2010

Adaptive feature extraction and selection for robust facial expression recognition

Peiyao Li
University of Wollongong

Recommended Citation
NOTE

This online version of the thesis may have different page formatting and pagination from the paper copy held in the University of Wollongong Library.

UNIVERSITY OF WOLLONGONG

COPYRIGHT WARNING

You may print or download ONE copy of this document for the purpose of your own research or study. The University does not authorise you to copy, communicate or otherwise make available electronically to any other person any copyright material contained on this site. You are reminded of the following:

Copyright owners are entitled to take legal action against persons who infringe their copyright. A reproduction of material that is protected by copyright may be a copyright infringement. A court may impose penalties and award damages in relation to offences and infringements relating to copyright material. Higher penalties may apply, and higher damages may be awarded, for offences and infringements involving the conversion of material into digital or electronic form.
Adaptive Feature Extraction and Selection for Robust Facial Expression Recognition

A thesis submitted in partial fulfilment of the requirements for the award of the degree

Master of Engineering by Research

from

UNIVERSITY OF WOLLONGONG

by

Peiyao Li

School of Electrical, Computer and Telecommunications Engineering

October 2010
Statement of Originality

I, Peiyao Li, declare that this thesis, submitted in partial fulfilment of the requirements for the award of Master of Engineering - Research, in the School of Electrical, Computer and Telecommunications Engineering, University of Wollongong, is wholly my own work unless otherwise referenced or acknowledged. The document has not been submitted for qualifications at any other academic institution.

Peiyao Li

October 14, 2010
Contents

Acronyms

Abstract

Acknowledgments

1 Introduction

1.1 Research objectives

1.2 Thesis organisation

1.3 Contributions

1.4 Publications

2 Literature Review

2.1 Geometric-based approach

2.2 Appearance-based approach

2.3 Fusion feature-based approach

2.4 Spatio-temporal features

2.4.1 Feature point tracking based approach

2.4.2 Model-based approach
## 4 Facial Expression Recognition System

### 4.1 System architecture overview

### 4.2 Stage 1: Fixed, directional filters

### 4.3 Stage 2: Adaptive filters

### 4.4 Stage 3: Classification

### 4.5 Training methods

#### 4.5.1 Resilient backpropagation algorithm

#### 4.5.2 Levenberg-Marquardt algorithm

### 4.6 Support vector machines

#### 4.6.1 Mathematical background

#### 4.6.2 Non-linear SVMs

#### 4.6.3 Multi-class SVMs

### 4.7 Feature selection

#### 4.7.1 General characteristics of feature selection

#### 4.7.2 General feature selection algorithms

#### 4.7.3 Proposed feature selection method

### 4.8 Chapter summary

## 5 Experimental Results

### 5.1 Databases and experimental steps

#### 5.1.1 JAFFE and MMI databases

#### 5.1.2 Experimental steps and parameters

### 5.2 Results of eye detection and face alignment

### 5.3 Results of facial expression recognition
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.3.1 FER result using linear classifiers</td>
<td>82</td>
</tr>
<tr>
<td>5.3.2 FER result using SVMs</td>
<td>83</td>
</tr>
<tr>
<td>5.3.3 FER results of two-class classification</td>
<td>85</td>
</tr>
<tr>
<td>5.4 Results of feature selection</td>
<td>86</td>
</tr>
<tr>
<td>5.5 Comparison with other FER methods</td>
<td>89</td>
</tr>
<tr>
<td>5.6 Chapter summary</td>
<td>90</td>
</tr>
<tr>
<td>6 Conclusion</td>
<td>91</td>
</tr>
<tr>
<td>6.1 Research summary</td>
<td>91</td>
</tr>
<tr>
<td>6.2 Future work</td>
<td>93</td>
</tr>
<tr>
<td>6.3 Conclusion</td>
<td>94</td>
</tr>
<tr>
<td>References</td>
<td>96</td>
</tr>
</tbody>
</table>
List of Figures

1.1 A seal-mimetic robot therapy at a nursing service centre. (a) A seal-mimetic robot PARO. (b) The robot is interacting with aged people. .......................................................... 2

1.2 Examples of major facial expressions. ................................. 3

1.3 General procedures of a facial expression recognition system. . 4

2.1 Major stages in a typical automatic FER system. ................. 10

2.2 Existing techniques and procedures of facial feature extraction. 11

2.3 Geometric feature points (fiducial points of the contours of the face) [1]. ................................................................. 12

2.4 Seven facial areas for feature extraction [2]. .......................... 15

2.5 The geometrical relationships of facial feature points, where the rectangles represent the regions of furrows and wrinkles [3]. . . . 16

2.6 12 facial feature points to be tracked. .................................. 18

2.7 (a) The 3-D face model with manually selected feature points. (b) The face model are fitting with the face scan data [4]. ............ 19

2.8 Example of optic flow experienced by a rotating observer [5]. . . 21
2.9 Facial expression decomposition to person subspace, expression subspace, and feature subspace[6]. ........................................ 27

3.1 Four basic types of rectangle features [7]: (a) and (b) are two-rectangle features, (c) is a three-rectangle feature, and (d) is a four-rectangle feature. ................................................................. 41

3.2 An example of computing the rectangle feature value. Note that each small rectangle indicates a pixel value. ................................. 41

3.3 An example of OpenCV AdaBoost-based face detection. ................. 42

3.4 An example result of the OpenCV face detector. Even the face is found, its coordinates and rotation angle can be better estimated. . 43

3.5 Examples of image preprocessing. (a) input image. (b) histogram equalization. (c) illumination normalisation. (d) contrast-normalisation. 44

3.6 Steps for finding eye candidates using the horizontal non-linear filter: (a) output of the non-linear filter, (b) connected component labelling, (c) image erosion and dilation, (d) candidate eye points (shown in red colour). ......................................................... 45

3.7 Filters for eye candidate detection (a) Gabor filter (b) Circular filter. 46

3.8 Steps for finding eye candidates using eye filters: (a) eye filter output, (b) connected component labelling, (c) image erosion and dilation, (d) candidate eye points (shown in red colour). ....................... 47

3.9 Determining the face boundary for a given eye pair using a geometric face model [8]. ................................................................. 49
3.10 A face template constructed using 15,000 frontal face images from the ECU database [9].................. 50

3.11 Eye detection and face alignment: (a) valid eye points after face verification, (b) aligned face pattern. .................. 50

3.12 An example of multiple eye detection. (a) The original image. (b) Eye detection results on the original image. .................. 51

4.1 Block diagram of the proposed system. .................. 54

4.2 The sub-sampling operations performed in stage one. ............... 56

4.3 The sub-sampling operations performed in stage two. ............... 58

4.4 Examples of different linear classifiers. .................. 59

4.5 (a) a number of hyperplanes that can separate the two classes (rectangles and circles). (b) the hyperplane generated by SVM which has the maximal margin. .................. 64

4.6 An example of separating hyperplane and support hyperplane. .... 66

4.7 Classify in high dimensional feature space. (a) features in the input space. (b) mapping the features into high dimensional space. .... 67

4.8 Apply SVMs to multi-class problem (4 classes). .................. 68

4.9 Feature selection procedures. .................. 71

5.1 Example facial expression images in JAFFE database. .................. 79

5.2 Example facial expression images in MMI database. .................. 80

5.3 A visual result of the automatic system: (a) original colour image, (b) grey scale image, (c) eye detection, (d) expression recognition. .... 87
5.4 Locations of selected features superimposed on a face image as yellow-red patches. The first four images correspond to features in the four directions, $\theta = 0^\circ, 45^\circ, 90^\circ, \text{and} 135^\circ$. The last image shows the selected features, combined from all directions. 88

5.5 System performance using feature selection method one. 88

5.6 System performance using feature selection method two. 89
List of Tables

2.1 Existing FER systems and performances. .......................... 34

3.1 Eye detection performances of the horizontal filter and eye filter
tested on the JAFFE database. ........................................... 51

5.1 Comparison of different values for M - the order of Gaussian deriva-
tive filters. ................................................................. 82

5.2 Eye detection performance on the JAFFE database. .............. 82

5.3 Classification rates for different facial expression categories (clas-
sifier using mirror image). The entry at (row r, column c) is the
percentage of facial expression r that is classified as facial expres-
sion c. ................................................................. 84

5.4 Classification rates for different facial expression categories using
FER system with feature selection. ................................. 84

5.5 Classification rates tested on MMI database. ....................... 85

5.6 Classification rates for different two-class facial expressions. ... 86

5.7 Comparison of recognition accuracies for non-aligned and aligned
faces on the JAFFE database. ......................................... 86
5.8  Confusion matrix for the two facial expressions. The entry at (row $r$, column $c$) is the percentage of facial expression $r$ that is classified as facial expression $c$. ........................................ 86

5.9  Classification rates of FER methods on the JAFFE database. .......... 90
### Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAM</td>
<td>Active appearance model</td>
</tr>
<tr>
<td>ANN</td>
<td>Artificial neural network</td>
</tr>
<tr>
<td>AWN</td>
<td>Active wavelet network</td>
</tr>
<tr>
<td>AUs</td>
<td>Action units</td>
</tr>
<tr>
<td>BN</td>
<td>Bayesian network</td>
</tr>
<tr>
<td>CSS</td>
<td>Class separation score</td>
</tr>
<tr>
<td>CDA</td>
<td>Clustering based discriminant analysis</td>
</tr>
<tr>
<td>CF</td>
<td>Certainty factor</td>
</tr>
<tr>
<td>CMU</td>
<td>Carnegie mellon university</td>
</tr>
<tr>
<td>CR</td>
<td>Classification rate</td>
</tr>
<tr>
<td>DCT</td>
<td>Discrete cosine transform</td>
</tr>
<tr>
<td>DSI</td>
<td>Directional shunting inhibition</td>
</tr>
<tr>
<td>FACS</td>
<td>Facial action coding system</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------------------------------------------------------</td>
</tr>
<tr>
<td>FAPs</td>
<td>Facial animation parameters</td>
</tr>
<tr>
<td>FER</td>
<td>Facial expression recognition</td>
</tr>
<tr>
<td>FSLP</td>
<td>Feature selection linear programming</td>
</tr>
<tr>
<td>GWN</td>
<td>Gabor wavelet network representation</td>
</tr>
<tr>
<td>HLAC</td>
<td>Higher-order local auto-correlation</td>
</tr>
<tr>
<td>HMM</td>
<td>Hidden markov model</td>
</tr>
<tr>
<td>HOSVD</td>
<td>Higher-order singular value decomposition</td>
</tr>
<tr>
<td>ICA</td>
<td>Independent component analysis</td>
</tr>
<tr>
<td>IR</td>
<td>Infrared</td>
</tr>
<tr>
<td>JAFFE</td>
<td>Japanese ATR female facial expression database</td>
</tr>
<tr>
<td>KLT</td>
<td>Kanade-lucas-tomasi tracker</td>
</tr>
<tr>
<td>LBP</td>
<td>Local binary patterns</td>
</tr>
<tr>
<td>LDA</td>
<td>Linear discriminant analysis</td>
</tr>
<tr>
<td>LM</td>
<td>Levenberg-marquardt algorithm</td>
</tr>
<tr>
<td>LOO</td>
<td>Leave-one-out</td>
</tr>
<tr>
<td>LOOCV</td>
<td>Leave-one-out cross validation</td>
</tr>
<tr>
<td>MP-PCA</td>
<td>Mixture of probabilistic PCA</td>
</tr>
<tr>
<td>P3-DHMM</td>
<td>Pseudo 3-D HMM</td>
</tr>
<tr>
<td>Acronyms</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>PBVD</td>
<td>Piece-wise baizer volume deformation</td>
</tr>
<tr>
<td>PCA</td>
<td>Principal component analysis</td>
</tr>
<tr>
<td>PDF</td>
<td>Probability density function</td>
</tr>
<tr>
<td>ROIs</td>
<td>Regions of interest</td>
</tr>
<tr>
<td>Rprop</td>
<td>Resilient backpropagation algorithm</td>
</tr>
<tr>
<td>SRM</td>
<td>Structural risk minimisation</td>
</tr>
<tr>
<td>SVM</td>
<td>Support vector machine</td>
</tr>
<tr>
<td>TAN</td>
<td>Tree-augmented-naive bayes classifier</td>
</tr>
</tbody>
</table>
Abstract

Facial expression is one of the means humans convey their emotional states. Accurate recognition of facial expressions should therefore lead to advances in human-computer communication and applications in robotics and mimetic games. This project investigates a novel approach to recognise facial expressions from static images. We propose a method for face alignment to address the localisation error in existing face detection methods, through eye detection and face verification.

In the proposed facial expression recognition approach, fixed and adaptive 2-D filters are combined in a hierarchical structure. The fixed filters are used to extract primitive features such as edges and directions, whereas the adaptive filters are trained to extract more complex and subtle facial features for classification. Both types of filters are non-linear and they are based on the biological mechanism of shunting inhibition. Linear classifier and Support Vector Machines are both used as classifiers in the facial expression recognition system.

To improve the system performance, two feature selection algorithms are proposed to select salient features for classification. One is based on symmetric Kullback-Leibler divergence, the other is based on mutual information. The proposed approaches are evaluated on the JAFFE database, which has seven types of
facial expressions: anger, disgust, fear, happiness, neutral, sadness and surprise. The proposed face detection and alignment method can correctly align and crop all the images from the database. The system with linear classifiers achieves a facial expression classification rate of 95.9%, while system based on SVMs has a higher recognition rate of 96.7%. Using an efficient feature selection method, the system can achieve the best performance with a smaller feature subset. This facial expression recognition system compares favourably with several existing techniques tested on the same database.
Acknowledgments

I would like to express my gratitude to my parents and husband, who have supported me during my research studies.

I also wish to give the deepest appreciation to my principal supervisor, Dr. Son Lam Phung, for all of his time, guidance, counsel, and technical support. Special thanks also go to my co-supervisor Professor Salim Bouzerdoum for all his guidance, assistance, and knowledge.

