ and $CIELuv$ are the best among other colour channels in all resolution images. The average recognition rate is improved as well when the resolution is increased from 16×16 to 64×64. Furthermore, the accuracy of the FER system as shown in Figures 7.9-7.11 is significantly improved when tensor colour images are adopted to the system for all different resolutions and expressions.

Since, colour images are more sensitive to illumination than gray-scale im-

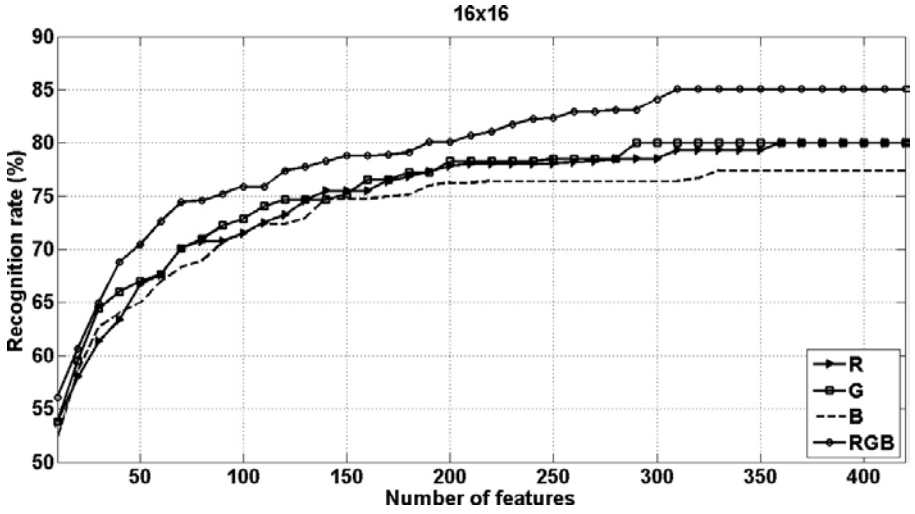
Table 7.2: Comparison for different resolutions and colour spaces.

Size	SSIM	Gray	RGB	YC_bC_r	$CIELab$	$CIELuv$
16x16	0.25	35.04	41.54	42.57	65.65	72.26
	0.78	67.86	72.26	71.04	79.21	81.76
	0.90	74.21	76.89	75.91	80.72	83.23
	0.93	77.04	79.48	80.65	83.17	84.27
	1	80.95	84.11	84.91	84.61	84.93
32x32	0.30	54.37	63.46	64.62	79.71	79.81
	0.81	79.17	82.87	82.92	83.45	84.61
	0.92	80.59	83.47	83.62	84.47	85.54
	0.94	81.68	85.23	85.31	85.13	86.52
	1	83.66	86.38	86.70	85.28	86.68
64x64	0.35	45.83	46.21	47.39	68.7	73.31
	0.85	72.35	73.91	75.81	80.61	82.43
	0.93	79.47	80.43	82.17	83.32	84.51
	0.96	81.43	81.78	83.49	85.71	86.68
	1	86.09	86.97	86.98	86.52	87.05

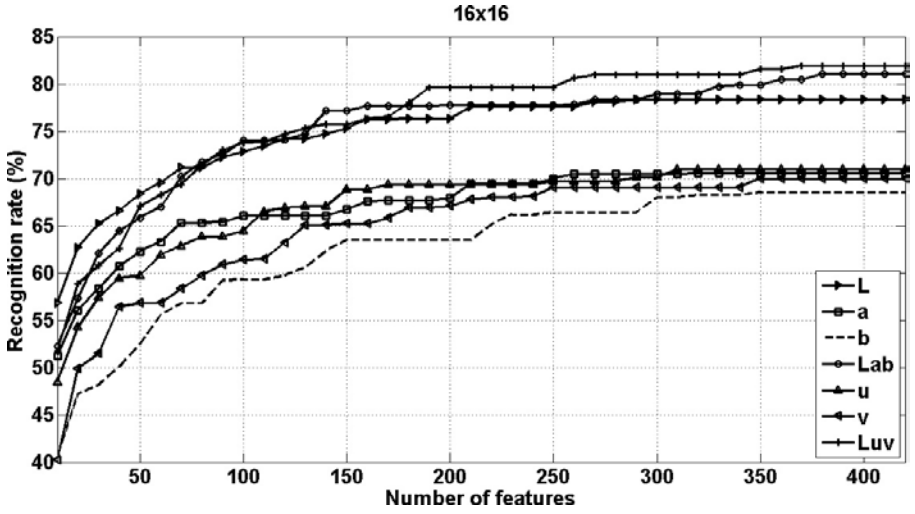
ages, the testing set under slight illumination variation is used to evaluate the robustness of the system performance in term of recognition rate and error rate. The illumination of testing set images are varied in RGB colour space. Then these images are used as input to the FER system. The SSIM index is adopted to assess the quality of the image under slight illumination variation. Table 7.2 summarizes the average recognition rates under different illumination and resolutions. It can be easily seen from Table 7.2 that the perceptual colour space such as $CIELuv$ is more robust than other colour spaces under slight changes of illumination for facial images of different resolution. Furthermore, it has a comparable or slightly superior performance to that of others in terms of recognition rate for images without any illumination variations (i.e., $SSIM = 1$ in Table 7.2 in all resolutions.



(a)

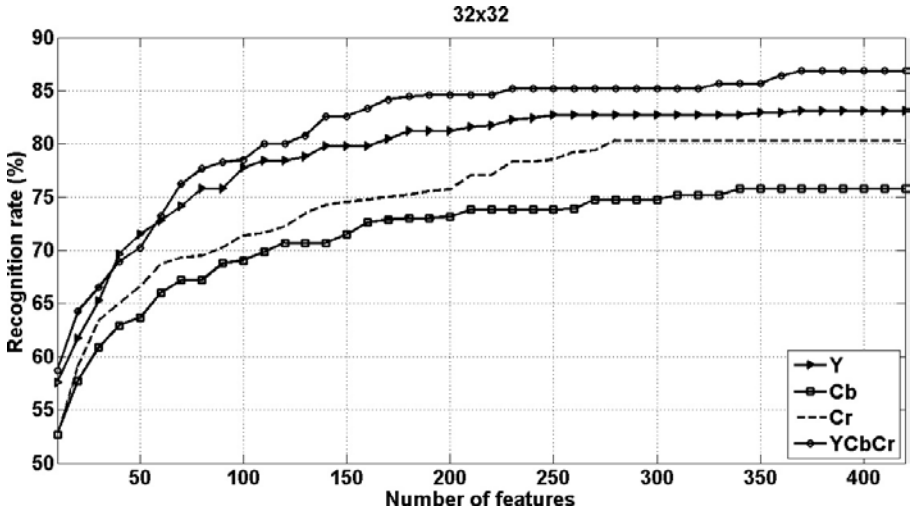


(b)

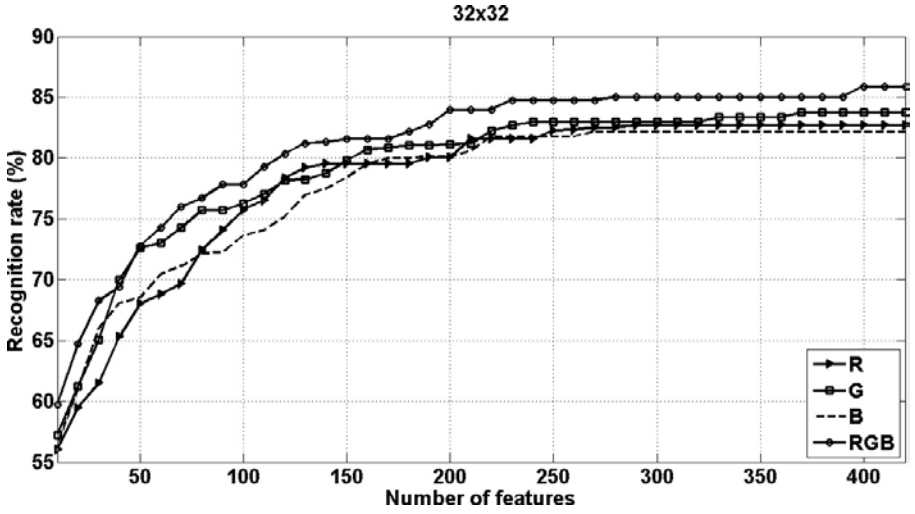


(c)

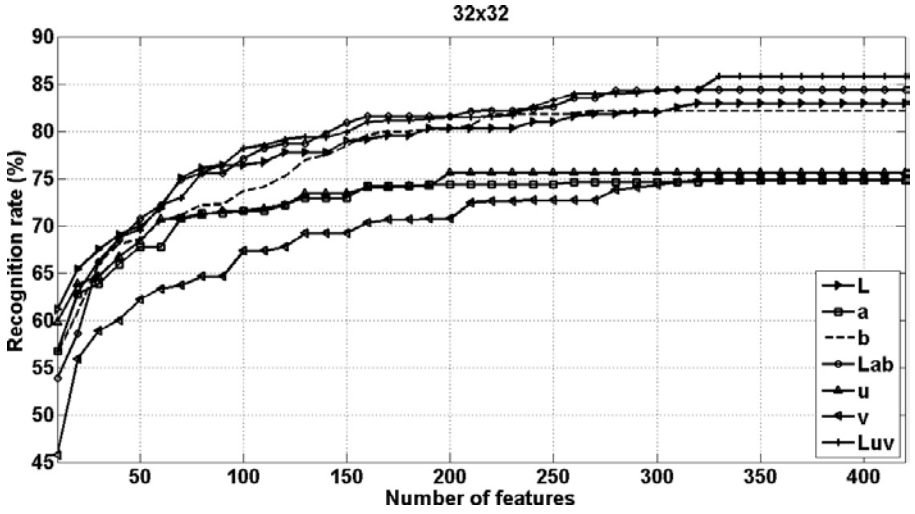
Figure 7.6: Comparative evaluation of performance in different colour spaces from 16x16 images (a) YC_bC_r, (b) RGB, (c) CIELab and CIELuv



(a)

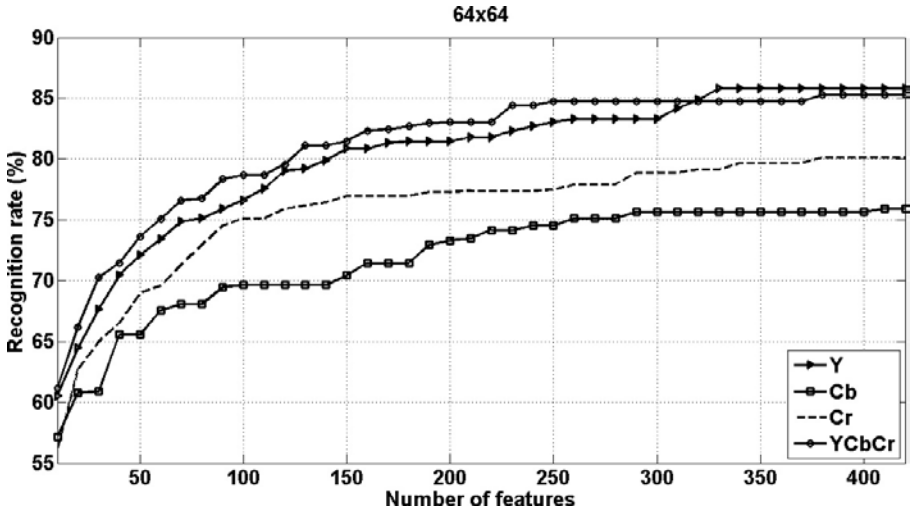


(b)

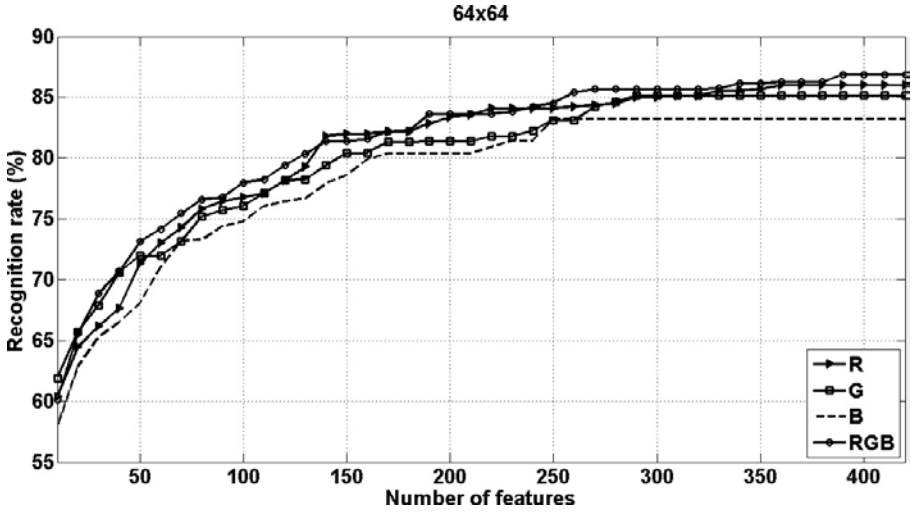


(c)

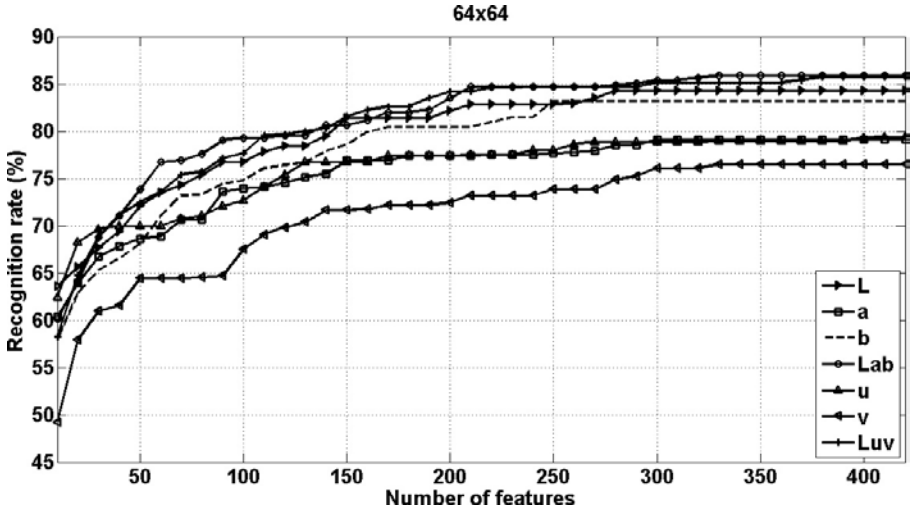
Figure 7.7: Comparative evaluation of performance in different colour spaces from 32x32 images (a) YC_bC_r, (b) RGB, (c) CIELab and CIELuv



(a)



(b)



(c)

Figure 7.8: Comparative evaluation of performance in different colour spaces from 64x64 images (a) YC_bC_r, (b) RGB, (c) CIELab and CIELuv

7.6 Discussion

To investigate the performance of FER system for each expression and ascertain the performance of classifier in terms of recognition rate and error rate, the receiver (relative) operation characteristic (ROC) curve is plotted for both original and images under illumination variations. The ROC curve represents the fraction of true positive out of the positives, which is called true positive (TP) rate, versus the fraction of false positive out of the negatives, which is called false positive (FP) rate [44]. Figures 7.9-7.11 show, respectively, the ROC curves of each of six expressions for different resolutions. They illustrate that the two expressions. i.e., happy and surprise, have the minimum error and the maximum recognition rates compared with those of the other expressions. In addition, they are more robust under slight illumination changes. With regard to colour space, the $CIELuv$ has the maximum recognition rate for each expression with different resolutions, whereas, $Gray$ is very sensitive to illumination variation and it has the worst recognition rate for images under illumination variations.

For comparison, the proposed tensor perceptual colour framework is benchmarked against a number of well known state-of-the-art techniques for FER system using the same database with results presented in Table 7.3 which demonstrates the accuracy of the FER system in terms of the best recognition rate. In Gabor methodology, 18 Gabor filters were applied on 34 fiducial points and the amplitude of the filter's outputs were used to create a feature vector [71]. In addition, the implementation of 40 Gabor filters is used for feature extraction as reported in [139]. For context methodology, the feature statistics were represented as the feature vector of facial expression image [71]. It can be easily seen that the TPCF outperformed other popular methods in terms of average recognition rate.

From computational complexity point of view, the memory usage is increased since the tensor is made by concatenating all the colour information.

Table 7.3: Comparison between TPCF and the state-of-the-art methods

Methodology	Recognition Rate (%)
Context method [71]	79.2
Gabor method [71]	74.1
40 Gabor filters [139]	73.8
$CIELuv$ TPCF	87.05

As well, there is an increase in computation complexity associated with the colour space conversion. The number of operations required for colour space conversions is examined. For instance, from RGB to gray-scale conversion, each pixel requires 3 multiplications and 2 additions. With $3N_1N_2$ pixels in a full RGB colour image, the total number of operations required for forward colour conversion is $9N_1N_2$ multiplications and $6N_1N_2$ additions. Table 7.4 demonstrates the number of operations for all colour space conversions.