Moreover, I gratefully acknowledge the support from the research fellow, Fok Tivive, and staff of the School of Electrical, Computer and Telecommunications Engineering. They have provided me personal and professional support during my studies at the University of Wollongong.

Finally, special thanks also go to my fellow students and friends, who have encouraged and helped me during my research journey.
Humans interact with each other in various ways such as speech, eye contact, facial expression, and gesture. Among these, facial expression, controlled by a complex mesh of nerves and muscles beneath the face skin, enables people to convey their emotions and perform nonverbal communications almost instantly. Approximately 55% of information about the emotional states is conveyed through facial expression, and 7% and 33% are conveyed via language and sound, respectively [10]. There are seven basic facial expressions that reflect distinctive psychological activities: anger, disgust, fear, happiness, neutral, sadness and surprise [11]. Examples of these facial expressions are shown in Fig. 1.2.

Facial expression recognition (FER) aims to determine the facial expressions of a person from an image or a video sequence. In recent decades, it has become an important and active area in the field of pattern recognition. FER has a wide range
of applications in human-machine interactions, face biometrics, and psychology studies [12]. Automatic facial expression recognition could bring considerable benefits for human-computer interaction as it provides strong cue in measuring levels of interest of a person while interacting with a machine. Figure 1.1 provides an example when FER has been used in practical applications. A seal-mimetic robot named ”PARO”, which is able to response to humans based on their emotions, is employed at a health service centre to help the aged people. It is proved that this robot can help to increase the interactions among the aged, remedy depressive state, cheer up and motivate people. Researchers have been developing various methods for recognising facial expressions, yet to achieve human-like performance remains elusive.

![Figure 1.1: A seal-mimetic robot therapy at a nursing service centre. (a) A seal-mimetic robot PARO. (b) The robot is interacting with aged people.](http://www.aist.go.jp/aist_e/latest_research/2004/20041208_2/20041208_2.html)

People convey facial expressions via the movements of facial muscles. The facial muscles enable us to make up to 7000 different facial expressions [13]. Ekman and Friesen [14] developed the Facial Action Coding Systems (FACS) to describe each facial expression in terms of visually observable changes in facial muscles.

---

1Online access: http://www.aist.go.jp/aist_e/latest_research/2004/20041208_2/20041208_2.html
Depending on the anatomy characteristics, Ekman and Friesen systematically divided human face into several Action Units, AUs, which expressively reveal the corresponding relationships between facial movements and facial expressions.

![Examples of major facial expressions.](image)

A basic FER system is comprised of three stages: face detection and alignment, facial feature extraction, and facial expression classification, as shown in Fig. 1.3. The input images are firstly sent through the face detection and alignment stage to generate upright face images for further processing. The second stage is to extract facial features from the aligned face images. As salient features can significantly improve the system performance, the feature extraction stage is essential in the FER system. Developing an efficient and robust method for feature extraction is considered to be vital in this project. In the last stage, by choosing an appropriate classifier, the extracted facial features are classified into corresponding facial expression categories.
1.1 Research objectives

This research project aims to develop a novel FER system that is capable of recognising facial expressions with high accuracy. The key research questions of FER are as follows:

- How to detect faces from complex backgrounds to reduce redundant information?
- What method should be adopted to extract salient facial feature for analysis?
- What feature selection methods can be integrated with the FER system to select salient feature for classification?
- Which classification architecture can achieve a satisfactory recognition performance?

The specific objectives of this project are as follows:

- Develop a fast and accurate method for facial expression recognition.
- Evaluate the FER system on standard benchmark databases and compare it with other existing methods.
- Improve the FER method to achieve better performances using fewer facial features.
1.2 Thesis organisation

This thesis consists of six chapters:

- **Chapter 1** outlines the project background and objectives. It highlights the research contributions and publications.

- **Chapter 2** gives a comprehensive literature review on FER. In this chapter, reviews of the state-of-art feature extraction techniques as well as classification approaches are presented.

- **Chapter 3** presents a novel face detection and alignment method. The face images are firstly detected by an OpenCV AdaBoost-based method. Then a face alignment method is proposed. This method consists of two steps. In the first step, two types of filters, namely, horizontal non-linear filters and eye filters, are implemented to generate the possible eye candidates, respectively. Performances of these two filters are analysed and compared. In the next step, we select the accurate eye points based on a template matching approach.

- **Chapter 4** describes the proposed FER system. The proposed system contains three stages. In the first stage, fixed, directional filters are used to extract low-level features. In the second stage, we train the adaptive filters to detect more complex facial features. A linear classifier and SVMs are used in the third stage to make the final decision. In this chapter, the feature selection technique is also discussed.

- **Chapter 5** reveals the results of the proposed FER system. The evaluation
results include face detection and alignment, FER, and FER with feature selection. In this chapter, we also compare our method to several existing methods, on the same database.

- Chapter 6 summarises the research activities and provides the concluding remarks.

1.3 Contributions

The main contributions of this thesis are:

- We propose a FER system that contains fixed, directional filters and adaptive filters connecting in a hierarchical structure. The proposed system is applied to the problem of classifying seven basic facial expressions. The performances of the FER system are evaluated on both JAFFE database and MMI database.

- A linear classifier and SVMs have been studied and employed to classify the facial expressions categories. We use a majority voting approach to generate multi-class SVM classifiers from the existing two-class SVMs. This approach can solve the seven class FER problem using two-class SVM classifiers.

- A novel approach for detecting eye positions and performing face alignment has been proposed. This approach uses the eye filter, which is the combination of a Gabor filter and a circular filter, to process the face image to generate eye candidates. After the eye candidates have been detected, we design a face model and use a template matching method to select the
accurate eye pair to perform face alignment. The proposed eye detection and face alignment method has been evaluated on the JAFFE database.

• To improve the recognition accuracy, we introduce feature selection technique to the FER system. The salient features are selected based on the mutual information. In this thesis, two methods are implemented to generate mutual information for each feature.

1.4 Publications

The publications arised from this Masters research project (August 2009 - October 2010) are listed as follows.


Abstract: We propose a novel approach that combines fixed and adaptive 2-D filters in a hierarchical structure for FER. The fixed, directional filters are used to extract primitive features. They are followed by the adaptive filters that are trained to extract more complex facial features. The features are finally classified by SVMs. The experimental result compares favourably with several existing techniques tested on JAFFE database.

Abstract: This work focuses on the selection of salient features for accurate recognition of facial expressions. It presents a feature selection approach that is based on the reduction of mutual information among the selected features. The classification accuracy has been improved.


Abstract: This paper focuses on recognising the smiling from the neutral facial expression. We propose a method to solve the face alignment problem in existing FER systems. The result shows that the recognition rate can be improved using the aligned faces. This approach is evaluated on JAFFE database and it correctly detects and aligns all face images.
This chapter presents an overview of the existing techniques for automatic facial expression recognition (FER). An automatic FER system typically consists of three main stages: face detection and alignment, facial feature extraction, and classification. Each stage is further divided into subcategories as follows:

1. **Geometric-based approach**
2. **Appearance-based approach**
3. **Fusion feature-based approach**
4. **Spatio-temporal features**
   - Feature point tracking based approach
   - Model-based approach
   - Optical flow based approach
5. **Feature selection and dimensionality reduction**
6. **Facial feature decomposition**
7. **Classification**
   - Rule-based classifier
   - Artificial neural network
   - Bayesian classifier
   - AdaBoost
   - Support vector machine
   - Hidden Markov model
8. **Performance evaluation**
9. **Chapter summary**

This chapter provides a comprehensive review of the methodologies employed in FER, highlighting advancements and challenges in the field.
facial expression classification (see Fig. 2.1). Face detection aims to find the locations of faces in an image. Face alignment is required to normalise the faces to reduce the effects of rotation and imperfect localisations. Facial feature extraction aims to extract features from the input pattern that can be used to classify the corresponding facial expression type. In facial expression classification stage, the classifier categorises the input image pattern into its corresponding facial expression based on the extracted facial features.

Figure 2.1: Major stages in a typical automatic FER system.

Existing approaches for FER differ mainly in the way features are extracted from an image or a video sequence. Facial feature extraction is an important step in FER systems, since salient features can improve the system performance. An advanced feature extraction method generally meets the following requirements:

- It extracts representative facial features that can be used to differentiate facial expressions;
- It removes interference caused by noise, illumination variations and other environmental factors;
- It reduces the dimension of image patterns.

Figure 2.2 illustrates different types of facial feature extraction techniques. In the following sections we review four main approaches for feature extraction:

- geometric feature-based approach (Section 2.1);
2.1. Geometric-based approach

In geometric-based approaches, faces are represented geometrically via fiducial points or the shape of facial regions. Classification is achieved by analysing the distances between feature points and the relative sizes of the facial components. Pantic and Rothkrantz [1] proposed a method for detecting facial actions by analysing the profile-contours and the contours of facial components, such as the eyes and the mouth. They used an approach with multi-detectors to spatially sample the contours. Each detector is a known feature filter which is simple and easy to implement. To locate six regions of interest (ROIs) in the entire face, namely, two eyebrows, two eyes, nose, and mouth (see Fig. 2.3), the multi-detectors are either a single filter, which detects all facial features in a face region, or a set of different filters, which are all assigned to the same face region. For example, two different filters are utilised to locate the contours of the eyebrows

---

Figure 2.2: Existing techniques and procedures of facial feature extraction.

- appearance feature-based approach (Section 2.2);
- fusion feature-based approach (Section 2.3);
- spatio-temporal feature-based approach (Section 2.4).

2.1 Geometric-based approach
2.1. Geometric-based approach

in the eyebrow ROIs, and the contours of the eyes are depicted in the eye ROIs by a single filter.

![Image of geometric feature points and contours](image)

**Figure 2.3**: Geometric feature points (fiducial points of the contours of the face) [1].

Each extracted point is assigned with a certainty factor (CF). To generate CFs, the first step is to manually mark the fiducial points on an image with neutral expression, which can be found at the beginning of each video sequence. Then the distance between two points, namely, the fiducial point detected on the current image, \( B_{\text{current}} \) and the corresponding point on the original neutral expression image, \( B_{\text{neutral}} \), is calculated. Finally, the pertinent \( CF_B \) can be generated as:

\[
CF_B = \text{sigm}(d(B_{\text{current}}, B_{\text{neutral}}), 7, 3.5),
\]

where \( d(B_{\text{current}}, B_{\text{neutral}}) \) is the distance between points measured in pixels on the current and neutral images, and \( \text{sigm}(\chi, \mu, \sigma) \) is a Sigmoid function given as

\[
\text{sigm}(\chi; \mu; \sigma) = \frac{1}{1 + \exp \left[ (\chi - \mu)/\sigma \right]}
\]

Geometric feature-based approaches can cope well with variations in skin patterns or dermatoglyphics. However, they require highly accurate detection of facial fiducial points, which is difficult when the image has a low-quality or
a complex background. These approaches also have difficulty in differentiating subtle facial expressions.

2.2 Appearance-based approach

Geometric-based approaches extract salient facial points to infer the shape and location of facial components. However, facial features also exhibit appearance changes such as wrinkles and furrows that are important cues for FER. These appearance changes are difficult to capture using geometric-based approaches. To deal with this problem, appearance-based approaches try to use all image pixels. Typical appearance-based approaches utilise image filters, such as Gabor filters or linear filters, to extract feature vectors, namely, gradient, correlation, or texture, from the whole face or specific regions. For example, because the texture variations are quite stable to illumination changes, they are widely used as the cues for FER and can be detected by the Gabor filter [15, 16, 17, 18, 19].

Zhen and Huang [19] proposed a ratio-image based appearance feature extraction method using Gabor filters. They used $R(u, v)$ to express the ratio-image as a texture plane, which was independent of the face albedos. Here $(u, v)$ are the two axes of the texture plane. The compact appearance features are extracted from $R(u, v)$ using the high frequency components in the frequency domain. Their research indicates that, instead of extracting features at certain points, it is more efficient to extract texture features in eleven facial regions where appearance changes are most likely to occur. Features at each region are extracted using Gabor wavelets as a set of multi-scale and multi-orientation coefficients. Two spatial frequency scales with wavelengths of 5 and 8 pixels are used, and there are 6
2.2. Appearance-based approach

orientations for each scale. Based on weighted average of each facial region, the final feature values are computed and then used to generate eleven appearance feature vectors corresponding to each of the facial regions. This approach can cope with different people and illumination conditions.

Feng [20] used Local Binary Patterns (LBP) to extract facial texture features and combined different local histograms to recover the shape of the face. A basic LBP operator is used to detect local texture primitives and label each pixel of an image with an LBP code. The labelled face image is then divided into several regions; while LBP histograms are calculated to represent texture information of each facial region. A single vector is then concatenated from different LBP histograms to recover the shape of the face. A coarse-to-fine scheme for classification is proposed where seven templates are formed for the seven basic facial expressions. Firstly, two expression classes are selected according to the distances from the test image to the seven templates. The final facial expression is then determined via a K-nearest neighbour classifier with weighted Chi-square statistic. This approach is capable of representing facial expressions in terms of visible appearance changes, such as shape and textural structure, which indicate different facial expressions.

Shinohara and Otsuf [21] proposed a hybrid method by combining Higher-order Local Auto-Correlation (HLAC) features and Fisher weight maps to quantify the importance of each facial area and extract effective features for classification. Donato et al. [17] compared a number of different image analysis methods. Results showed that the one based on local Gabor wavelet decomposition and the Independent Component Analysis (ICA) had the best performance for classifying 12 facial actions of the upper and lower face, comparing to Principal Component
Other appearance-based approaches include: local facial asymmetry based method [22] and moment invariants method [2]. In [22], Mitra and Yanxi defined density difference, D-face, and edge orientation similarity, S-face, as facial asymmetries of a normalised face image. These two face features are called AsymmetryFaces. Experimental results show that facial asymmetry features can perform well in expression recognition tasks. Zhu et al. [2] defined seven important areas (see Fig. 2.4) on the face for feature extraction. They modified the formula of moment invariants by replacing central moments with ordinary moments to track the relative position changes of facial features. These position changes reflected the deformation of facial muscle actions. After calculating the feature vector in each area using modified moment invariants, Zhu et al. divided each element of the feature vector by a scaling factor, $\alpha$. By adjusting the value of $\alpha$, they could change the strength of the feature vectors to improve the classification accuracy.