Table 7.4: Number of operations required for colour space conversions

Mode	Multiplications	additions
Gray	$9N_1N_2$	$6N_1N_2$
YC_bC_r	$9N_1N_2$	$8N_1N_2$
$CIELab$	$36N_1N_2$	$18N_1N_2$
$CIELuv$	$42N_1N_2$	$21N_1N_2$

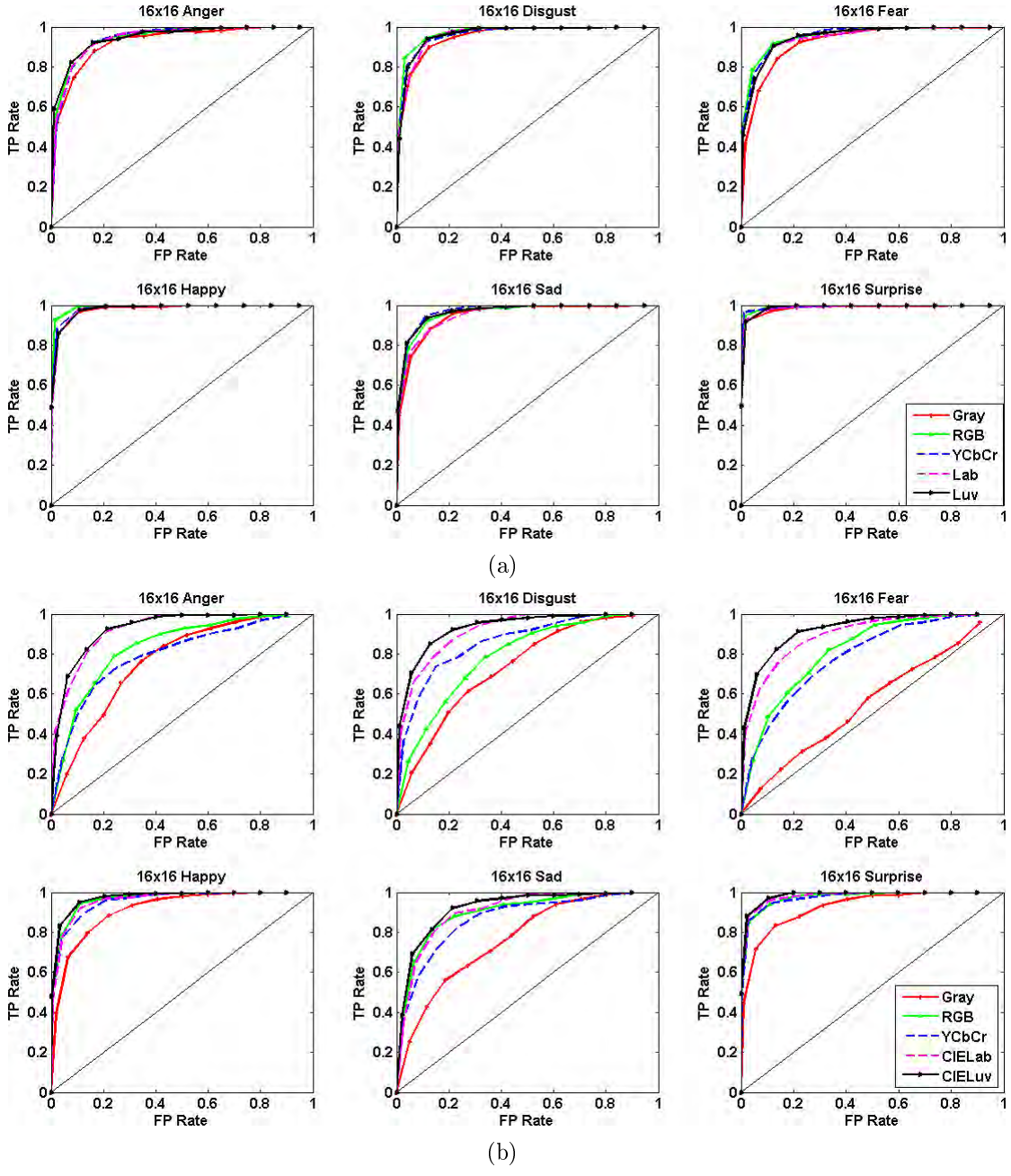
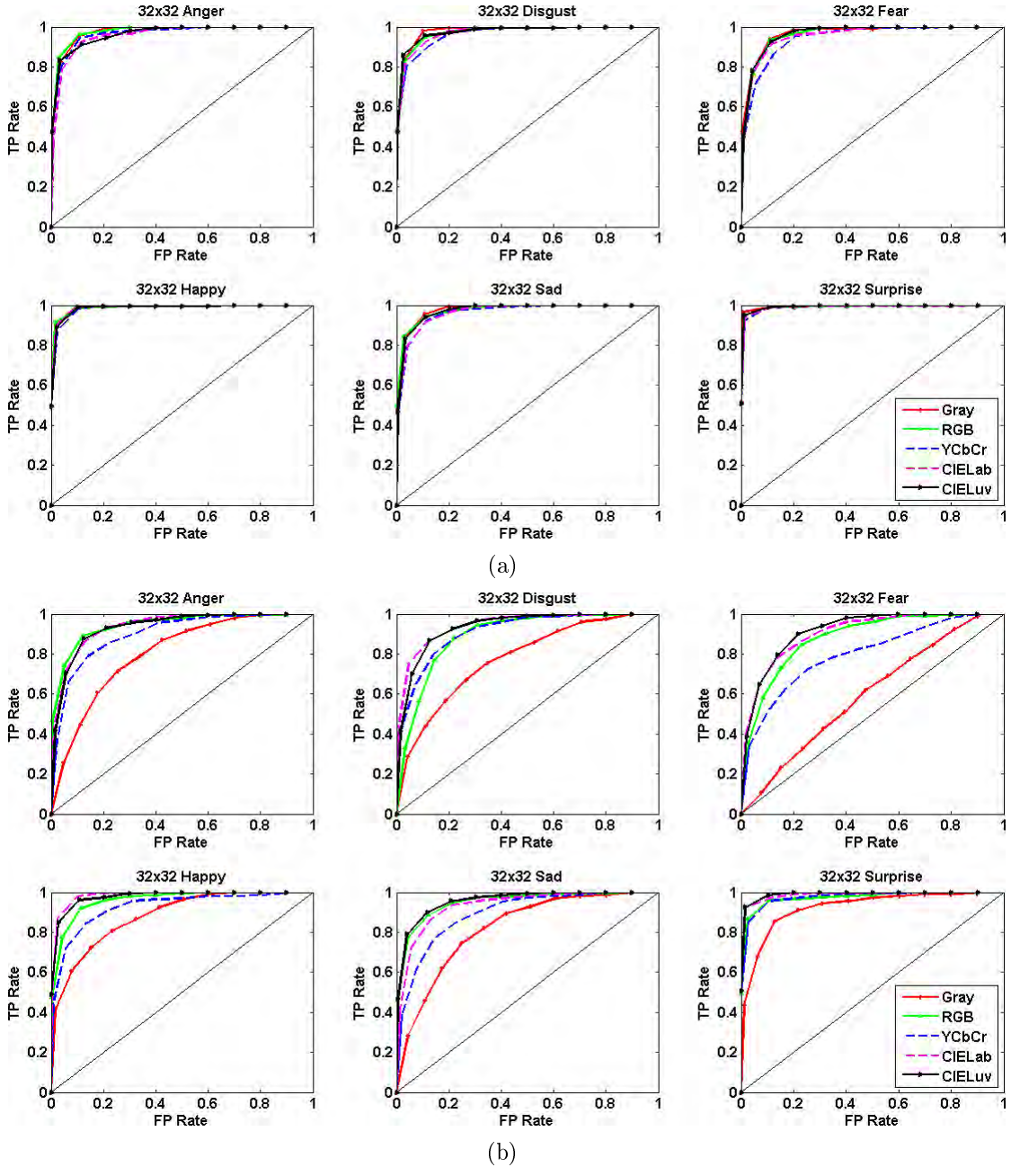