Figure 2.4: Seven facial areas for feature extraction [2].

### 2.3 Fusion feature-based approach

Geometric-based methods capture macro-variations of facial structures while appearance-based approaches are capable of identifying local subtle changes.
By combining these two, the performance of FER system can be improved. Zhang and Ji [3] proposed a multi-feature technique for FER. This method is based on the detection of facial points, nasolabial folds, and edges in the forehead area. The last two are both used for detecting transient variations of the facial feature (see Fig. 2.5). To ensure the accuracy of feature measurement, this approach chooses two fiducial points at eye inner canthi (F and $F'$) as a reference to measure locations of other feature points. The rectangles in Fig. 2.5 highlight several transient wrinkles and furrows caused by facial muscle movements in the appearance of certain face regions. These appearance features provide additional supporting cues to make the recognition decision. The facial features are extracted by associating each AU with a set of movements, and then classified using a Bayesian Network model.

![Figure 2.5: The geometrical relationships of facial feature points, where the rectangles represent the regions of furrows and wrinkles [3].](image)

Active Appearance Model (AAM) [23] has been widely used for facial feature extraction [6, 24, 25, 26]. It combines appearance and texture information to construct a parametrised model of facial features. Ya et al. [27] applied Active Wavelet Network (AWN) to align the face image, rather than using PCA-based texture model such as AAM. AWN employs a Gabor wavelet network representation (GWN) to model the texture variation in the training set. The GWN
is robust to partial obstruction and changes of lighting conditions. Compared to appearance-based methods, methods using models generally obtain more reliably facial features. Nevertheless, they may have disadvantages such as calculation complexity, and difficulty in acquiring initial points.

The techniques presented in this section extract facial features mainly from static images. In the next section, we review the methods extracting spatio-temporal features from image sequences.

## 2.4 Spatio-temporal features

The spatio-temporal features are extracted to detect movements of the facial components, which reflect certain facial expressions. There are three general approaches for extracting spatio-temporal features, namely, feature point tracking, model-based tracking, and optical flow.

### 2.4.1 Feature point tracking based approach

Feature point tracking based method chooses the facial points, such as the corner of the eye or the corner of the mouth, of which the grey-scale values have considerable variations. Around these facial points, we can generate information about motion and deformation of facial features, because they are easily to be tracked. Bourel et al. [28] proposed a method that used a robust tracker to detect 12 manually defined facial feature points (see Fig. 2.6). This facial point tracker is based on the Kanade-Lucas-Tomasi (KLT) tracker [29]. It detects and recovers the 12 facial points if any of them is lost due to illumination variation and rigid or non-rigid motions. After tracking the facial features, a spatially-localised geometric facial
2.4. Spatio-temporal features

Figure 2.6: 12 facial feature points to be tracked.

The model is generated to represent the expression. This model is created by taking the differences between the geometric coefficients of feature points of the first frame and the current frame using Euclidean distance. This approach utilises a scalar quantisation of the temporal evolution of geometric facial features. Experimental results demonstrate that the performance of this FER method remains stable in the context of partially facial occlusion, or a variety of levels of noise introduced at the feature tracking level.

Zhang and Ji [3] developed a technique based on IR illumination camera system and Kalman filter to detect pupils. Given the detected pupil positions, the initial regions of eyes, nose, and mouth are located based on the anthropometric statistics, which provide the spatial relationships between facial regions. By utilising the IR device, this method can provide good approximations for the future positions of feature points, even with significant head movement involved.

2.4.2 Model-based approach

The model-based approach uses models to track human faces. Facial motions are recognised based on the model parameters and the inter-frame information. In contrast to feature tracking that focuses on specific facial features, model-based
tracking aims to track the entire face. Face models can be 2-dimensional or 3-dimensional, and most of the models require complex calculation.

Huang et al. [4] used a multi-resolution 3-D deformable face model and a hierarchical tracking scheme to track intra-frame changes that caused by highly local skin deformations. The first level of the system hierarchy is a coarse-level mesh with 1000 nodes and the second level is a fine-level mesh with 16000 nodes. An overview of the system face model is provided in Fig. 2.7. The 1000-node face model is used to track 3-D dynamic range deformations, which is known as global tracking or coarse level tracking. The authors then use a non-rigid 3-D shape registration algorithm at a higher level to register the subdivided 16000 nodes mesh to the frame based on the global tracking result. This local registration detects local subtle deformation and combines with global tracking to provide refined fitting to the 3-D face image. The results show that this approach can efficiently parametrise a large amount of data by dealing with large-scale deformations in the facial expressions. Hence, subtle expression details are accurately captured and tracked for motion analysis and expression recognition.

![Figure 2.7: (a) The 3-D face model with manually selected feature points. (b) The face model are fitting with the face scan data [4].](image)

Tao and Huang [30] proposed a motion tracking algorithm based on a piece-
wise Bazier volume deformation (PBVD) model. Feature points, such as corners of the eyes and mouth, are extracted manually in the initial frame. A generic 3D model is then adjusted to fit the selected feature points, and a set of predefined action units are used to express motions of several facial areas. The tracking result represents the motions of the face, as well as the direction and intensity of the motions. This model and action units have also been used in other research methods [19, 31, 32].

Braathen et al. [33] used a 3-D face model to track out-of-plane head rotations which appear in spontaneous face images. The faces are tracked by a particle filter with 100 particles tracks, and then warped onto 3-D face models to get the canonical views. The warped images are rotated to frontal views of the faces to reduce the complexity in recognition. Gokturk et al. [34] developed a system used a 3-D deformable model-based tracker to detect facial poses and deformations in every frame of a monocular video sequence. A shape vector is extracted and considered as robust facial features for classification. In addition, the discrepancies of the shape vectors within each frame are also considered as part of the facial features. Results show that this method is adequate for recognising facial expressions with large rotational and translational head movements.

The model-based tracking approaches provide detailed matching between face models and face data. However, they may have restrictions in applications, such as the requirement of manually marking of the facial feature points and the complexity in face alignment and calculation increased by large number of marked feature points.
2.4.3 Optical flow based approach

Optical flow is used in the velocity field which wraps one image into another. Figure 2.8 illustrates an example of an optical flow application [5]. Here a rotating object is used to observe the optic flow. Each arrow indicates the direction and magnitude of optic flow at each location. Optical flow reflects essential information between frames, therefore, it has been adopted in many dynamic facial expression recognition systems [35, 36, 37, 38]. Keith and Peter [38] used a face tracker, which is based on an improved optical flow algorithm, and SVM classifiers to recognise facial expressions. The optical flow algorithm is designed to generate motion information of the face region. The results show that this system can operate in real-time, and the recognition rate is comparable to other techniques that use a basic optical flow. The disadvantage of optical flow based approaches lies in that non-homogeneous illumination and facial non-rigid motion may influence the effect of feature extraction. To cope with this problem, Yacoob and Davis [35] proposed an approach to analyse and classify facial expressions based on optical flow that tracks principal regions of the face, and integrates spatial and temporal information in the tracking algorithm.

![Diagram of optical flow](image)

Figure 2.8: Example of optic flow experienced by a rotating observer [5].

Section 2.3 and 2.4 outline techniques that generate primitive features from
static images and image sequences. However, it is possible that these extracted features contain redundant information which increases the difficulty of FER. The following section presents techniques that transform the full size of input features into a reduced representation set of features.

2.5 Feature selection and dimensionality reduction

To improve the efficiency of FER systems, several methods have been developed to generate informative features and reduce the computational costs. This section reveals some of the feature selection and dimensionality reduction methods.

Because the number of extracted primitive features generally is very large, especially using appearance-based methods, it is necessary to select efficient features and reduce the feature dimensions. Feature selection removes irrelevant and redundant features to build robust training models and improve system performance.

There are two general strategies for feature selection. One is filter, such as information gain. Information gain is known as Kullback-Leibler divergence, which measures the difference between two probability distributions \( p \) and \( q \). The following equation defines the K-L divergence from probability distributions \( p \) to \( q \):

\[
D_{KL}(p||q) = \int_{-\infty}^{\infty} p(x) \log \frac{p(x)}{q(x)} \, dx,
\]

where \( p \) and \( q \) indicate the probability densities. It is proved that \( D_{KL}(p||q) \) is non-negative, \( D_{KL}(p||q) \geq 0 \), and \( D_{KL}(p||q) = 0 \) if and only if \( p = q \). Features that best separate different facial expressions typically produce the largest K-L divergence.

The other strategy is wrapper, which utilises an algorithm to search for the
2.5. Feature selection and dimensionality reduction

optimal feature among all the available features. Stepwise regression, a greedy algorithm, is a common form of feature selection using wrapper strategy. At each round, the best feature is selected based on the classification accuracy. Cross-validation is used to choose the stop criteria, but it can still cause over-fitting to the system. Wrappers may have large computational expense.

Feature selection can be categorised into two types: feature ranking and subset selection. Feature ranking categorises features by a metric, and the features fail to meet the required score are eliminated from the feature pool. Subset selection evaluates a possible subset of features for the optimal suitability. A common subset searching algorithm is called greedy hill climbing. At each evaluation round, this approach modifies the current subset to generate a new subset and compares the two subsets. If the modified subset achieves a better performance, it will be reserved as initial feature subset at next round. Otherwise, the current subset is modified until an improved subset is found. To search for new features, there are several searching algorithms, such as exhaustive search, best first search, and greedy forward selection. Exhaustive search typically consumes massive resources and time, therefore it is necessary to introduce stop criteria. A possible approach is to stop searching when a subset score exceeds a threshold, or the program’s run time limit has been reached.

One of the popular filter metrics for feature selection is mutual information which selects the features with highest relevance to the target class [39]. To generate mutual information between two random values \( x \) and \( y \), we use the
2.5. Feature selection and dimensionality reduction

following equation,

\[ I(x, y) = \int \int p(x, y) \log \frac{p(x, y)}{p(x)p(y)} \, dx \, dy. \]  \hspace{1cm} (2.4)

where \( p(x) \) and \( p(y) \) are the probability density functions (PDF) of \( x \) and \( y \), respectively, and \( p(x, y) \) is their joint-PDF.

Commonly used methods for dimensionality reduction are Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), Independent Component Analysis (ICA). PCA is a multivariate analysis and aims at finding a small number of factors that can express most of the variations in a large number of variables. It transforms the data to a new coordinate system such that the greatest variance of any projection of the data is along the first coordinate (called the first principal component), the second greatest variance is along the second coordinate, and so on [40]. However, the PCA may fail to distinguish the classes. LDA is used to find the linear combination of features which best models the separation between different classes of data. This method places emphasis on seeking for the direction provided with the maximum resolution power.

Dubuisson et al. [41] proposed a feature selection method that sorted the principal components, generated by PCA, in the order of importance according to their eigenvalues. Then a forward stepwise selection approach was introduced to select \( K \) components. Finally the authors used LDA to process the sorted eigenspace and yielded a \((c-1)\)-dimensional subspace for classification, where \( c \) is the number of classes. This method is called Sorted PCA plus LDA and it establishes optimal subspaces for different recognition tasks. Aalewski [24] introduced a method called mixture of probabilistic PCA (MP-PCA). In contrast to the tradi-
Facial feature decomposition, the MP-PCA defines a probabilistic model for the traditional PCA. The information that is not located along the principal axes is estimated as Gaussian noise, instead of being directly abandoned. The optimal statistical model is generated using a maximum likelihood solution method that generates model parameters. This approach solves non-linear distributions in feature space caused by large pose variation, and recognises facial expression together with the Bayesian network. Chen and Huang [42] developed a novel discriminant feature extraction method for FER, named the clustering based discriminant analysis (CDA). This method is modified from LDA and aims to solve the problem when multiple clusters exist in a class due to variations in illumination. A fuzzy c-means algorithm is used for clustering. This CDA method exploits cluster information and separates clusters associated with different classes. Their experimental results indicate that this method performs well when there are multiple clusters in one class.

2.6 Facial feature decomposition

Facial images contain abundant information, and the importance of the information varies with the types of recognition tasks. Face detection seeks for mutual coherence among facial images; and face recognition needs information of individual differences; while FER emphasises the variations between different types of expressions. Information which is in favour of certain recognition task may interfere other types of pattern recognition tasks. A novel way to solve this inconvenience is to separate different facial factors, such as expressional factor and individual factor. Thus with the selected facial factors, recognition tasks can be
conducted in corresponding spaces to avoid interference.

Wang and Ahuja [6] proposed a novel approach using Higher-Order Singular Value Decomposition (HOSVD) for facial expression decomposition. This approach analyses multiple feature factors and is utilised for both face and facial expression recognition. The authors use three orders of tensor to express different persons and facial expressions. These three orders stand for number of persons, number of facial expressions for each person, and the dimension of the facial feature vector (extracted using AAM), respectively. By using HOSVD method to decompose the tensors, they are able to generate different subspaces associated with the person and expression mode for facial expression synthesis and recognition (see Fig. 2.9). The person and expression subspace models are derived from training images with seven basic facial expression image samples. The facial expression subspace model captures the signatures of expressions correctly. Because of its speciality to decompose multiple feature factors of human face, this method can be implemented in FER systems and facial synthesis tasks. By adding new feature factors, this method can even cope with illumination and pose changes.