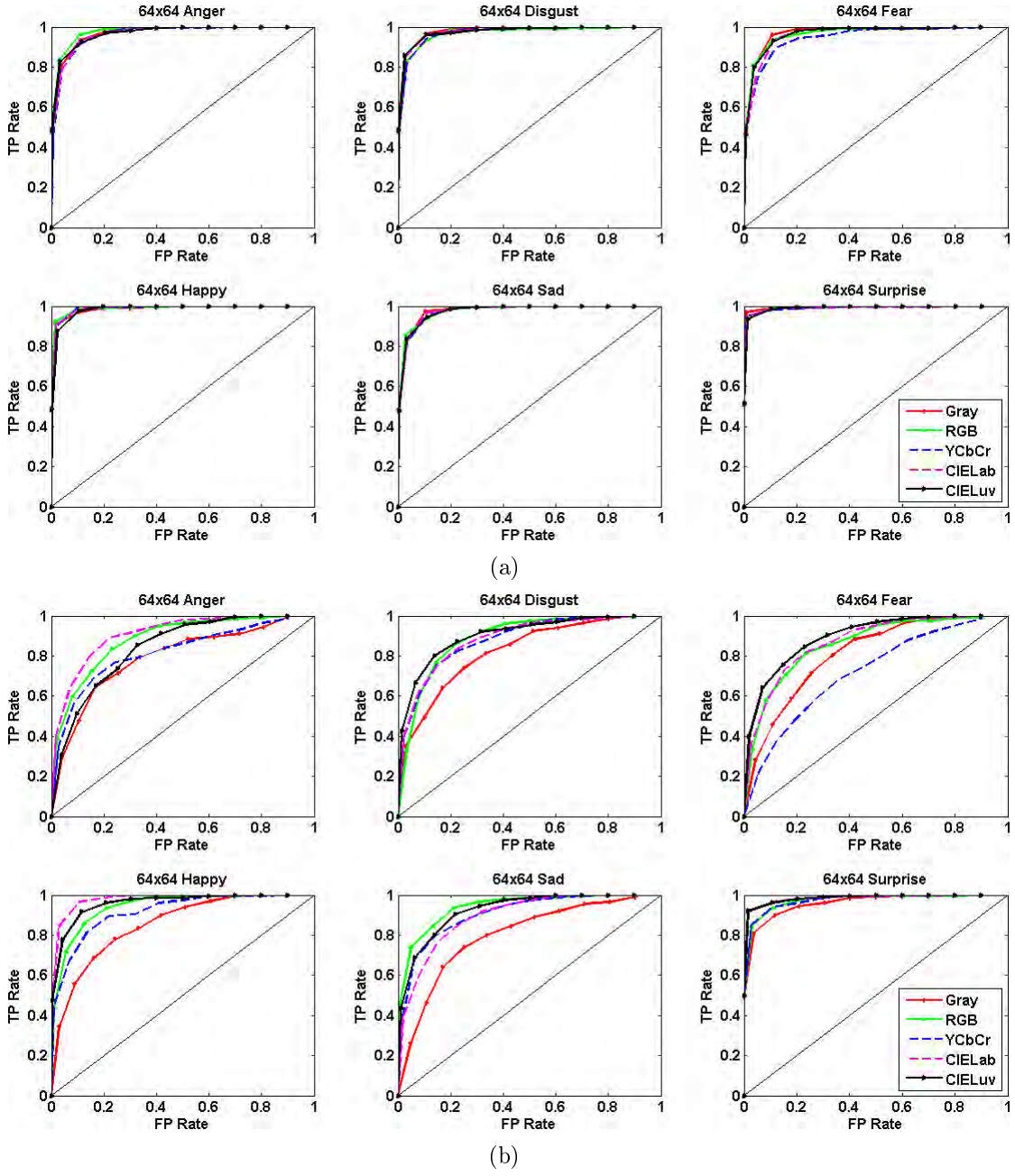
Figure 7.9: ROC curves of different tensors for different expressions from 16×16 images (a) original image with no illumination change, (b) image under illumination variation (SSIM Index=0.25).



(a)

(b)

Figure 7.10: ROC curves of different tensors for different expressions from 32×32 images (a) original image with no illumination change, (b) image under illumination variation (SSIM Index=0.36).



(a)

(b)

Figure 7.11: ROC curves of different tensors for different expressions from 64×64 images(a) original image with no illumination change, (b) image under illumination variation (SSIM Index=0.60).

7.7 Summary

A novel tensor perceptual colour framework for facial expression recognition system in perceptual colour space is introduced in this chapter. Using this

approach, the RGB colour images are first transformed to perceptual colour spaces after which the horizontal unfolded tensor is adopted to generate the 2-D tensor for feature extraction. The 2-D tensor is normalized before the features are extracted using a bank of 24 Log-Gabor filters and the optimum feature are selected based on the MIQ algorithm. The images under slight illumination variation are used to test robustness of the FER system performance. Experimental results show that the perceptual colour space ($CIELuv$) has significantly improved the performance of the system in terms of recognition rate for all expression under varying illumination situations. In addition, the performance of the TPCF has marked advantages in facial expression recognition for low-resolution facial images in terms of recognition rate and error rate. To the best of author's knowledge, the achieved recognition accuracy of the TPCF for FER system is better than any other reported in the literature so far using the BU-3DFE database.

Chapter 8

Conclusions

8.1 Primary Finding

This dissertation presents an investigation and implementation of a facial expression recognition framework which improves the performance of facial expression recognition and reduces the computational complexity. Several issues that are relevant to design of FER systems have been addressed. Chapter 2 describes the main system modules including image pre-processing, feature extraction, feature selection and classification and investigates the fundamental state-of-the-art techniques for FER systems.

Chapter 3 presents two novel methods for image pre-processing module. The first is a novel facial detection technique for sequence of images [11]. By adopting this technique, the FER system only processes facial images with certain expression instead of processing sequence of images. As a result, the performance of the system in terms of computational complexity has been improved and the processing time has been reduced for pre-processing module. Secondly, the colour facial image pre-processing is presented for FER system. Using this approach, the colour images are transformed to 2-D tensors based on multi-linear analysis [6]. The results have shown that the tensors improved the performance of FER system in terms of recognition rate using colour information.

Chapter 4 discusses several feature extraction methods. The Zernike mo-

ments have been adopted and investigated the performance of FER system in presence of the image noise and image rotation [7]. Overall, the ZM are robust in terms of recognition rate for images with noise and orientation. The HLACLF method [9], the LGFCT method [103] and the HFR method [10] have been proposed and investigated. The HLACLF method is based on autocorrelation and it has the best performance for FER system. Using the LGFCT method, the features are extracted by both Log-Gabor filters and Contourlet transform and, then, these features are combined together. The results have shown that the LGFCT method has improved the performance of the FER system compared with the CT or the LGF method. However, its computational complexity is higher than other feature extraction methods. For the HFR method, the features are extracted from face regions (eyes and mouth) and the results have demonstrated that the eyes give more information for expression recognition than other parts of the face. Furthermore, the combination of features in different parts of the face has improved the performance of FER system.

In Chapter 5, several feature selection techniques have been investigated. Feature selection has employed to determine the most discriminative features and reduced the complexity of features for classification. Both filter and wrapper approaches have been investigated. In the filter approach, the MI and MIQ techniques have adopted to find the most discriminative features for expression recognition. In wrapper approach, the GA has employed for feature selection. the results have shown that the MIQ method performance has outperformed the others in terms of recognition rate.

Chapter 6 discusses several classification methods including NB and LDA. The GFSSIM has been proposed and employed on facial expression images with different noises. Experimental results have shown that the GFSSIM has better performance than other methods for expression recognition in presence of noise.

Chapter 7 investigates the effect of colour information on expression recog-

nition. The colour images have been converted to 2-D tensors and then the features have been extracted from 2-D tensors using Log-Gabor filters. In addition, the perceptual colour spaces have been investigated and a novel tensor perceptual colour framework is proposed for FER system. Overall, the TPCF outperforms the methods using gray-scale facial images. Furthermore, it has the best recognition rate compared with other colour spaces.

8.2 Future Work

In this thesis, the FER system is designed only for constrained condition. For it to be used in practice, several issues still need to be solved. Most of face image processing applications are video based. Therefore, both algorithm and code need to be optimized for employing in real time analysis. Sometimes, its inevitable to trade between accuracy and speed. Although investigations and experimental results presented in this thesis are with respect to an FER system, the TPCF itself is designed to be a general method for colour images. It can be used in modelling, recognition and therefore, it can be employed for other pattern recognition applications, i.e., face recognition, colour object recognition. The other problem is the limitation of the databases for facial expressions. The number of subjects are limited to 100 and they are well-defined images without any occlusions (i.e., sunglasses, long beard). Robustness of the FER system against occlusion and other non-ideal conditions requires further research and investigation.

Appendices

Appendix A

Eigenface

Appendix A: Eigenface

Based on a statistical technique known as Principal Component Analysis (PCA), the number of eigenvectors can be reduced for the covariance matrix from N (the number of pixels in a given image) to N_s (the number of images in the dataset). In general, PCA is used to describe a large dimensional space with a relative small set of vectors. It is a popular technique for finding patterns in data of high dimension, and is used commonly in both face recognition and image compression. PCA is applicable to face recognition because face images usually are very similar to each other (relative to images of non-faces) and clearly share the same general pattern and structure. PCA indicates that since only N_s images are exist, it can only have N_s non-trivial eigenvectors. The covariance matrix ($\mathbf{c}$) for the training set is:

$$\mathbf{c} = \frac{1}{N_s}\sum_{i=1}^{N_s} \mathbf{x}_d^f[\bullet, i]\mathbf{x}_d^f[\bullet, i]^T = \frac{1}{N_s}\mathbf{A}\mathbf{A}^T \tag{A.1}$$

where $\mathbf{A} = \{\mathbf{x}_d^f[\bullet, 1], \mathbf{x}_d^f[\bullet, 2], \cdots, \mathbf{x}_d^f[\bullet, N_s]\}$ and

$$\mathbf{x}_d^f[\bullet, i] = \mathbf{x}^f[\bullet, i] - \mathbf{x}_\mu^f \tag{A.2}$$

and

$$\mathbf{x}_{\mu}^{f} = \frac{1}{N_s} \sum_{i=1}^{N_s} \mathbf{x}^{f}[\bullet, i] \tag{A.3}$$

where $\mathbf{x}_{\mu}^{f}$ is the average of training samples. The eigenvectors of $\mathbf{c}$ can be solved by taking the eigenvectors of a new $N_s \times N_s$ matrix:

$$\mathbf{c}' = \mathbf{A}^{\mathbf{T}}\mathbf{A} \tag{A.4}$$

Because of the following math trick:

$$\begin{aligned} \mathbf{A}^{\mathbf{T}}\mathbf{A}\nu_i &= u_i\nu_i \\ \mathbf{A}\mathbf{A}^{\mathbf{T}}\mathbf{A}\nu_i &= u_i\mathbf{A}\nu_i \end{aligned} \tag{A.5}$$

where ν_i is an eigenvector of $\mathbf{c}'$. From this simple proof, it can be seen that is an eigenvector of $\mathbf{c}$. The N_s eigenvectors of $\mathbf{c}'$ are finally used to form the N_s eigenvectors $\mathbf{u}_l$ of $\mathbf{c}$ that form the eigenface basis:

$$\mathbf{u}_l = \sum_{k=1}^{N_s} \nu_{lk}\mathbf{x}_{d}^{f}[\bullet, k] \tag{A.6}$$