Feature decomposition based methods need to analyse the entire image database during the decomposition process. Wang and Ahuja [6] hypothesised that the person vector corresponding to the test samples is denoted by the first training sample. They generate the similarity (such as cosine distance or Euclidean distance) of facial feature vectors by decomposing the training and test samples. All of the training samples in the database need to go through this process and finally be categorised to the nearest facial expression type.
2.7 Classification

Facial expression classification is the last stage in a FER system. Based on the extracted facial features, the classifiers “identify” the facial expression types of the input patterns. Facial expressions can be categorised in various ways. For example, they can be categorised based on Facial Action Coding System (FACS), which utilises the deformation in facial muscles, or non-prototypic expressions such as “raised brows”, or prototypic expressions such as emotional expressions [43]. Contractions of the facial muscles cause independent subtle changes in facial appearance and are able to be detected by human observers using FACS. A trained human AU-coder categorises an input image pattern into specific AUs associated with muscle changes and corresponding facial expressions. There are many FACS-based automatic FER approaches; however, applying the automated facial action coding methods to automatic AU-encoding still requires further development [17, 43].

Most FER approaches are based on emotional classification. In other words,
the classification focuses on the existence of "universal categories of emotional expressions". As described in [44], Ekman defined six expression categories that are widely used as prototype expressions in conducting expression classification. However, most FER experiments also consider neutral as the seventh basic expression prototype. Many classifiers for facial expression classification have been developed and applied to different FER systems. The subsequent sections reveal some of the typical classification approaches and compare their performances.

2.7.1 Rule-based classifier

This classifier uses decision rules based on human observations. It determines the facial expression rules that the extracted features belong to and relegates the facial image to the corresponding expression category. Pantic and Rothkrantz [1, 43] utilised AU-coded description to represent the input expressions. The classifier then sorts the input pattern to the nearest facial expression. Matsugu and Haneda [45, 46] also used a rule-based algorithm for FER. Their rule-based algorithm utilises the differences of specific local facial areas between neutral and other expressional face images. For example, the rules used for detecting smiling can be summarised as follows:

- The distance between endpoints of eye and mouth gets shorter (lip being raised);
- The length of horizontal line segment in mouth gets longer (lip being stretched);
- The length of line segments in eye gets longer (wrinkle around the tail of eye gets longer);
2.7. Classification

- The gradient of line segment connecting the midpoint and endpoint of mouth gets steeper (lip being raised);

- The number of edges around mouth increases (teeth get appeared);

- The number of edges in cheeks increases (wrinkle around cheeks gets grown).

Experimental results indicate that the rule-based classifier depends less on individual characteristics for FER. These approaches are able to accurately classify human facial expressions.

2.7.2 Artificial neural network

Artificial neural network (ANN) is commonly used in FER systems to classify static images [18, 47, 48, 49]. Gueorguieva et al. [49] used a multi-layer perceptron to perform FER. They develop and test four networks and conclude that a combination of sigmoidal and radial basis functions hidden neurons is better suited in the feed-forward neural network. Ma and Khorasani [47] used the two-dimensional discrete cosine transform (2-D DCT) over the entire face as a feature detector. For facial expression classification, a constructive feed-forward neural network with one hidden layer is applied as the classifier. The results indicate that this technique yields a higher recognition rate than other neural networks. Disadvantage of neural network based FER is that classifier training becomes difficult when recognising fusion facial features.
2.7. Classification

2.7.3 Bayesian classifier

Sebe et al. [50] introduced a Cauchy Naive Bayes classifier. Their experimental results prove that the Cauchy distribution assumption is better than Gaussian distribution assumption in terms of classification performance. This classifier is based on the assumption that the features are mutually independent. Nevertheless, when recognising different facial expressions, the features are not fully independent, but interactively related. This is a disadvantage of Cauchy Naive Bayes classifier.

Cohen et al. [31] used the Naive-Bayes classifiers which are combined with the Tree-Augmented-Naive Bayes (TAN) classifiers in the FER system. They utilised unlabelled image sets to train the Bayesian network classifier. To improve the classification performance, labelled training sets and large amount of unlabelled data sets are combined to train the classifiers [31]. Zhang and Ji [3] used a three-layer Bayesian network (BN) model to represent the causal relationships between facial expressions and facial AUs. The lowest level of layer in the model is the sensory data layer containing visual information variables, such as brows, lips, lip corners, eyelids, cheeks, chine, mouth, nasolabial furrow and wrinkles. The middle layer is facial AU layer and the upper layer is classification layer. This BN model provides the best representation of facial expressions for static face images. Combining with Hidden Markov model (HMM), it performs the spatio-temporal analysis for facial expressions.
2.7.4 AdaBoost

AdaBoost [51] is a learning algorithm that selects a small number of classifiers from a large number of weak classifiers or a hypothesis space to construct a strong classifier. The AdaBoost algorithm has been successfully used in face detection [52] and facial expression classification [53]. Wang et al. [53] used the continuous multi-class AdaBoost algorithm to train several weak classifiers for his FER system. Experimental results show that the boosting method outperforms SVM classifiers, especially in speed.

Guo [54] proposed a method based on linear programming for both feature selection and classifier training. This Feature Selection Linear Programming (FSLP) algorithm can select features and train the classifier simultaneously without using heuristics. Experimental results show that the FSLP algorithm copes well with small sample recognition problems.

2.7.5 Support vector machine

Support Vector Machine (SVM), developed from statistical learning theory, is a widely used classifier for facial expression classification [55, 56, 57, 58, 59]. It has several advantages in solving small sample size, non-linear, or high dimensional classification problems. Bartlett et al. [56] developed a technique using AdaBoost to choose a subset of features extracted from Gabor filters. The SVMs are then employed to perform classification. Their system achieves 93% recognition accuracy in recognising seven facial expressions on their own database, which is higher than other classification methods tested on the same database.
2.7. Classification

2.7.6 Hidden Markov model

A commonly used classifier for spatio-temporal applications is the Hidden Markov Model (HMM) [31, 33, 60, 61, 62, 63, 64]. Muller et al. [61] employed pseudo 3-D HMM (P3-DHMMs) for dynamic facial expression classification. This method uses a pseudo 2-D HMM to model each facial image, and then uses the conventional 1-D HMM to model the super-state that denotes the temporal information. Yeasin et al. [62] introduced an approach to recognise the six universal facial expressions from visual data. Their approach relies on a two-step classification strategy which is built on top of refined optical flow that is computed from sequential images. A set of linear classifiers is firstly applied at frame level and the output is integrated to produce a temporal signature. The next stage is to use the temporal signatures obtained from the training data set to train discrete HMMs.

The HMM classification approach needs to identify the start and stop states. Therefore, it is generally applied to image sequences or sorted expression sequences. Cohen et al. [31] proposed a multi-level HMM. The first level consists of independent HMMs that are related to the six basic facial expressions, and the second layer is the high-level Markov model composed of seven states that represent the changes of facial expressions. The motion features are continuously used as input to the six emotion-specific HMMs. The state sequence of each HMM is decoded and used as the observation vector for the high-level Markov model. The state transition probability of the six facial expressions is derived via model training. Thereby, the videos are automatically separated to different facial expression segments for recognition.
2.8 Performance evaluation

There are three major databases for FER system evaluation.

- The Japanese ATR Female Facial Expression (JAFFE) database \cite{16} is a widely used database for FER system evaluation. This database composes of 7 basic facial expressions, including 10 Japanese females, and has approximately 213 images. Each actress is directed to perform two to three poses for each facial expression.

- Another common database for FER evaluation is the Cohn-Kanade database \cite{65} of CMU. It is an AU-coded face expression image database which contains 2105 digitised image sequences of 210 subjects of various ethnicity and age.

- Pantic et al. \cite{66} established a web-based database, MMI, for facial expression analysis. This database contains more than 2000 video sequences that demonstrate not only various facial expressions, but also single and multiple facial muscle movements. Because the MMI database was recently released in 2005, it has just started becoming popular in FER system evaluation.

Besides these three databases, there are other image databases that have been applied to FER system evaluation, such as the Yale database, and PIE database of CMU. Table 1 summarises the experiments implemented on Cohn-Kanade database and JAFFE database, respectively. This table includes the information about the recognition tasks (AUs or basic expressions), data types (static, sequence, or frames out of sequences), data sizes, methodologies, and their classification results.
2.9. Chapter summary

A significant number of researches have been conducted for automatic FER. However, the following weaknesses still prevent the practical applications of existing FER systems:

- The feature extraction methods are not sufficiently robust. Model-based
methods require manual identification of facial points or areas. Appearance-based methods can be implemented automatically, but the extracted features are unreliable and sensitive to noise.

- The recognition rate varies depending on the type of facial expressions. While it is easy to differentiate happiness and surprise expressions, it is difficult to compare fear versus sadness. Other expressions such as suspicion or commiseration are even more difficult to recognise.

- Most existing FER techniques that are designed and evaluated on images or video sequences are collected under constrained conditions. A practical FER system is required to operate on a wider range of conditions.

The key research directions in FER can be listed as follows.

- A change in facial expression will produce several changes across the facial image, from the location of fiducial points and face shape to skin texture and shadows. Facial expressions can be identified more accurately through fusion of facial features.

- Appearance-based methods are widely used for FER but these methods produce a large number of features, some of which are not relevant or salient for recognition. Feature selection and dimensionality reduction will play an essential role in FER research.

- Machine learning has been used for recognition and classification of facial expression with reasonable results. However, human facial expressions follow certain psychological rules. FER can be improved by combining
2.9. Chapter summary

psychological and biological knowledge with machine learning.

- In existing FER systems, facial expressions to be recognised are generally exaggerated expressions. However, in practice, the facial expression of a person may be a hybrid facial expression. For example, at certain level between neutral and happiness. FER research can put more emphasis on finding the degree of a facial expression.

- Facial expression recognition are affected by many factors, such as lighting conditions, pose changes, and partial occlusions of the face. There are some preliminary works addressing the problem of partially occluded faces in FER systems [3, 67], but recognition accuracy can still be improved. The 3-D modelling is commonly employed to deal with illumination and posture changes, but 3-D face modelling technology is not yet matured. Feature decomposition is another solution to these problems, but it still requires further development.
Face Detection and Alignment

Chapter contents

3.1 Preliminary face detection ................................................. 38
   3.1.1 Statistical-based approaches ..................................... 38
   3.1.2 Knowledge modelling based approaches ......................... 39
   3.1.3 Face detection based on Haar-like features ....................... 40
3.2 Image preprocessing for face alignment ............................... 42
3.3 Horizontal non-linear filter ............................................. 44
3.4 Eye filters ............................................................... 46
3.5 Extraction of eye candidate regions .................................... 48
3.6 Geometric face model and template matching ......................... 48
3.7 Chapter summary ......................................................... 51

Face detection and alignment is the foremost step of an automatic FER system. It can be considered as locating the human faces from the input images and providing upright face patterns for FER system. This chapter discusses general face detection approaches and outlines the proposed face detection and alignment method.
3.1 Preliminary face detection

A face detector must be able to precisely locate faces in complex background and at different face angles. The basic idea of face detection is to use knowledge or statistical-based face modelling, which compares the input image pattern with a face model. The existing approaches for face detection can be broadly divided into two categories:

- **Statistical-based approaches** - which consider facial images as high-dimensional vectors, and convert the face detection problem to the problem of signal detection in high-dimensional spaces.

- **Knowledge modelling based approaches** - which set a number of rules based on human knowledge, thereby, convert the face detection problem to the problem of hypothesis and test.

The following sections present detailed descriptions of these approaches.

3.1.1 Statistical-based approaches

- **Example learning**: Converting face detection and alignment problems to a pattern classification problem of separating faces from none-face patterns. The categorizer can be generated by learning from face patterns and none-face patterns. The ANN is a typical example learning approach which has been widely used for face detection in recent years [68, 69, 70].

- **Model learning**: There are two kinds of models, namely, intransigent model and deformable model. The former aims to measure certain distance between test pattern and the reference model. Face detection depends on the
threshold value. This method is used primarily in image preprocessing for
face detection. The deformable model contains non-fixed variables, and
penalty functions. It uses parametrised or adaptive curves and curved faces
to construct facial models.

3.1.2 Knowledge modelling based approaches

- **Face rules**: Face rules are the universal spatial correlations between facial
  features. They include:

  - *Gray level distribution rule*. This rule indicates that under same illumi-
    nation circumstances, the grey level of eyes is always lower than that
    of forehead and cheek-bones.

  - *Contour rule*. It treats the contour of the face as an oval, therefore,
    simply detect faces by detecting ovals in the images.

  - *Motion rule*. Generally, human faces are moving relative to the back-
    ground. Thus, based on the motion information, we can effectively
    extract faces from arbitrary complex backgrounds.

- **Colour and texture information**: It is most likely to find concentrated dis-
  tribution in the colour spatial of facial colours where people are from the
  same ethnicity. Face colour, to a certain extent, can differentiate face from
  most backgrounds.

- **Symmetry**: Human face has certain symmetry characteristic while facial
  components are also symmetrical. Reisfeld [71] used a generalised sym-
  metrical transformation to locate facial components. This method detects
3.1. Preliminary face detection

high-order spatial relationships among the facial pixels which contain strong symmetrical characteristics.

3.1.3 Face detection based on Haar-like features

The rectangle features, also known as Haar-like features, were first been introduced by Papageorgiou [72] in 1998 for detecting human faces. Based on these rectangle features, Viola and Jones proposed a real-time object detection algorithm which has been widely used in face detection and other pattern recognition applications [7, 73, 74]. This algorithm is designed to detect faces based on the geometric feature values of facial objects, such as the eyes, nose, and mouth.

The rectangle features have been commonly applied to geometric texture detection. There are several types of rectangle feature and each of them is considered as a sub-window. Accordingly, each sub-window is designed to detect specific types of geometric textures. Viola and Jones [7] used four types of rectangle features, which are shown in Fig. 3.1, to detect human faces. Figure 3.1a and 3.1b have similar appearances, however, they are used to detect vertical and horizontal geometric boundaries, respectively. These boundaries indicate the edges of certain object in the original image and separate the object from its background. The rectangle feature, in Fig. 3.1c, is designed to detect geometric features that have complex texture in the middle and smooth texture on both sides. The four-rectangle feature shown in Fig. 3.1d is created to estimate the difference between diagonal pairs of rectangles [7].

The rectangle features consist mainly of black and white rectangles which are used to calculate the corresponding feature values. Figure 3.2 provides an
3.1. Preliminary face detection

Figure 3.1: Four basic types of rectangle features [7]: (a) and (b) are two-rectangle features, (c) is a three-rectangle feature, and (d) is a four-rectangle feature.

Example of computing the value of the rectangle feature shown in Fig 3.1d:

$$V = (A + D) - (B + C),$$  \hspace{1cm} (3.1)

where $V$ is the rectangle feature value, $A, B, C,$ and $D$ are the sums of pixel values covered by the four small rectangles.

Figure 3.2: An example of computing the rectangle feature value. Note that each small rectangle indicates a pixel value.

Each rectangle feature is a sub-window that can be placed at different location in the input image, and its size can also be changed. Based on this property, Viola et al. [74] consider each rectangle feature of different parameters (such as location and size) as a weak classifier. Therefore, a large number of weak classifiers are generated using different rectangle features. However, not all the classifiers are beneficial in human face detection. For example, in terms of locations, rectangle
features at the eye area contribute more than those at the cheek area. It is because rectangle features are mainly used in classifying texture information, and the eye area which has more complex textures presents better evidence for face detection.

To reduce the computational cost and improve system efficiency, it is important to extract and combine a collection of more distinctive weak classifiers to form a stronger classifier for face detection. A learning algorithm called AdaBoost is employed to select the efficient features and train the classifiers [74]. In this project, we use the OpenCV AdaBoost-based method to detect faces. An example of the face detection result is shown in Fig. 3.3.

![Figure 3.3: An example of OpenCV AdaBoost-based face detection.](image)

### 3.2 Image preprocessing for face alignment

Many recognition algorithms depend on careful positioning of an object into a canonical pose, so the position of features relative to a fixed coordinate system can be examined. Generally, this positioning is done either manually or by training a class-specialised algorithm with samples of the hand-labelled parts or poses.
3.2. Image preprocessing for face alignment

Due to the amount of supervision required by these methods, the alignment step of the recognition pipeline (described in Chapter 1) is often ignored, under the assumption that the initial detection will perform rough alignment.

Even with the state-of-the-art face detectors, the boundary for the facial region is sometimes incorrectly calculated. The localisation error is more common when there is rotation (in-plane or out-of-plane rotation). To improve accuracy in facial image analysis tasks such as face, gender and facial expression recognition, it is essential to correctly align the face images. We propose a face alignment method which consists of two main steps. First, the possible eye candidates are detected using a combination of a Gabor filter and a circular filter. Second, for each candidate eye pair, two face candidates are constructed using a geometric face model, and each face candidate is compared with a face template to remove false detection.

Figure 3.4: An example result of the OpenCV face detector. Even the face is found, its coordinates and rotation angle can be better estimated.

For eye detection, several image preprocessing steps are performed to reduce the effect of the lighting condition. First, the image is histogram equalised. Second, a normalised image is calculated as follows

$$
N(x, y) = \frac{I(x, y)}{\mu(x, y)}
$$

(3.2)
where $I(x, y)$ is an image pixel, $N(x, y)$ is the normalised pixel, and $\mu(x, y)$ is the mean intensity of the neighbouring pixels (3-by-3). Third, a contrast-normalisation is applied

$$C_N(x, y) = \frac{I(x, y) + N(x, y)}{I(x, y) + N(x, y) + \delta},$$

(3.3)

where $\delta$ is the mean pixel intensity of image $N$ and the division is performed pixel-wise. The image preprocessing steps are illustrated in Fig. 3.5.

Figure 3.5: Examples of image preprocessing. (a) input image. (b) histogram equalization. (c) illumination normalisation. (d) contrast-normalisation.

The candidate eye regions are detected, based on the elongated shape of the eye and the circular shape the the pupil. We will use two approaches to process the face image. Firstly, we proposed a horizontal non-linear filter to extract elongated regions on the face. Secondly, an eye filter combining a Gabor filter and a circular filter is presented to analyse possible eye region candidates.

### 3.3 Horizontal non-linear filter

The horizontal non-linear filter is used to extract primitive features such as horizontal regions and edges that are present in a wide range of objects. This filter is designed based on the biological mechanism called *shunting inhibition*. It is first presented in [75] which is as follows

$$h(x, y) = \frac{D \ast I}{G \ast I},$$

(3.4)
3.3. Horizontal non-linear filter

Figure 3.6: Steps for finding eye candidates using the horizontal non-linear filter: (a) output of the non-linear filter, (b) connected component labelling, (c) image erosion and dilation, (d) candidate eye points (shown in red colour).

where $I$ is a 2-D input face pattern, $D$ and $G$ are the filter coefficients, “$*$” denotes 2-D convolution, and the division is done pixel-wise. The kernel $G$ is chosen as an isotropic Gaussian kernel:

$$G(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right). \quad (3.5)$$

To extract the horizontal elementary facial features, the kernel $D$ is formulated as the second-order derivative Gaussian of $x$. Its coefficients are defined as

$$D(x, y) = G_x''(x, y), \quad (3.6)$$

where

$$G_x''(x, y) = \left(\frac{x^2 - \sigma^2}{2\pi\sigma^6}\right) \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right), \quad (3.7)$$
3.4. Eye filters

Figure 3.6 shows the outputs of the processing stages using the horizontal non-linear filter.

3.4 Eye filters

The eye filter is used to extract possible eye regions based on the elongated shape of the eye and the circular shape the pupil. The circular filter is first proposed by Park et al. [76].

- Gabor filter

The Gabor filter, as shown in Fig. 3.7a, is the product of a harmonic function and a Gaussian function. Real part of a 2-D Gabor filter is defined as

\[
h(x, y) = \frac{K_v^2}{\sigma} \exp\left(-\frac{K_v^2(x^2 + y^2)}{2\sigma}\right) \times \left[\cos(x \cdot K_v \cos Q_u + y \cdot K_v \sin Q_u) - \exp\left(-\frac{\sigma}{2}\right)\right],
\]

(3.8)

where \( K_v = \pi \exp[-(v + 2)/2] \) and \( Q_u = \pi \times u / 6 \). The parameter \( \sigma \) is the ratio of the width of the Gaussian window over the length of the Gabor wavelets. Parameter \( u \) is the orientation of Gabor wavelets. Herein, \( u \) and \( v \) control the orientation and scale of the Gabor wavelets respectively. In this thesis, we use \( \sigma = \pi \) and \( Q_u = \pi \).

- Circular filter

Figure 3.7: Filters for eye candidate detection (a) Gabor filter (b) Circular filter.
A circular filter, shown in Fig. 3.7b, is defined as

\[ h_2(x, y) = (\frac{2}{1 + [(x^2 + y^2)/2\alpha^2]^n} - 1) \times \frac{1}{1 + [(x^2 + y^2)/2\beta^2]^m}, \]  

(3.9)

where \( \alpha \) is the inner cut-off variable, \( \beta \) is the outer cut-off variable, \( n \) is the inner order, \( m \) is the outer order, \( \alpha < \beta \), and \( n < m \).

The Gabor filter and the circular filter are combined to generate the proposed eye filter and the output of the eye filter is shown in Fig. 3.8a:

\[ h(x, y) = h_1(x, y) + h_2(x, y). \]  

(3.10)

Figure 3.8: Steps for finding eye candidates using eye filters: (a) eye filter output, (b) connected component labelling, (c) image erosion and dilation, (d) candidate eye points (shown in red colour).
3.5 Extraction of eye candidate regions

The connected components are extracted from the filter output. Image erosion and dilation, based on the following equations, are performed to remove noise regions.

\[
\begin{align*}
\text{Erosion} : & \quad A \otimes B \\
\text{Dilation} : & \quad A \circ B
\end{align*}
\]

(3.11)

where \( B \) is a square structuring element suitable for the erosion or dilation of \( A \). Figure 3.6d and 3.8d show the final eye candidates detected using horizontal filter and eye filter, respectively. Note that it is essential to eliminate the false candidates.

3.6 Geometric face model and template matching

To eliminate false eye candidates, we construct an image region based on each eye pair and verify if the region is a face pattern. For a given eye pair, there are two possible face candidates, as shown in Fig. 3.9. We construct this geometric face model that reflects the relative anthropometric distances between the facial landmarks. This face model can also cope with all in-plane rotation and some out-of-plane rotation [9].

Based on the two eye points \( e_1 = (e_{1x}, e_{1y}) \) and \( e_2 = (e_{2x}, e_{2y}) \), the four corners of the face region are determined as follows [8].

- Compute the half distance between the two points:

\[
D = \frac{1}{2} \sqrt{(e_{2x} - e_{1x})^2 + (e_{2y} - e_{1y})^2},
\]

(3.12)
3.6. Geometric face model and template matching

- Compute the boundary points $p_1$ and $p_2$ along the eye line, as in Fig. 3.9:

  \[ p_{1x} = \frac{(3e_{1x} - e_{2x})}{2}, p_{1y} = \frac{(3e_{1y} - e_{2y})}{2}, \]
  \[ p_{2x} = \frac{(3e_{2x} - e_{1x})}{2}, p_{2y} = \frac{(3e_{2y} - e_{1y})}{2}. \]  
  (3.13)

- Compute the four face corners $r_1, r_2, r_3, r_4$. Assuming that the face angle is $\alpha$, the following formulas are applicable to both cases in Fig. 3.9:

  \[ r_{1x} = p_{1x} + D \sin \alpha, \quad r_{1y} = p_{1y} - D \cos \alpha, \]
  \[ r_{2x} = p_{1x} - 3D \sin \alpha, \quad r_{2y} = p_{1y} + 3D \cos \alpha, \]
  \[ r_{3x} = p_{2x} - 3D \sin \alpha, \quad r_{3y} = p_{2y} + 3D \cos \alpha, \]
  \[ r_{4x} = p_{2x} + D \sin \alpha, \quad r_{4y} = p_{2y} - D \cos \alpha. \]  
  (3.14)

- Form face candidates. Two face masks can be formed along two opposite orientations, but only the one that is part of the original face bounding area is selected. This face candidate is rotated by a face angle $\alpha$ to the upright position.

![Figure 3.9: Determining the face boundary for a given eye pair using a geometric face model [8].](image-url)
3.6. Geometric face model and template matching

- Compare aligned face candidates with a face template. The face template (Fig. 3.10) used in this thesis is generated by averaging 15,000 aligned upright frontal face patterns. The correlation coefficient between the face candidate and the face template is calculated. Among the overlapping candidates, the one with the maximum correlation score is considered as the true face. Figure 3.11b shows the corrected upright face image after rotation by the angle of $\alpha$.

![Figure 3.10: A face template constructed using 15,000 frontal face images from the ECU database [9].](image)

![Figure 3.11: Eye detection and face alignment: (a) valid eye points after face verification, (b) aligned face pattern.](image)

Table 3.1 shows the results of both eye detection methods evaluated on the JAFFE database. Compared to the face image processed using the horizontal non-linear filter, the output of the eye filter has less noise points and is more suitable for generating eye candidates as it reduces the computational cost.

The proposed eye detection method can also cope with images containing
Table 3.1: Eye detection performances of the horizontal filter and eye filter tested on the JAFFE database.

<table>
<thead>
<tr>
<th>Method</th>
<th>Detection Rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Horizontal filter + template matching</td>
<td>98.1</td>
</tr>
<tr>
<td>Eye filter + template matching</td>
<td>100.0</td>
</tr>
</tbody>
</table>

Figure 3.12: An example of multiple eye detection. (a) The original image. (b) Eye detection results on the original image.

multiple faces, Fig. 3.12 shows an example of detecting eye positions on an image that has more than one face.

3.7 Chapter summary

In this chapter, we have proposed a novel eye detection and face alignment method which are used to correct localisation errors in the existing face detectors. In the first step, we generate possible eye candidates. Two approaches are presented: One uses a fixed horizontal non-linear filter and the other uses an eye filter which is the combination of a Gabor filter and a circular filter. In the second step, we use
a template matching approach to eliminate the non-eye candidates and select the accurate eye pairs. Experimental results show that the second approach, which is based on the eye filter, outperforms the one using horizontal non-linear filter.
Chapter 4

Facial Expression Recognition System

Chapter contents

4.1 System architecture overview ........................................ 54
4.2 Stage 1: Fixed, directional filters ................................. 55
4.3 Stage 2: Adaptive filters ............................................. 57
4.4 Stage 3: Classification .................................................. 58
4.5 Training methods ....................................................... 60
   4.5.1 Resilient backpropagation algorithm ....................... 60
   4.5.2 Levenberg-Marquardt algorithm ............................. 62
4.6 Support vector machines .............................................. 63
   4.6.1 Mathematical background .................................... 64
   4.6.2 Non-linear SVMs ................................................ 67
   4.6.3 Multi-class SVMs ............................................... 68
4.7 Feature selection ....................................................... 69
   4.7.1 General characteristics of feature selection ............... 70
   4.7.2 General feature selection algorithms ....................... 73
   4.7.3 Proposed feature selection method .......................... 75
4.8 Chapter summary ....................................................... 76

A basic FER system has three major steps - face detection and alignment, facial feature extraction, and facial expression classification. Chapter 3 proposed a novel face detection and alignment method which provides upright frontal face images
4.1 System architecture overview

As input to the FER system. In this chapter, we propose a hierarchical architecture for facial feature extraction and facial expression classification.

To reduce the effects of skin tone, we utilise the grey-scale images for FER. Colour images provide satisfactory results in terms of feature extraction, however, the differences in skin tone may increase the complexity in pattern recognition. In practice, grey-scale images can provide sufficient facial features for recognition tasks. Algorithms developed for grey-scale images can also be implemented to process colour images.

4.1 System architecture overview

The proposed feature extraction architecture consists of three processing stages [75]. Figure 4.1 gives an overview of the new hierarchical structure. Stages 1 and 2 comprise non-linear filters, which are designed for extracting hierarchical visual

![Block diagram of the proposed system.](image-url)
features, whereas Stage 3 is used for classification of these features. For a given input pattern, Stage 1 uses fixed filters to detect elementary features at several directions. In comparison, Stage 2 uses adaptive filters to extract more specific, salient features.