Appendix B

The Naive Bayes probabilistic model

Appendix B: Proof of Equation (6.1)

Abstractly, the probability model for a classifier is a conditional model

$$p(c|x_1^f, \cdots, x_m^f) \tag{B.1}$$

over a dependent class variable c with a small number of outcomes or classes, conditional on several feature variables x_1^f through x_m^f. The problem is that if the number of features m is large or when a feature can take on a large number of values, then basing such a model on probability tables is infeasible. Therefore, the model is reformulated to make it more tractable. Using Bayes theorem, it can be defined by

$$p(c|x_1^f, \cdots, x_m^f) = \frac{p(c)\, p(x_1^f, \cdots, x_m^f|c)}{p(x_1^f, \cdots, x_m^f)} \tag{B.2}$$

In plain English the above equation can be written as:

$$\text{posterior} = \frac{\text{prior} \times \text{likelihood}}{\text{evidence}} \tag{B.3}$$

In practice, only the numerator of that fraction is important. Since the denominator does not depend on c and the values of the features x_m^f are given, Therefore, the denominator is effectively constant. The numerator is equiva-

lent to the joint probability model:

$$p(c, x_1^f, \cdots, x_m^f) \tag{B.4}$$

which can be rewritten as follows, using repeated applications of the definition of conditional probability:

$$\begin{aligned}
&p(c, x_1^f, \cdots, x_m^f) \\
&= p(c)\, p(x_1^f, \cdots, x_m^f | c) \\
&= p(c)\, p(x_1^f | c) p(x_2^f, \cdots, x_m^f | c, x_1^f) \\
&= p(c)\, p(x_1^f | c) p(x_2^f | c, x_1^f) p(x_3^f, \cdots, x_m^f | c, x_1^f, x_2^f) \\
&= p(c)\, p(x_1^f | c) p(x_2^f | c, x_1^f) p(x_3^f | c, x_1^f, x_2^f) p(x_4^f, \cdots, x_m^f | c, x_1^f, x_2^f, x_3^f) \\
&= p(c)\, p(x_1^f | c) p(x_2^f | c, x_1^f) p(x_3^f | c, x_1^f, x_2^f) \cdots p(x_m^f, \cdots, x_m^f | c, x_1^f, x_2^f, x_3^f, \cdots, x_{m-1}^f)
\end{aligned} \tag{B.5}$$

and so forth. Now the "naive" conditional independence assumptions come into play: assume that each feature x_i^f is conditionally independent of every other feature x_j^f for $j \neq i$. This means that:

$$p(x_i^f | c, x_j^f) = p(x_i^f | c) \tag{B.6}$$

and so the joint model can be expressed as

$$\begin{aligned}
&p(c, x_1^f, \cdots, x_m^f) = p(c) p(x_1^f, c) p(x_2^f, c) p(x_3^f, c), \cdots \\
&= p(c) \prod_{i=1}^{m} p(x_i^f | c)
\end{aligned} \tag{B.7}$$

This means that under the above independence assumptions, the conditional distribution over the class variable c can be expressed like this:

$$p(c|x_1^f, \cdots, x_m^f) = \frac{1}{s} p(c) \prod_{i=1}^{m} p(x_i^f|c) \tag{B.8}$$

where s is a scaling factor dependent only on $x_1^f, \ldots, x_m^f$, i.e., a constant if the values of the feature variables are known.

Appendix C

Linear Discriminant function

Appendix C: LDA Classifier

If there are g groups, the Bayes' rule is minimize the total error of classification by assigning the object to group conditional probability i which has the highest conditional probability where:

$$p(i|\mathbf{x}^f) > p(j|\mathbf{x}^f); \quad \text{for} \quad \forall \quad \text{j} \neq \text{i} \tag{C.1}$$

Since $p(i|\mathbf{x}^f)$ cannot be found (i.e., given the measurement, what is the probability of the class?) directly from the measurement and $p(\mathbf{x}^f|i)$ can be obtained (i.e., given the class, the measurement are used to compute the probability for each class), therefore, Bayes theorem is used:

$$p(i|\mathbf{x}^f) = \frac{p(\mathbf{x}^f|i).p(i)}{\sum_{i,j} p(\mathbf{x}^f|j).p(j)} \tag{C.2}$$

Thus, the Bayes' Rule becomes: Assign the object to group i if

$$\frac{p(\mathbf{x}^f|i).p(i)}{\sum_k p(\mathbf{x}^f|k).p(k)} > \frac{p(\mathbf{x}^f|j).p(j)}{\sum_k p(\mathbf{x}^f|k).p(k)}; \quad \text{for} \quad \forall \quad \text{j} \neq \text{i} \tag{C.3}$$

The denominators for both sides of inequality are positive and the same, therefore, they are cancel out to become

$$p(\mathbf{x}^f|i).p(i) > p(\mathbf{x}^f|j).p(j); \quad \text{for} \quad \forall \quad \text{j} \neq \text{i} \tag{C.4}$$

If many classes are exist and many dimension of measurement which each dimension will have many values, the computation of conditional probability $p(\mathbf{x}^f|i)$ requires a lot of data. It is more practical to assume that the data come from some theoretical distribution. The most widely used assumption is that the data come from multivariate normal distribution which is given by

$$p(\mathbf{x}^f|i) = \frac{1}{(2\pi)^{\frac{n}{2}}|\mathbf{c}_i|^{\frac{1}{2}}} \exp(-\frac{1}{2}(\mathbf{x}^f - \mu_i)^T\mathbf{c}_i^{-1}(\mathbf{x}^f - \mu_i)) \tag{C.5}$$

where, μ_i is vector mean and $\mathbf{c}_i$ is covariance matrix of group i. Inputting the distribution formula into Bayes rule, the following Equations (Since factor of $(2\pi)^{\frac{n}{2}}$ are equal for both sides, they cancel out) assign object with $\mathbf{x}^f$ to group i if:

$$\frac{p(i)}{|\mathbf{c}_i|^{\frac{1}{2}}} \exp(-\frac{1}{2}(\mathbf{x}^f - \mu_i)^T\mathbf{c}_i^{-1}(\mathbf{x}^f - \mu_i)) > \frac{p(j)}{|\mathbf{c}_j|^{\frac{1}{2}}} \exp(-\frac{1}{2}(\mathbf{x}^f - \mu_j)^T\mathbf{c}_j^{-1}(\mathbf{x}^f - \mu_j)); \quad j \neq i \tag{C.6}$$

take logarithmic of both sides

$$-\frac{1}{2}\ln(|\mathbf{c}_i|)+\ln(p(i))-\frac{1}{2}(\mathbf{x}^f-\mu_i)^T\mathbf{c}_i^{-1}(\mathbf{x}^f-\mu_i) > -\frac{1}{2}\ln(|\mathbf{c}_j|)+\ln(p(j))-\frac{1}{2}(\mathbf{x}^f-\mu_j)^T\mathbf{c}_j^{-1}(\mathbf{x}^f-\mu_j) \tag{C.7}$$

multiply both sides with -2, the sign of inequality needs to be changed

$$\ln(|\mathbf{c}_i|)-2\ln(p(i))+(\mathbf{x}^f-\mu_i)^T\mathbf{c}_i^{-1}(\mathbf{x}^f-\mu_i) < \ln(|\mathbf{c}_j|)-2\ln(p(j))+(\mathbf{x}^f-\mu_j)^T\mathbf{c}_j^{-1}(\mathbf{x}^f-\mu_j) \tag{C.8}$$

if all covariance matrices are equal $\mathbf{c} = \mathbf{c}_i = \mathbf{c}_j$, then it can simplify further into

$$\ln(|\mathbf{c}|)-2\ln(p(i))+(\mathbf{x}^f-\mu_i)^T\mathbf{c}^{-1}(\mathbf{x}^f-\mu_i) < \ln(|\mathbf{c}|)-2\ln(p(j))+(\mathbf{x}^f-\mu_j)^T\mathbf{c}^{-1}(\mathbf{x}^f-\mu_j) \tag{C.9}$$

it can be written $(\mathbf{x}^f - \mu_i)^T\mathbf{c}^{-1}(\mathbf{x}^f - \mu_i)$ into

$$\mathbf{x}^f\mathbf{c}^{-1}\mathbf{x}^{fT} - 2\mu_i\mathbf{c}^{-1}\mathbf{x}^{fT} + \mu_i\mathbf{c}^{-1}\mu_i^T \tag{C.10}$$

thus, after summarizing the inequality becomes:

$$-2\ln(p(i)) - 2\mu_i\mathbf{c}^{-1}\mathbf{x}^{fT} + \mu_i\mathbf{c}^{-1}\mu_i^T < -2\ln(p(j)) - 2\mu_j\mathbf{c}^{-1}\mathbf{x}^{fT} + \mu_j\mathbf{c}^{-1}\mu_j^T \tag{C.11}$$

multiplying both sides of inequality with $-\frac{1}{2}$ (the sign of inequality reverse because of multiplying with negative value), it defined by:

$$\ln(p(i)) + \mu_i\mathbf{c}^{-1}\mathbf{x}^{fT} - \frac{1}{2}\mu_i\mathbf{c}^{-1}\mu_i^T > \ln(p(j)) + \mu_j\mathbf{c}^{-1}\mathbf{x}^{fT} - \frac{1}{2}\mu_j\mathbf{c}^{-1}\mu_j^T \tag{C.12}$$