### 4.2 Stage 1: Fixed, directional filters

Stage 1 has fixed, directional filters and is applied to extract features at different orientations. These filters are based on a biological mechanism, known as the *shunting inhibition*. This mechanism, found in the cortical cells of the human visual system [77], has been adopted for digital image enhancement [78]. The output response of the first stage is given as

\[
Z_{1,i} = \frac{D_i \ast I}{G \ast I},
\]

where \(I\) is a 2-D input face pattern, \(Z_{1,i}\) is the output of the \(i\)-th filter, \(D_i\) and \(G\) are the filter coefficients, \(\ast\) denotes 2-D convolution, and the division is done pixel-wise. In this thesis, the subscripts 1 and 2 in \(Z_{1,i}\) and \(Z_{2,i}\) indicate the outputs of the first and second processing step, respectively. The kernel \(G\) is chosen as an isotropic Gaussian kernel:

\[
G(x, y) = \frac{1}{2\pi \sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right).
\]

To extract elementary facial features at different directions, the kernel \(D_i\) is formulated as the \(M\)-th order derivative of Gaussian. Its coefficients are defined as [79]

\[
D_i(x, y) = \sum_{k=0}^{M} \frac{M!}{k!(M-k)!} \frac{x^k y^{M-k}}{\partial x^k \partial y^{M-k}} \frac{\partial^M G(x, y)}{\partial x^k \partial y^{M-k}},
\]

where
4.2. Stage 1: Fixed, directional filters

- $M$ is the derivative order, $M = 1, 2, \ldots$,
- $\theta_i$ is the angle of rotation, $\theta_i = \frac{(i-1)\pi}{N_1}$ for $i = 1, 2, \ldots, N_1$,
- $s_x = \sin \theta_i$ and $s_y = \cos \theta_i$.

The partial derivative of the Gaussian with respect to dimension $x$ or $y$ can be computed as the product of the Hermite polynomial and the Gaussian function,

$$
\frac{\partial^k G(x, y)}{\partial x^k \partial y^k} = \frac{(-1)^k}{(\sqrt{2}\sigma)^k} H_k\left(\frac{x}{\sqrt{2}\sigma}\right) H_k\left(\frac{y}{\sqrt{2}\sigma}\right) G(x, y),
$$

(4.4)

where $H_k()$ is the Hermite polynomial of order $k$.

To achieve image classification that tolerates small translations and geometric distortions in the input image, a sub-sampling operation on the filter output is performed to reduce their spatial resolution by half. This operation decomposes each filter output into four smaller maps as shown in Fig. 4.2:

$$
Z_{1,i} \rightarrow \{Z_{2,4i-3}, Z_{2,4i-2}, Z_{2,4i-1}, Z_{2,4i}\}.
$$

(4.5)

Figure 4.2: The sub-sampling operations performed in stage one.
The next processing step is motivated by the centre-surround receptive fields that are found in the lateral geniculate nucleus of the thalamus in the brain. There are two major configurations: on-centre and off-centre. Accordingly, each sub-sampled map \( Z_{2,i} \), where \( i = 1, 2, ..., 4N_1 \), is separated into an on-response map and an off-response map, using zero as threshold:

\[
Z_{2,i} \rightarrow \begin{cases} 
\text{on} & : Z_{3,2i-1} = \max(Z_{2,i}, 0) \\
\text{off} & : Z_{3,2i} = -\min(Z_{2,i}, 0)
\end{cases}.
\] (4.6)

For the on-response map, all negative outputs are set to 0. For the off-response map, all positive outputs are set to 0 and the entire map is then negated. At the end of Stage 1, the features in each map are contrast-normalised, using the following transformation

\[
Z_{4,i} = \frac{Z_{3,i}}{(Z_{3,i} + \mu_i)},
\] (4.7)

where \( \mu_i \) is the mean value of the absolute response of the output map of the directional filters before down sampling, and the division is performed pixel-wise.

### 4.3 Stage 2: Adaptive filters

Stage 2 aims at detecting more specific features that will simplify the classification task. The output maps produced by each directional filter in Stage 1 are processed by exactly two filters in Stage 2, one for on-response maps and the other for off-response maps. The filters in Stage 2 are also based on the shunting inhibition mechanism. Consider \( Z_{4,i} \) as the input of Stage 2. Suppose that \( P_k \) and \( Q_k \) are two convolution masks of the adaptive filter. The filter output is calculated as

\[
Z_{5,i} = \frac{g(P_k * Z_{4,i} + b_k) + c_k}{a_k + f(Q_k * Z_{4,i} + d_k)},
\] (4.8)
where $a_k$, $b_k$, $c_k$ and $d_k$ are adjustable bias terms, and $f$ and $g$ are two activation functions.

To form a feature vector, a sub-sampling operation is performed across each set of four output maps. From four output maps, each non-overlapping block of size $(2 \times 2 \text{ pixels}) \times (4 \text{ maps})$ is averaged into a single output signal, as shown in (4.9):

$$\{Z_{5,4i-3}, Z_{5,4i-2}, Z_{5,4i-1}, Z_{5,4i}\} \rightarrow Z_{6,i}. \quad (4.9)$$

![Diagram of sub-sampling operations](image)

Figure 4.3: The sub-sampling operations performed in stage two.

### 4.4 Stage 3: Classification

The features produced by Stage 2 are sent to the classification stage that may consist of any type of classifiers. In this section, we detail the classifiers used in the proposed FER system.

In the field of pattern recognition, a classifier aims to determine the correct class an object belonging to, based on the object’s characteristics which are also
known as feature vectors. The linear classifier can be considered as the simplest classification tool, which uses a linear combination of the feature vectors to make the decision. Figure 4.4 shows an example of different linear classifiers.

Assuming $C_1$ and $C_2$ are two groups of features to be classified, and $h_1$ (blue), $h_2$ (red) and $h_3$ (green) are three hyperplanes found by different linear classifiers. Both $h_1$ and $h_2$ can separate these two classes correctly. However, $h_2$ is considered to be better, because it has the biggest distances between each class. Hyperplane $h_3$ fails to classify the two groups of features. Generally, the samples are linear separable if a linear function can be found which correctly separates the sample groups. Here, this linear function is called a hyperplane.

To solve a simple two-class classification problem, a threshold value is added to the linear function to decide the corresponding class of the test sample. For example, a linear function is defined as

$$g(x) = wx + b,$$  \hspace{1cm} (4.10)

and the threshold value is chosen to be 0. There is a sample $x_i$ to be classified, and
the value of the classification function is \( g(x_i) \). If \( g(x_i) > 0 \), the sample \( x_i \) belongs to class \( C_1 \); otherwise, if \( g(x_i) < 0 \), the sample is classified to \( C_2 \).

In the proposed FER system, a simple linear classifier is used, whose output \( y \) is given as

\[
y_j = \sum_{i=1}^{N_3} w_{ij} Z_{6,i} + b_j, \quad j = 1, 2, ..., N_4
\]  

(4.11)

where \( w_{ij} \)'s are adjustable weights, \( b_j \) is an adjustable bias term, \( Z_{6,i} \)'s are input features to Stage 3, \( N_3 \) is the number of input features, and \( N_4 \) is the number of output nodes. The output \( y = [y_1, y_2, ..., y_{N_4}]^T \) indicates the class or the label of the input pattern \( I \).

4.5 Training methods

Two training methods are implemented to train the adaptive parameters of the proposed method, namely, Resilient backpropagation algorithm and Levenberg-Marquardt algorithm.

4.5.1 Resilient backpropagation algorithm

To satisfy the large demand of memory for experiments with large databases, the Resilient backpropagation (Rprop) [80] training algorithm is used. It is a first-order optimisation algorithm which has been used for supervised learning in feed-forward neutral networks.

The training technique used in this project is proposed by Tivive et al. [75], which combines Rprop with Least-square (LS) method. Consider a training set of \( K \) input patterns \( I_1, I_2, ..., I_K \), and \( K \) corresponds to desired outputs \( d = d_1, d_2, ..., d_K \). Let \( w \) be all parameters of the adaptive filters and the linear classifier, arranged...
4.5. Training methods

as a column vector, \( \mathbf{w} = [w_1, w_2, ..., w_N]^T \). The networks are trained based on the following steps.

**Step 1:** The trainable coefficients of non-linear filters are initialised with random values in the range \([-1, 1]\) in Stage 2.

**Step 2:** The outputs of each stage are found using forward computation based on the input training patterns.

**Step 3:** The weights and bias of the linear classifier are determined by the least square method. Consider \( \mathbf{Z}_6 \) be the input vectors, generated from the given training sets, which are sent to the linear classifier. All of the weights and bias of the linear classifier are defined as: \( \mathbf{w} = [w_1, w_2, ..., w_N, b]^T \). Then we can find the parameters of the linear classifier by solving the following equation:

\[
\text{minimize error function } E(\mathbf{w}) = \| \mathbf{Z}_6 \mathbf{w} - \mathbf{d} \|^2.
\]  

The least-square solution is given as

\[
\mathbf{w} = (\mathbf{Z}_6^T \mathbf{Z}_6)^{-1} \mathbf{d}.
\]  

**Step 4:** The error between the actual outputs of the linear classifier and the desired outputs are computed. We use back-propagation to compute the gradient error \( g(\mathbf{w}) \) for all trainable coefficients \( \mathbf{w} \) in Stage 2. The trainable coefficients \( \mathbf{w} \) of the non-linear filters at epoch \( t \) is updated using the following equation.

\[
\mathbf{w}(t + 1) = \mathbf{w}(t) + \Delta \mathbf{w}(t) + \mu(t)\Delta \mathbf{w}(t - 1).
\]  

The weight update \( \Delta \mathbf{w}(t) \) is computed based on the sign of the error gradient

\[
\Delta \mathbf{w}(t) = -\text{sign}[g(t, \mathbf{w})] \gamma(t),
\]
4.5. Training methods

where

$$\gamma(t) = \begin{cases} \max(0.5\gamma(t-1), 10^{-10}), & \text{if } g(t,w)g(t-1,w) < 0 \\ \min(1.2\gamma(t-1), 10), & \text{if } g(t,w)g(t-1,w) > 0 \\ \gamma(t-1), & \text{if } g(t,w)g(t-1,w) = 0 \end{cases}$$ \quad (4.16)

In Eq. (4.14), $\mu(t)$ is the momentum term which is calculated as

$$\mu(t) = \left| \frac{g(t,w)}{g(t-1,w) - g(t,w)} \right|. \quad (4.17)$$

**Step 5:** Repeat Steps 2 to 4 until the maximum number of training epochs is reached or the error is below a predefined limit.

4.5.2 Levenberg-Marquardt algorithm

A second-order training method called the Levenberg-Marquardt (LM) algorithm [81] is used to optimise the parameters of the adaptive filters in Stage 2 and the classifier in Stage 3. It is a fast and effective training method as it combines the stability of the gradient descent with the speed of Newton algorithm. The main steps of the LM algorithm are as follows:

**Step 1:** Initialise trainable coefficients of nonlinear filters in Stage 2 and the parameters of the linear classifier in Stage 3 with random values from a uniform distribution in the range $[-1, 1]$.

**Step 2:** Perform forward computation to find the outputs of each stage in response to the training patterns.

**Step 3:** Calculate the weight update at iteration $t$ as

$$\Delta w(t) = [J^T(t)J(t) + \mu(t)I]^{-1} J(t) e(t) \quad (4.18)$$

where $J(t)$ is the Jacobian matrix, $I$ is the identity matrix and $\mu(t)$ is a regularization term to avoid the singularity problem. During training, the
4.6 Support vector machines

The regularization parameter is increased or decreased by a factor of ten, depending on the decrease or increase of the mean-square-error, respectively. Let $e = d - y$ be the error matrix. The Jacobian matrix is defined as the partial derivative of the mean-square-error with respect to each trainable parameter:

$$ J = \begin{bmatrix} \frac{\partial e_1}{\partial w_1} & \frac{\partial e_1}{\partial w_2} & \cdots & \frac{\partial e_1}{\partial w_N} \\ \frac{\partial e_2}{\partial w_1} & \frac{\partial e_2}{\partial w_2} & \cdots & \frac{\partial e_2}{\partial w_N} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial e_N}{\partial w_1} & \frac{\partial e_N}{\partial w_2} & \cdots & \frac{\partial e_N}{\partial w_N} \end{bmatrix}, \quad (4.19) $$

The partial derivatives can be computed via error back-propagation, similarly to [82].

**Step 4:** Repeat Steps 2 to 3 until the maximum number of training epochs is reached or the error is below a predefined limit.

### 4.6 Support vector machines

The Support Vector Machine (SVM) is a widely used classifier in machine learning and pattern recognition. It was first proposed by Vapnik in 1995 [83] based on the theories of Vapnik-Chervonenkis dimension and the structural risk minimisation (SRM). The SVM can be applied to solve non-linear or high dimensional classification problems. It is a supervised learning algorithm and capable of generating the minimum error and maximum margins between classes.

We assume there are two classes in the sample space. A number of hyperplanes
can be found to separate these two classes, as shown in Fig. 4.5a. However, only
one hyperplane has the maximum margins between the two classes which can get
the most effective separation, as indicated in Fig. 4.5b. The SVM classifier aims to
generate the hyperplane that has the maximum margins.

![Figure 4.5: (a) a number of hyperplanes that can separate the two classes (rectangles and circles). (b) the hyperplane generated by SVM which has the maximal margin.](image)

### 4.6.1 Mathematical background

Assume the training set for SVM is:

\[
(x_1, y_1), ..., (x_i, y_i), ..., (x_N, y_N), \quad x_i \in \mathbb{R}^n, y_i \in \{+1, -1\}
\]  

(4.20)

where \(x_i\) is the \(i\)-th feature vector, \(y_i\) indicates the class of vector \(x_i\), and \(N\) is
the number of feature vectors. There exists a hyperplane that can separate the
training set, and it can be expressed as:

\[
f(x) = w \cdot x + b
\]

(4.21)

where \(w\) is the normal vector of the hyperplane, and \(b\) is a bias term. To best
separate the training set, we need to find the maximal margin hyperplane which
4.6. Support vector machines

meets the following requirements:

\[
\begin{align*}
\mathbf{w} \cdot \mathbf{x}_i + b &\geq +\delta \quad \text{for} \quad y_i = +1 \\
\mathbf{w} \cdot \mathbf{x}_i + b &\leq -\delta \quad \text{for} \quad y_i = -1
\end{align*}
\] (4.22)

The above equations indicate that there exists a pair of \( \mathbf{w} \) and \( b \), which can separate all the feature vectors in the training sample into +1 or -1 group. These two equations can be further combined when \( \delta \) is set to 1:

\[y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1, \forall i\] (4.23)

Figure 4.6 provides an example of both the separating hyperplane and the support hyperplane, where \( d_1 \) and \( d_2 \) are the distances from the original point to \( L1, L2 \), respectively. The distances can be expressed as:

\[d_1 = \frac{|1 - b|}{||\mathbf{w}||}, \quad d_2 = \frac{|-1 - b|}{||\mathbf{w}||}\] (4.24)

The distance of the boundaries indicated by \( L1 \) and \( L2 \) is \( 2/||\mathbf{w}|| \). The maximum value of margin is \( 2/||\mathbf{w}|| \).

To get the optimal solution, we can use the Lagrange Multiplier method [84]. Equation (4.23) can be then expressed as:

\[L(\mathbf{w}, b, \alpha) = \frac{1}{2}||\mathbf{w}||^2 - \sum_{i=1}^{N} \alpha_i [y_i(\mathbf{w} \cdot \mathbf{x} - 1)]\] (4.25)

where \( \alpha_i \geq 0 \) is the Lagrange multiplier of \( \mathbf{x}_i \), \( i = 1, ..., N \). Although the optimal values of \( \mathbf{w} \) and \( b \) can be found using Lagrange Multiplier method, it may also cause a problem known as Wolfe dual [85]. Therefore, a linear second-order equation can be solved using Karush-Kuhn-Tucker conditions [86] to avoid the dual problem.
4.6. Support vector machines

The following equations are used to compute the partial derivatives of Eq. (4.25):

\[ \begin{aligned}
\text{partial derivative of } w: & \quad \frac{\partial}{\partial w} L(w, b, \alpha) = w - \sum_{i=1}^{N} \alpha_i y_i x_i = 0 \\
\text{partial derivative of } b: & \quad \frac{\partial}{\partial b} L(w, b, \alpha) = \sum_{i=1}^{N} \alpha_i y_i = 0
\end{aligned} \]  

(4.26)

where \( \alpha_i \) corresponds to each feature vector \( x_i \) of the training set. If \( \alpha_i \geq 0 \), then \( x_i \) is a support vector located on the support hyperplane. Taking this \( \alpha_i \) into the partial derivative of \( w \), we can get the corresponding \( w \). To calculate the value of \( b \), we use Karush-Kuhn-Tucker, which is shown as follows:

\[ b = y_i - \sum_{i=1}^{N} \alpha_i y_i (x_i \cdot x_j^T), \quad j \in \left\{ j \mid \alpha_j > 0 \right\}. \]  

(4.27)

Then a discriminant equation can be generated as

\[ f(x) = \text{sgn} \left( \sum_{i=1}^{N} \alpha_i y_i (x_i \cdot x + b) \right). \]  

(4.28)

where \( f(x) \) denotes the output test result. If \( f(x) > 0 \), the input sample belongs to class of \( +1 \), otherwise, the input data belongs to class of \( -1 \).
4.6. Support vector machines

4.6.2 Non-linear SVMs

In the previous section, the input patterns can be simply classified by a linear function. However, in practice, the input patterns may not be separable using linear classifiers. Therefore, the non-linear SVMs are designed. Figure 4.7a shows the features in the input space that cannot be separated using linear functions. Figure 4.7b demonstrates the hyperplane which separates the features in a high dimensional space that is mapped from the input space using a kernel function $\phi$.

![Figure 4.7: Classify in high dimensional feature space. (a) features in the input space. (b) mapping the features into high dimensional space.](image)

Equation (4.29) indicates the classification function in high dimensional space:

$$f(x) = \text{sgn}\left(\sum_{i=1}^{n} \alpha_i y_i K(x_i, x) + b\right).$$

(4.29)

where $K(x_i, x)$ represents the kernel function that transfers the features from the input space to a high dimensional feature space. There are four basic kernel functions [87]:

- linear kernel: $K(x_i, x) = x_i^T x_j$.
- polynomial: $K(x_i, x) = (\gamma x_i^T x_j + r)^d$, $\gamma > 0$.
- radial basis function (RBF): $K(x_i, x) = \exp(-\gamma \|x_i - x_j\|^2)$, $\gamma > 0$. 

---

67
4.6. Support vector machines

- sigmoid: \( K(x_i, x) = \tanh(\gamma x_i^T x_j + r) \).

Here, \( \gamma, r, \) and \( d \) are kernel parameters. In this thesis, we focus on the RBF kernel.

4.6.3 Multi-class SVMs

SVMs have been developed originally for two-class problems. To cope with problems of multi-class classification, we can construct several SVMs to differentiate each pair of classes, as shown in Fig. 4.8. This method is known as pair-wise SVMs. It converts a \( k \)-class classification problem into a \( k(k-1)/2 \) two-class classification problem. For example, for seven facial expressions, we need 21 pair-wise SVMs.

![Figure 4.8: Apply SVMs to multi-class problem (4 classes).](image)

The final result of pair-wise SVM classification is generated using a majority voting strategy based on all the two-class SVM classifiers. We assume each two-class SVM classifier, \( C_{n,m} \), produces a decision value, \( w \). If \( w > 0 \), the test pattern is categorised into class \( n \) and the voting score, \( V_n \), is increased by one; otherwise, the test pattern belongs to class \( m \) and \( V_m = V_m + 1 \). This procedure is repeated \( k(k - 1)/2 \) times to derive the final voting scores of all classes. Class \( i \) with the maximum voting score, \( \max(V) \), is considered to be the corresponding facial expression.
expression category of the input pattern.

We use SVMs for Stage 3 to improve classification accuracy. A two-step process is adopted to integrate SVM to the proposed FER system. First, we assume that a linear classifier is used in Stage 3, and calculate the coefficients of the adaptive filters in Stage 2 and the weights of the linear classifier, using the LM or Rprop training method and the least-squares method. Second, once the filters in Stage 2 are trained and $N_3$ features are extracted, we train the multi-class SVM to classify a selected subset of these features.

4.7 Feature selection

Feature selection is a research topic that spans research disciplines such as statistics [88], pattern recognition [89], data mining [90], machine learning [91], neural networks [92], mathematical programming [93], and many others.

All input features, including irrelevant features, can be utilised when there are sufficient data and time. However, in practice, there are two problems with using all of the features.

- The computational cost is increased by the irrelevant input features.
- Over-fitting is induced by the irrelevant input features.

Feature selection can eliminate the features that have little effect on the output, to keep the size of the essential feature vectors small [94]. It reduces the dimensionality of the feature space and removes the redundant information. The immediate effects for feature selection are reducing the training time of the learning algorithms, improving the feature quality, increasing the overall system performances,
and building a more accurate classification model.

Formally, the problem of feature selection is to find a sub-space $K \subseteq M$ such that:

$$C(M) = \max_{K \subseteq M, |K| = p} C(K).$$

(4.30)

where $M$ is the original feature space which contains $q$ features, and $K$ is the feature space that has been selected from the original space with a cardinality of $p$, $p \leq q$, and $K \subseteq M$. Here, $C(K)$ indicates the selection criterion for selected feature space $K$, which can also be considered as the performance of the selected sub-space. In general, the higher is the value of $C$, the better is the feature space. Equation (4.30) aims to find the feature space that produces the maximum value of $C$.

### 4.7.1 General characteristics of feature selection

Feature selection performs a search through the space of all feature subsets. Blum and Langley summarised that there are four basic issues affect the performance of heuristic search and highlighted the characteristics of feature selection.

- **Starting point**

This is the point to start the selection of feature subset. There are several ways to start the searching process. Figure 4.9 shows an example of the feature selection procedure. One option, which is considered to proceed forward through the search space, is to begin with no features and successively add new attributes. This case begins from the left in Fig. 4.9 and incrementally moves to the right. By contrast, the search can begin with all features and successively remove the irrelevant ones, as shown in Fig.
4.7. Feature selection

4.9 from right to left. In this case, the search proceeds backward through the search space. Another option is to begin search in the middle and move outwards.

Figure 4.9: Feature selection procedures.

- Search strategy

To search for an optimal feature subspace, an exhaustive search can be applied. However, this search method is very costly in terms of time and resources when the number of features is large. For example, if the total number of features is $M$, there are $2^M$ possible feature subsets. To cope with large number of features as well as reduce computational expenses, the heuristic search strategies are introduced which are more feasible than exhaustive search. Result generated by heuristic search strategy may not be optimal, nevertheless, they still provide satisfactory recognition accuracy.

- Evaluation strategy

The evaluation of selected feature subsets is considered as an important step in different feature selection methods. The selected feature subsets are
expected to optimally separate the target classes and improve the system classification performance. As mentioned by [97], the class separation ability is generally evaluated by an intra-class distance measure [98]. Another commonly used measure approach is known as the Wilks’ lambda [99]:

\[ \lambda = \frac{|W|}{|W + B|} \]  

(4.31)

where \( W \) is the intra-class matrix dispersion corresponding to the selected feature set, \( B \) is the intra-class matrix, \(|W|\) is the determinant of matrix \( W \).

The \( W \) and \( B \) can be generated as follows:

\[ W = \sum_{j=1}^{g} \sum_{l=1}^{N_j} (x_l' - \mu_j')(x_l' - \mu') \]  

(4.32)

\[ B = \sum_{j=1}^{g} N_j(\mu - \mu_j)(\mu - \mu_j) \]  

(4.33)

where \( g \) is the number of classes, \( N_j \) is the number of samples in class \( j \), \( \mu_j \) is the average value of class \( j \), and \( \mu \) is the global average value. A smaller value of \( \lambda \) indicates a better separation ability.

To evaluate the performance of the selected feature subsets, we utilise the classification accuracy that is defined as the correctly classified samples over the entire input samples.

- **Stopping criterion**

It is vital for the feature selection algorithm to know when to stop searching. It may stop selecting features when none of the unselected features can provide improvement to the classification accuracy of the selected feature subset. In other words, the selection algorithm keeps updating the selected
feature subset until the evaluation performance starting to degrade. However, there exist searching algorithms that search through the entire feature space and select the feature subset based on the best evaluation performance. This type of searching approaches is considered to be time-consuming.

4.7.2 General feature selection algorithms

The feature selection strategies differ mainly in the way how the feature subsets are selected for evaluation. In this section, we introduce two basic feature selection algorithms: brute force selection and forward selection [94].

• Brute force selection

The brute force selection algorithm aims to exhaustively evaluate all possible combinations of the input features to find the best feature subset. This algorithm firstly searches for the best subset with one feature that is defined as $X_i$ from all the input features, $X_1, X_2, ..., X_M$, where $M$ is the number of the input features. Then it starts to find the best feature subset containing two features that might be any two of the input features. In the same way, this algorithm incrementally searches for the best feature subset until the optimal number of features, $m$, has been reached. The cost of exhaustive search can be calculated as

$$M \times C_1 + F(M, 2) \times C_2 + ... + F(M, i) \times C_i + ... + F(M, m) \times C_m,$$

where $C_i$ is the computational expense of a single evaluation. We can compute $F(M,i)$ using equation defined as

$$F(M, i) = \frac{M!}{i! \cdot (M - i)!},$$

where $i$ is the number of elements in each combination. Equation (4.34) indicates
that the brute force algorithm has extremely high computational cost and may lead to over-fitting of the system.

• **Forward selection**

The first step of forward feature selection algorithm is similar as the brute force algorithm, which evaluates all the feature subsets with one feature selected. Generally, this algorithm utilises the Leave-One-Out (LOO) cross validation error to measure the performances of the feature subsets with one feature component, \(X_1, X_2, \ldots, X_M\), and selects the best feature subset \(X_1\) [94]. In the second step, \(M - 1\) feature subsets with two features are formed. Each of the feature subset consists of the first selected feature and one feature from the remaining \(M - 1\) features. The second feature is selected from the feature subset with least LOO cross validation (LOOCV) error. Then additional features are added one-by-one from the remaining \(M - 2\) input features. This process needs \(M\) steps to complete searching and, finally, the best feature subset with \(m\) features, \(X_1, X_2, \ldots, X_m\), is selected.

To evaluate the cost of forward feature selection algorithm, we assume \(C_i\) is the computational cost of evaluating \(i\) features. Therefore, the overall computational cost for a feature subset of size \(m\) is

\[
M \times C_1 + (M - 1) \times C_2 + \ldots + (M - m + 1) \times C_m.
\]  

(4.36)

Forward selection algorithm is less costly compared to brute force selection algorithm. However, the evaluation performance of forward selection may not be comparable to that of the brute force algorithm. For example, if \(X_1\) is the best feature selected in the first step, it cannot guarantee that the classification ability of feature subset \(\{X_2, X_3\}\) is worse than that of \(\{X_1, X_2\}\) or \(\{X_1, X_3\}\). Thus, a feature
4.7. Feature selection

subset selected by the forward selection algorithm may not be the same as that selected by brute force algorithm.

4.7.3 Proposed feature selection method

The purpose of feature selection is to find a subset of features that jointly lead to the best separation of the target classes. The steps in our approach can be described as follows. Let \( S_t \) be the set of selected features at round \( t \). Let \( D_{\text{train}} \), \( D_{\text{valid}} \), and \( D_{\text{test}} \) be the training, validation and test set, respectively.