Let $f_i = \ln(p(i)) + \mu_i\mathbf{c}^{-1}\mathbf{x}^{fT} - \frac{1}{2}\mu_i\mathbf{c}^{-1}\mu_i^T$, it will assign object with measurement $\mathbf{x}^f$ to i if:

$$f_i > f_j; \quad i \neq j \tag{C.13}$$

that is linear discriminant function.

Thus, LDA has assumption of multivariate normal distribution and all groups have the same covariance matrix.

Bibliography

[1] G. Donato, M. S. Bartlett, J. C. Hager, P. Ekman, and T. J. Sejnowski, "Classifying facial actions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 10, pp. 974–989, 1999.

[2] A. Mehrabian, *Nonverbal communication*. Aldine, 2007.

[3] P. Ekman, E. T. Rolls, D. I. Perrett, and H. D. Ellis, "Facial expressions of emotion: An old controversy and new findings [and discussion]," *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, vol. 335, no. 1273, pp. 63–69, 1992.

[4] P. Ekman, W. Friesen, and P. Ellsworth, *Emotion in the human face: guidelines for research and an integration of findings*. Pergamon General Psychology Series, Pergamon Press, 1972.

[5] I. A. Essa and A. P. Pentland, "Facial expression recognition using a dynamic model and motion energy," in *Proceeeding of the Fifth International Conference on Computer Vision (ICCV'95)*, (Massachusetts, USA), pp. 360–367, June 1995.

[6] S. M. Lajevardi and Z. M. Hussain, "Emotion recognition from color facial images based on multilinear image analysis and log-Gabor filters," in *Proceeding of the 25th International Conference on Image and Vision Computing New Zealand (IVCNZ 2010)*, (Queenstown, New Zealand), pp. 10–14, December 2010.

[7] S. M. Lajevardi and Z. M. Hussain, "Higher order orthogonal moments for invariant facial expression recognition," *Digital Signal Processing*, vol. 20, pp. 1771–1779, 2010.

[8] S. M. Lajevardi and Z. M. Hussain, "Contourlet structural similarity for facial expression recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2010)*, (Dallas, Texas, USA), pp. 1118–1121, March 2010.

[9] S. M. Lajevardi and Z. M. Hussain, "Novel higher-order local autocorrelation-like feature extraction methodology for facial expression recognition," *IET on Image Processing*, vol. 4, pp. 114–119, 2010.

[10] S. M. Lajevardi and Z. M. Hussain, "Feature extraction for facial expression recognition based on hybrid face regions," *Advances in Electrical and Computer Engineering*, vol. 9, no. 3, pp. 63–67, 2009.

[11] S. M. Lajevardi and M. Lech, "Facial expression recognition from image sequences using optimized feature selection," in *Proceeding of the 23rd International Conference on Image and Vision Computing New Zealand (IVCNZ'08)*, (Christchurch, New Zealand), pp. 1–6, November 2008.

[12] B. Fasel and J. Luettin, "Automatic facial expression analysis: a survey," *Pattern Recognition*, vol. 36, no. 1, pp. 259 – 275, 2003.

[13] T. Kanade, Y. Tian, and J. F. Cohn, "Comprehensive database for facial expression analysis," in *Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition*, (Grenoble, France), pp. 46–53, March 2000.

[14] P. Ekman and E. Rosenberg, *What the face reveals: basic and applied studies of spontaneous expression using the facial action coding system (FACS)*. Series in affective science, Oxford University Press, 1997.

[15] J. J. Gross and R. W. Levenson, "Emotion elicitation using films," *Cognition & Emotion*, vol. 9, no. 1, pp. 87–108, 1995.

[16] J. Sung and D. Kim, "Pose-robust facial expression recognition using view-based 2D + 3D AAM," *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans*, vol. 38, no. 4, pp. 852–866, 2008.

[17] M. Pantic and I. Patras, "Dynamics of facial expression: recognition of facial actions and their temporal segments from face profile image sequences," *IEEE Transactions on Systems, Man, and Cybernetics: Part B*, vol. 36, pp. 433–449, 2006.

[18] M. K. M. Lyons, S. Akamatsu and J. Gyoba, "Coding facial expressions with Gabor wavelets," in *Proceedings of the 3rd International Conference on Face and Gesture Recognition (FG'98)*, (Nara, Japan), pp. 200–205, April 1998.

[19] R. C. Gonzalez, R. E. Woods, and S. L. Eddins, *Digital Image Processing Using MATLAB*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2003.

[20] I. Guyon, S. Gunn, M. Nikravesh, and A. Zadeh, *Feature Extraction Foundations and Applications,*. Springer, 2006.

[21] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, pp. 1226–1238, 2005.

[22] C. C. Chibelushi and F. Bourel, "Facial expression recognition: a brief tutorial overview," *CVonline: On-Line Compendium of Computer Vision*, vol. 9, pp. 1–5, 2003.

[23] H. A. Rowley, S. Baluja, and T. Kanade, "Neural network-based face detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, pp. 23–28, 1998.

[24] K. kay Sung and T. Poggio, "Example-based learning for view-based human face detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 1, pp. 39–51, 1998.

[25] P. Viola and M. J. Jones, "Robust real-time face detection," *International Journal of Computer Vision*, vol. 57, pp. 137–154, 2004.

[26] M. Bichsel and A. P. Pentland, "Human face recognition and the face image set's topology," *CVGIP: Image Understanding*, vol. 59, no. 2, pp. 254–261, 1994.

[27] S. Z. Li and A. K. Jain, *Handbook of face recognition*. Springer, 2005.

[28] M. H. Yang, D. J. Kriegman, and N. Ahuja, "Detecting faces in images: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 34–58, 2002.

[29] A. Mohan, C. Papageorgiou, and T. Poggio, "Example-based object detection in images by components," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, pp. 349–361, 2001.

[30] B. Heisele, T. Serre, and T. Poggio, "A component-based framework for face detection and identification," *International Journal of Computer Vision*, vol. 74, no. 2, pp. 167–181, 2007.

[31] Y. Amit and D. Geman, "A computational model for visual selection," *Neural Computation*, vol. 11, no. 7, pp. 1691–1715, 1999.

[32] F. Fleuret and I. Guyon, "Fast binary feature selection with conditional mutual information," *Journal of Machine Learning Research*, vol. 5, pp. 1531–1555, 2004.

[33] C. Papageorgiou and T. Poggio, "A trainable system for object recognition," *International Journal of Computer Vision*, vol. 38, pp. 15–33, 2000.

[34] P. Viola, M. J. Jones, and D. Snow, "Detecting pedestrians using patterns of motion and appearance," *International Journal of Computer Vision*, vol. 63, pp. 153–161, 2005.

[35] P. Dollar, Z. Tu, H. Tao, and S. Belongie, "Feature mining for image classification," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'07)*, (Minneapolis, Minnesota, USA), pp. 1–8, June 2007.

[36] C. Huang, H. Ai, Y. Li, and S. Lao, "High-performance rotation invariant multiview face detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29.

[37] F. Fleuret and D. Geman, "Coarse-to-fine face detection," *International Journal of Computer Vision*, vol. 41, pp. 85–107, 2001.

[38] I. Pitas, *Digital image processing algorithms and applications*. A Wiley-Interscience publication, Wiley, 2000.

[39] S. Z. Li, A. K. Jain, Y.-L. Tian, T. Kanade, and J. F. Cohn, "Facial expression analysis," in *Handbook of Face Recognition*, pp. 247–275, Springer New York, 2005.

[40] A. M. Martinez and A. C. Kak, "Pca versus lda," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, pp. 228–233, 2001.

[41] M. Rosenblum, Y. Yacoob, and L. S. Davis, "Human expression recognition from motion using a radial basis function network architecture," *IEEE Transactions on Neural Network*, vol. 7, pp. 1121–1138, 1996.

[42] I. A. Essa and A. P. Pentland, "Coding, analysis, interpretation, and recognition of facial expressions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 757–763, 1997.

[43] Y. Yacoob and L. Davis, "Recognizing human facial expressions from long image sequences using optical flow," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 6, pp. 636–642, 1996.

[44] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification (2nd Edition)*. Wiley-Interscience, 2001.

[45] M. S. Bartlett, J. R. Movellan, P. Ekman, G. Donato, J. C. Hager, and T. J. Sejnowski, "Image representations for facial expression coding," *Advances in Neural Information Processing Systems*, vol. 12, pp. 886–892, 2000.

[46] L. I. Smith, "Tutorial on principal component analysis," tech. rep., Cornell University, USA, Feb. 2002.

[47] L. Sirovich and M. Kirby, "Low-dimensional procedure for the characterization of human faces," *Journal of the Optical Society of America A*, vol. 4, no. 3, pp. 519–524, 1987.

[48] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.

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

[50] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active appearance models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 681–685, 2001.

[51] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artificial Intelligence*, vol. 97, no. 1-2, pp. 273–324, 1997.

[52] L. C. Molina, L. Belanche, and A. Nebot, "Feature selection algorithms: a survey and experimental evaluation," in *Proceedings of the IEEE International Conference on Data Mining (ICDM'02)*, (Maebashi City, Japan), pp. 306–313, December 2002.

[53] R. Battiti, "Using mutual information for selecting features in supervised neural net learning," *IEEE Transactions on Neural Networks*, vol. 5, no. 4, pp. 537–550, 1994.

[54] A. Colmenarez, B. Frey, and T. S. Huang, "A probabilistic framework for embedded face and facial expression recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'99)*, (Fort Collins, CO, USA), pp. 592–597, June 1999.

[55] L. Ma and K. Khorasani, "Facial expression recognition using constructive feedforward neural networks," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 34, pp. 1588–1595, 2004.

[56] P. Michel and R. E. Kaliouby, "Real time facial expression recognition in video using support vector machines," in *Proceedings of the 5th International Conference on Multimodal Interfaces (ICMI)*, (Vancouver, Canada), pp. 258–264, November 2003.

[57] B. V. Dasarathy, *Nearest neighbor (NN) norms: NN pattern classification techniques*. Los Alamitos: IEEE Computer Society Press, 1990.

[58] D. A. Forsyth and J. Ponce, *Computer Vision: A Modern Approach*. Prentice Hall, 2002.

[59] I. G. B. Boser and V. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, (Pittsburgh, PA, USA), pp. 144–152, July 1992.

[60] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and other Kernel-based Learning Methods.* Cambridge University Press, 2000.

[61] C.Cortes and V.Vapnik, "Support vector networks," *Machine Learning*, vol. 20, pp. 273–297, 1995.

[62] Y. Lijun, C. Xiaochen, S. Yi, T. Worm, and M. Reale, "A high-resolution 3D dynamic facial expression database," in *Proceedings of the 3rd International Conference on Face and Gesture Recognition (FG'08)*, (Amsterdam, Netherlands), pp. 1–6, September 2008.

[63] E. R. Dougherty, *An introduction to morphological image processing.* SPIE Optical Engineering Press, Bellingham, Wash., USA, 1992.

[64] A. Nikolaidis and I. Pitas, "Facial feature extraction and pose determination," *Pattern Recognition*, vol. 33, pp. 1783–1791, 2000.

[65] M. A. O. Vasilescu and D. Terzopoulos, "Multilinear image analysis for facial recognition," in *Proceeding of the International Conference on Pattern Recognition (ICPR'02)*, (Quebec City, Canada), pp. 511–514, August 2002.

[66] M. Thomas, C. Kambhamettu, and S. Kumar, "Face recognition using a color subspace lda approach," in *Proceeding of the 20th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'08)*, (Dayton, OH, USA), pp. 231–235, November 2008.

[67] K. Plataniotis and A. Venetsanopoulos, *Color image processing and applications.* Digital signal processing, Springer, 2000.

[68] S. Schulte, V. D. Witte, M. Nachtegael, D. V. D. Weken, and E. E. Kerre, "Fuzzy two-step filter for impulse noise reduction from color images," *IEEE Transactions on Image Processing*, vol. 15, no. 11, pp. 3567–3578, 2006.

[69] Z. Xu, H. R. Wu, B. Qiu, and X. Yu, "Geometric features-based filtering for suppression of impulse noise in color images," *IEEE Transactions on Image Processing*, vol. 18, no. 8, pp. 1742–1759, 2009.

[70] B. K. Horn, *Robot Vision*. McGraw-Hill Higher Education, 1st ed., 1986.

[71] J. Wang, L. Yin, X. Wei, and Y. Sun, "3D facial expression recognition based on primitive surface feature distribution," in *Proceeding of IEEE Conference on Computer Vision and Pattern Recognition (CVPR'06)*, (New York, NY, USA), pp. 1399–1406, June 2006.

[72] T. S. Lee, "Image representation using 2D Gabor wavelets," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, pp. 959–971, 1996.

[73] J. G. Daugman, "Complete discrete 2-D Gabor transforms by neural networks for image analysis and compression," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 36, no. 7, pp. 1169–1179, 1988.

[74] D. Zheng, Y. Zhao, and J. Wang, "Feature selection for high-dimensional genomic microarray data," in *Proceedings of the Sixth IASTED International Conference Signal and Image Processing (SIP 2004)*, (Hawaii, USA), pp. 601–608, August 2004.

[75] A. K. Jain and F. Farrokhnia, "Unsupervised texture segmentation using Gabor filters," *Pattern Recognition*, vol. 24, no. 12, pp. 1167–1186, 1991.

[76] H. Feichtinger and T. Strohmer, *Gabor analysis and algorithms: theory and applications*. Applied and numerical harmonic analysis, Birkh "auser, 1998.

[77] S. M. Lajevardi and M. Lech, "Averaged Gabor filter features for facial expression recognition," in *Proceeding of the Digital Image Comput-*

ing: Techniques and Applications (DICTA '08), (Canberra, Australia), pp. 71–76, December 2008.

[78] S. M. Lajevardi and M. Lech, "Facial expression recognition using neural networks and log-Gabor filters," in *Proceeding of the Digital Image Computing: Techniques and Applications (DICTA '08)*, (Canberra, Australia), pp. 77–83, December 2008.

[79] S. M. Lajevardi and Z. M. Hussain, "A novel Gabor filter selection based on spectral difference and minimum error rate for facial expression recognition," in *Proceeding of the Digital Image Computing: Techniques and Applications(DICTA 2010)*, (Sydney, Australia), pp. 137–140, December 2010.

[80] D. J. Field, "Relations between the statistics of natural images and the response properties of cortical cells," *Journal of the Optical Society of America A*, vol. 4, pp. 2379–2394, Dec 1987.

[81] S. M. Lajevardi, K. L. Neville, and Z. M. Hussain, "Facial expression recognition over fft-ofdm," in *Proceeding of the International Conference on Advanced Technologies for Communications (ATC'09)*, (Haiphong, Vietnam), pp. 35–38, October 2009.

[82] D. D. Y. Po and M. N. Do, "Directional multiscale modeling of images using the contourlet transform," *IEEE Transactions on Image Processing*, vol. 15, no. 6, pp. 1610–1620, 2006.

[83] M. N. Do, *Directional multiresolution image representations*. Phd thesis, École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland, 2001.

[84] P. Burt and E. Adelson, "The laplacian pyramid as a compact image code," *IEEE Transactions on Communications*, vol. 31, no. 4, pp. 532–540, 1983.

[85] R. H. Bamberger and M. J. T. Smith, "A filter bank for the directional decomposition of images: theory and design," *IEEE Transactions on Signal Processing*, vol. 40, no. 4, pp. 882–893, 1992.

[86] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on feature distributions," *Pattern Recognition*, vol. 29, pp. 51–59, 1996.

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

[88] S. M. Lajevardi and Z. M. Hussain, "Local feature extraction methods for facial expression recognition," in *Proceeding of the 17th European Signal Processing Conference (EUSIPCO 2009)*, (Glasgow, Scotland), pp. 60–64, Augest 2009.

[89] S. M. Lajevardi and Z. M. Hussain, "Facial expression recognition using log-Gabor filters and local binary pattern operators," in *Proceeding of the International Conference on Communication, Computer and Power (ICCCP'09)*, (Muscat, Oman), pp. 349–353, Feb 2009.

[90] S. Smith, *The scientist and engineer's guide to digital signal processing*. California Technical Publication, 1997.

[91] C. Nikias and A. Petropulu, *Higher-order spectra analysis: a nonlinear signal processing framework*. Prentice Hall signal processing series, PTR Prentice Hall, 1993.

[92] N. Otsu and T. Kurita, "A new scheme for practical flexible and intelligent vision systems," in *Proceedings of the IAPR Workshop on Computer Vision*, (Tokyo, Japan), pp. 431–435, October 1988.

[93] T. Toyoda and O. Hasegawa, "Texture classification using extended higher order local autocorrelation features," in *Proceedings of the 4th IEEE International Workshop on Texture Analysis and Synthesis in conjunction with ICCV'05*, (Beijing, China), pp. 131–136, October 2005.

[94] M. R. Teague, "Image analysis via the general theory of moments," *Journal of the Optical Society of America*, vol. 70, no. 8, pp. 920–930, 1980.

[95] C. H. Teh and R. T. Chin, "On image analysis by the method of moments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, no. 4, pp. 496–513, 1998.

[96] M. K. Hu, "Visual pattern recognition by moment invariants," *IRE Transactions on Information Theory*, vol. 8, pp. 179–187, 1962.

[97] R. R. Bailey and M. Srinath, "Orthogonal moment features for use with parametric and non-parametric classifiers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 4, pp. 389–398, 1995.

[98] A. Khotanzad and Y. H. Hong, "Invariant image recognition by zernike moments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 5, pp. 489–497, 1990.

[99] H. S. Kim and H. K. Lee, "Invariant image watermark using zernike moments," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 8, pp. 766–775, 2003.

[100] S. M. Lajevardi and Z. M. Hussain, "Zernike moments for facial expression recognition," in *Proceeding of the International Conference on Communication, Computer and Power (ICCCP'09)*, (Muscat, Oman), pp. 378–381, Feb 2009.

[101] R. Zhi and Q. Ruan, "A comparative study on region-based moments for facial expression recognition," in *Proceedings of the Congress on Image*

and Signal Processing (CISP'08), (Sanya, Hainan, China), pp. 600–604, May 2008.

[102] F. Liu, Z. Wang, L. Wang, and X. Meng, "Facial expression recognition using hlac features and wpca," in *Affective Computing and Intelligent Interaction, First International Conference (ACII 2005)*, (Beijing, China), pp. 88–94, October 2005.

[103] S. M. Lajevardi and Z. M. Hussain, "Hybrid feature extraction for facial expression recognition," *Advances in Modelling Series B: Signal Processing and Pattern Recognition*, vol. 53, pp. 34–50, 2009.

[104] S. Das, "Filters, wrappers and a boosting-based hybrid for feature selection," in *Proceeding of the Eighteenth International Conference on Machine Learning (ICML '01)*, (Williams College, Williamstown, MA, USA), pp. 74–81, June 2001.

[105] A. L. Blum and P. Langley, "Selection of relevant features and examples in machine learning," *Artificial Intelligence on Relevance*, vol. 97, pp. 245–271, 1997.

[106] H. Liu and H. Motoda, *Feature selection for knowledge discovery and data mining*. Kluwer international series in engineering and computer science, Kluwer Academic Publishers, 1998.

[107] J. Doak, "An evaluation of feature-selection methods and their application to computer security," technical report (cse-92-18), University of California, Davis, 1992.

[108] N. Kwak and C. H. Choi, "Input feature selection for classification problems," *IEEE Transactions on Neural Networks*, vol. 13, no. 1, pp. 143–159, 2002.

[109] T. M. Mitchell, "Generalization as search," *Artificial Intelligence*, vol. 18, pp. 203–226, 1982.

[110] G. H. John, R. Kohavi, and K. Pfleger, "Irrelevant features and the subset selection problem.," in *Proceeding of Eleventh International Conference on Machine Learning (ICML'94)*, (Rutgers University, New Brunswick, NJ, USA), pp. 121–129, July 1994.

[111] C. Jutten and J. Herault, "Blind separation of sources, part 1: an adaptive algorithm based on neuromimetic architecture," *Signal Processing*, vol. 24, no. 1, pp. 1–10, 1991.

[112] J. Bins and B. A. Draper, "Feature selection from huge feature sets," in *The 8th International Conference On Computer Vision (ICCV-01)*, (Vancouvar, Canada), pp. 159–165, July 2001.

[113] E. P. Xing, M. I. Jordan, and R. M. Karp, "Feature selection for high-dimensional genomic microarray data," in *Proceedings of the Eighteenth International Conference on Machine Learning (ICML '01)*, (Williams College, Williamstown, MA, USA), pp. 601–608, June 2001.

[114] S. M. Lajevardi and Z. M. Hussain, "Feature selection for facial expression recognition based on mutual information," in *Proceeding of the 5th IEEE GCC Conference and Exhibition*, (Kuwait City, Kuwait), pp. 1–5, March 2009.

[115] A. L. Blum and R. L. Rivest, "Training a 3-node neural network is np-complete," *Neural Networks*, vol. 5, no. 1, pp. 117 – 127, 1992.

[116] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[117] J. H. Holland, *Adaptation in natural and artificial systems.* Cambridge, MA, USA: MIT Press, 1992.

[118] L. B. Booker, D. E. Goldberg, and J. H. Holland, "Classifier systems and genetic algorithms," *Artificial Intelligence*, vol. 40, pp. 235–282, 1989.

[119] S. M. Lajevardi and Z. M. Hussain, "Feature selection for facial expression recognition based on optimization algorithm," in *Proceeding of the Second International Workshop on Nonlinear Dynamics and Sunchronization (INDS'09)*, (Klagenfurt, Austria), pp. 182–185, July 2009.

[120] M. Srinivas and L. M. Patnaik, "Genetic algorithms: A survey," *Journal of the Optical Society of America A*, vol. 27, pp. 17–26, 1994.

[121] I. Rish, "An empirical study of the naive bayes classifier," in *IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence*, (Seattle, Washington, USA), pp. 41–46, August 2001.

[122] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Transaction on Image Processing*, vol. 13, pp. 600–612, 2004.

[123] S. M. Lajevardi and Z. M. Hussain, "Local correlation for noisy facial expression images," in *Proceeding of the International Symposium on Bioelectronics and Bioinformatics*, (Melbourne, Australia), pp. 64–67, December 2009.

[124] M. P. Sampat, W. Zhou, S. Gupta, A. C. Bovik, and M. K. Markey, "Complex wavelet structural similarity: a new image similarity index," *IEEE Transactions on Image Processing*, vol. 18, no. 11, pp. 2385–2401, 2009.

[125] L. Torres, J. Y. Reutter, and L. Lorente, "The importance of the color information in face recognition," in *Proceedings of the International Conference on Image Processing (ICIP'99)*, vol. 2, (Kobe, Japan), pp. 627–631, October 1999.

[126] P. Shih and C. Liu, "Comparative assessment of content-based face image retrieval in different color spaces," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 19, no. 7, pp. 873–893, 2005.

[127] Z. Liu and C. Liu, "A hybrid color and frequency features method for face recognition," *IEEE Transactions on Image Processing*, vol. 17, pp. 1975–1980, 2008.

[128] C. J. Young, R. Y. Man, and K. N. Plataniotis, "Color face recognition for degraded face images," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 39, pp. 1217–1230, 2009.

[129] M. Corbalán, M. S. Millán, and M. J. Yzuel, "Color pattern recognition with CIELab coordinates," *Optical Engineering*, vol. 41, no. 1, pp. 130–138, 2002.

[130] G. Finlayson, B. Schiele, and J. Crowley, "Comprehensive colour image normalization," in *Proceedings of the Fifth European Conference on Computer Vision (ECCV'98)*, (Freiburg, Germany), pp. 475–490, June 1998.

[131] G. Wyszecki and W. Stiles, *Color science: concepts and methods, quantitative data, and formulae*. Wiley classics library, John Wiley & Sons, 2000.

[132] C. A. Curcio, K. R. Sloan, R. E. Kalina, and A. E. Hendrickson, "Human photoreceptor topography," *The Journal of Comparative Neurology*, vol. 292, pp. 497–523, 1990.

[133] B. Wandell, *Foundations of vision*. Sinauer Associates, 1995.

[134] T. Smith and J. Guild, "The C.I.E. colorimetric standards and their use," *Transactions of the Optical Society*, vol. 33, no. 3, pp. 73–134, 1931.

[135] M. Fairchild, *Color appearance models*. Wiley-IS&T series in imaging science and technology, J. Wiley, 2005.

[136] D. Jobson, Z. Rahman, and G. Woodell, "A multiscale retinex for bridging the gap between color images and the human observation of scenes," *IEEE Transactions on Image Processing*, vol. 6, no. 7, pp. 965–976, 1997.

[137] M. Bertalmío, V. Caselles, and E. Provenzi, "Issues about retinex theory and contrast enhancement," *International Journal of Computer Vision*, vol. 83, no. 1, pp. 101–119, 2009.

[138] X. Xie and K.-M. Lam, "An efficient illumination normalization method for face recognition," *Pattern Recognition Letters*, vol. 27, no. 6, pp. 609–617, 2006.

[139] S. M. Lajevardi and Z. M. Hussain, "Automatic facial expression recognition: feature extraction and selection," *Signal, Image and Video Processing*, pp. 1–11, 2010.

VITA

Seyed Mehdi Lajevardi received his B.Eng. in electronic engineering from Islamic Azad University in 2007, Tehran, Iran. From 2008-2011, he was PhD candidate at RMIT University in Melbourne, Australia. Currently, he is sessional lecturer for computer vision system at Swinburne University of Technology, Melbourne, Australia. His research interests include digital signal processing, pattern recognition, image processing and computer vision.