- Step 1: Calculate the class separation score (CSS) for each feature in the feature pool. Let \( f^* \) be the feature with the highest CSS. Initialise \( S_0 = \{ f^* \} \).

- Step 2: At round \( t \), consider a feature \( f \) in the remaining feature pool. Train the classifier with features \( \{ S_{t-1}, f \} \) on \( D_{\text{train}} \), and evaluate it on \( D_{\text{valid}} \). Calculate also the mutual information score for \( \{ S_{t-1}, f \} \) on \( D_{\text{train}} \).

- Step 3: Repeat Step 2 for all features in the remaining feature pool.

- Step 4: Select \( a \) features that correspond to the highest CR when added to \( S_{t-1} \). If several features have the same CR, select features that have the lower mutual information scores.

- Step 5: Increment \( t \) and go to Step 2 until a defined number of features are selected.

- Step 6: Train final classifier on \( D_{\text{train}} \) and evaluate its performance on \( D_{\text{test}} \).

Next, we explain how the class separation score (CSS) and the mutual information are calculated. Let \( N \) be the number of classes, \( N = N_4 \) in the proposed
4.8. Chapter summary

architecture. For feature $f$, let $p_{fi}(x)$ be the class-conditional probability density function (pdf) for class $i$. The class separation score for feature $f$ is computed as

$$ C(f) = \sum_{i \neq j} \int p_{fi}(x) \log p_{fj}(x) dx. \quad (4.37) $$

A higher $C(f)$ means a better separation between the classes by feature $f$. In our work, the class-conditional pdfs are estimated via Parzen window method with Gaussian kernels.

The mutual information of a feature set is the sum of mutual information between all feature pairs:

$$ M = \sum_{m < n} M(f_m, f_n). \quad (4.38) $$

In this thesis, we analyse two methods of calculating the mutual information between two features. Consider two features $f_m$ and $f_n$. Let $p_m(x)$ and $p_n(x)$ be probability density functions of the two features, calculated on the entire training set. Let $p_{mn}(x, y)$ be the joint pdf of these two features.

**Method 1**: The mutual information is defined based on the symmetric Kullback-Leibler divergence:

$$ M(f_m, f_n) = - \int [p_m(x) \log p_n(x) + p_n(x) \log p_m(x)] dx, \quad (4.39) $$

**Method 2**: The mutual information is defined based on the joint pdf:

$$ M(f_m, f_n) = - \int \int p_{mn}(x, y) \log \frac{p_{mn}(x, y)}{p_m(x)p_n(y)} dxdy \quad (4.40) $$

4.8 Chapter summary

In this chapter, we described the design of the proposed FER system. This system consists of three stages. In Stage 1, the fixed, directional filters are employed
to extract primitive features. The adaptive filters in Stage 2 are used to extract more complex feature for classification in Stage 3. The filters in Stage 1 and 2 are designed based on a biological mechanism called *shunting inhibition*. In Stage 3, both linear and SVM classifiers are evaluated. To improve the system efficiency and eliminate redundant features, the feature selection technique is introduced to extract salient features for classification. The experimental steps and results are discussed in Chapter 5.
In this chapter, we present the experimental results of our proposed face alignment method and FER system. Section 5.1 introduces the two databases used in this project, and outlines the parameters chosen for the FER system. Section 5.2 presents the face detection and alignment results tested on JAFFE database. Section 5.3 presents the classification rates of the proposed FER system evaluated on both the JAFFE and MMI database. Section 5.4 analyses the classification results of the FER system with feature selection, and Section 5.5 compares the proposed FER system with several existing methods.
5.1 Databases and experimental steps

In this section, the databases used to evaluate the proposed FER system are introduced and the experimental steps and parameters are outlined.

5.1.1 JAFFE and MMI databases

We analyse the performance of the proposed method on the standard JAFFE database [15], which is commonly used for research on FER. This database consists of 213 images from 10 Japanese actresses. They are instructed to produce seven types of facial expressions. For each person, two to four images are recorded for each facial expression. Figure 5.1 provides several example images taken from the JAFFE database.

![Example facial expression images in JAFFE database.](image)

Figure 5.1: Example facial expression images in JAFFE database.
5.1. Databases and experimental steps

Another database, the MMI facial expression database [66], is also used to evaluate the proposed FER system. This database contains more than 2000 samples of static images and image sequences of faces in frontal or profile views. It has a web-based direct-manipulation platform which allows users to easily access and search for the expected facial expressions or single and multiple facial muscle actions. Among all video sequences in this database, 206 videos have been labelled with facial expression categories. Thus, the experiments are conducted using the image samples generated from these 206 video sequences. Figure 5.2 presents an example of the facial expression images taken from the MMI database.

![Facial expression images in MMI database.](image)

**Figure 5.2:** Example facial expression images in MMI database.

To prepare the training and test sets from the MMI database for FER system evaluation, we perform the following steps:

- Choose the first \(n\) frames at the beginning and \(m\) frames at the end of each video sequence to be sample images of neutral expression;

- Choose \(k\) frames from the centre section of each video sequence as sample images of the corresponding facial expression, where values of \(k\) can be
different according to each video.

After preprocessing of the video sequences, there are more than 15,000 facial images generated from the database.

5.1.2 Experimental steps and parameters

Guo and Dyer [54] evaluated several FER methods on the JAFFE database, using ten-fold validation. To enable comparison with their results, we adopt the same evaluation method. All images are divided into ten groups. For each validation fold, nine groups are used to train the classifier while the remaining group is used for testing. This step is repeated ten times, and the classification rates of the ten folds are averaged to form the final estimate of the classification rate.

For facial analysis, it has been shown that classification accuracy is improved by presenting both the input pattern and its mirror image to the classifier, and using the averaged response to form a classification decision [100]. Therefore, in this thesis we examine both classification approaches: (i) FER system with mirror image; (ii) FER system without mirror image.

The proposed system uses an input image of size $42 \times 32$ pixels. The order of the Gaussian derivative employed in Stage 1 to designed the directional non-linear filters can be varied. To determine a suitable value for $M$, we conducted preliminary experiments for $M$ equal to 1, 2, 3, 4, and 5. The classification rates for one trial are shown in Table 5.1. Based on this result, the order of Gaussian derivative filter is selected to be $M = 2$.

We also analysed different parameters such as the number of directions $N_1$, and the filter sizes in Stages 1 and 2. The number of directions is selected to be
5.2. Results of eye detection and face alignment

Table 5.1: Comparison of different values for $M$ - the order of Gaussian derivative filters.

<table>
<thead>
<tr>
<th>Order</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>CR %</td>
<td>94.4</td>
<td>96.3</td>
<td>94.5</td>
<td>94.9</td>
<td>94.5</td>
</tr>
</tbody>
</table>

$N_1 = 4$. The filter sizes for Stages 1 and 2 are 7-by-7 and 3-by-3, respectively.

5.2 Results of eye detection and face alignment

We use the JAFFE database to evaluate the proposed eye detection and face alignment method. As shown in Table 5.2, the detection rate of the proposed method is comparable to other existing methods, tested on the same database. The proposed eye detection method achieves 100% accuracy which is higher than other existing methods.

Table 5.2: Eye detection performance on the JAFFE database.

<table>
<thead>
<tr>
<th>Method</th>
<th>Year</th>
<th>Detection Rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed method</td>
<td>2010</td>
<td>100.0</td>
</tr>
<tr>
<td>Wang et al. [101]</td>
<td>2008</td>
<td>99.1</td>
</tr>
<tr>
<td>Wang and Yin [102]</td>
<td>2005</td>
<td>95.8</td>
</tr>
<tr>
<td>Zhou and Geng [103]</td>
<td>2004</td>
<td>97.2</td>
</tr>
</tbody>
</table>

5.3 Results of facial expression recognition

In this section, performances of the proposed FER system using different classifiers are discussed.

5.3.1 FER result using linear classifiers

The classification performance is first evaluated using linear classifiers on the JAFFE database. The classification rates for different facial expressions are shown in Table 5.3. The entry (at row $r$, column $c$) is the percentage of facial expression
that is classified as facial expression \( c \). For example, 95.4\% of anger expressions are correctly classified as anger, whereas 1.3\% of anger expression are misclassified as disgust.

The classification rates for the seven facial expressions are: anger 95.4\%, disgust 90.0\%, fear 100.0\%, happiness 96.7\%, neutral 96.7\%, sadness 96.0\% and surprise 96.0\%. The system recognises fear, happiness and neutral expressions well. It can recognise anger, sadness and surprise expressions better than disgust expression. In fact, the system considers disgust expression as sadness or fear at an error rate of 6.6\% and 2.7\%, respectively. For other existing methods \([15, 104, 105]\), fear expression is the most difficult to recognise.

### 5.3.2 FER result using SVMs

We evaluate the classification performance when SVMs are used in Stage 3. The classification rates for this system are shown in Table 5.4 for different categories of facial expressions. In this table, the entry (at row \( r \), column \( c \)) is the percentage of facial expression \( r \) that is classified as facial expression \( c \). For example, 96.67\% of anger expressions are correctly classified as anger, whereas 3.33\% of anger expression are misclassified as sadness.

The classification rates for the seven facial expressions are: anger 96.67\%, disgust 96.55\%, fear 96.88\%, happiness 100.0\%, neutral 96.67\%, sadness 96.77\% and surprise 93.33\%. The system recognises happiness and neutral expressions well. It can recognise anger, disgust, and sadness expressions better than fear and surprise expressions.

To demonstrate the robustness of the proposed FER system, we evaluated the
Table 5.3: Classification rates for different facial expression categories (classifier using mirror image). The entry at (row \( r \), column \( c \)) is the percentage of facial expression \( r \) that is classified as facial expression \( c \).

<table>
<thead>
<tr>
<th></th>
<th>Anger</th>
<th>Disgust</th>
<th>Fear</th>
<th>Happiness</th>
<th>Neutral</th>
<th>Sadness</th>
<th>Surprise</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anger</td>
<td>95.4</td>
<td>1.3</td>
<td>0.0</td>
<td>0.0</td>
<td>3.3</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Digust</td>
<td>0.7</td>
<td>90.0</td>
<td>2.7</td>
<td>0.0</td>
<td>0.0</td>
<td>6.6</td>
<td>0.0</td>
</tr>
<tr>
<td>Fear</td>
<td>0.0</td>
<td>0.0</td>
<td>100.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Happiness</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>96.7</td>
<td>3.3</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Neutral</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>96.7</td>
<td>2.0</td>
<td>1.3</td>
<td>1.3</td>
</tr>
<tr>
<td>Sadness</td>
<td>0.0</td>
<td>0.7</td>
<td>0.0</td>
<td>3.3</td>
<td>0.0</td>
<td>96.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Surprise</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>3.3</td>
<td>0.7</td>
<td>0.0</td>
<td>96.0</td>
</tr>
</tbody>
</table>

Table 5.4: Classification rates for different facial expression categories using FER system with feature selection.

<table>
<thead>
<tr>
<th></th>
<th>Anger</th>
<th>Disgust</th>
<th>Fear</th>
<th>Happiness</th>
<th>Neutral</th>
<th>Sadness</th>
<th>Surprise</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anger</td>
<td>96.67</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>3.33</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Disgust</td>
<td>0.00</td>
<td>96.55</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>3.45</td>
<td>0.00</td>
</tr>
<tr>
<td>Fear</td>
<td>0.00</td>
<td>0.00</td>
<td>96.88</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>3.12</td>
</tr>
<tr>
<td>Happiness</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>100.0</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Neutral</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>96.67</td>
<td>3.33</td>
<td>0.00</td>
</tr>
<tr>
<td>Sadness</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>3.23</td>
<td>0.00</td>
<td>96.77</td>
<td>0.00</td>
</tr>
<tr>
<td>Surprise</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>6.67</td>
<td>0.00</td>
<td>0.00</td>
<td>93.33</td>
</tr>
</tbody>
</table>
5.3. Results of facial expression recognition

system on the MMI database that generates more than 150,000 images. Table 5.5 shows the evaluation results using the linear classifier and SVMs. The evaluation process also employs ten-fold cross validation method and the final result is the average of all the ten folds.

Table 5.5: Classification rates tested on MMI database.

<table>
<thead>
<tr>
<th>Method</th>
<th>CR(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSI filters + Linear classifier</td>
<td>98.17</td>
</tr>
<tr>
<td>DSI filters + SVM</td>
<td>99.50</td>
</tr>
</tbody>
</table>

* DSI filters: Directional Shunting Inhibition filters.

5.3.3 FER results of two-class classification

Experiments have been conducted to solve two-class classification problems, namely, anger versus neutral, disgust versus neutral, fear versus neutral, happiness versus neutral, sadness versus neutral, and surprise versus neutral. Table 5.6 outlines the results of all the two-class expression classification tasks using aligned face images. Results in Table 5.7 indicate that recognition performance of happiness vs. neutral increases when the face pattern is better aligned and cropped and in-plane rotation is corrected. Table 5.8 shows the confusion matrix for smiling and neutral recognition, when the face image is aligned. The overall recognition accuracy is 99.00%.

An automatic FER system (including face detection and alignment) is implemented for differentiating smiling and neutral facial expressions. Figure 5.3 gives a visual example of using the proposed automatic system to recognise facial expressions when multiple faces are contained in the input image. All faces in the input pattern image have been correctly detected and aligned, and the facial expressions have been accurately recognised.
Table 5.6: Classification rates for different two-class facial expressions.

<table>
<thead>
<tr>
<th>%</th>
<th>Anger</th>
<th>Disgust</th>
<th>Fear</th>
<th>Happiness</th>
<th>Sadness</th>
<th>Surprise</th>
</tr>
</thead>
<tbody>
<tr>
<td>Neutral</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>99.00</td>
<td>97.10</td>
<td>99.67</td>
</tr>
</tbody>
</table>

Table 5.7: Comparison of recognition accuracies for non-aligned and aligned faces on the JAFFE database.

<table>
<thead>
<tr>
<th>Method</th>
<th>CR (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-aligned faces (OpenCV)</td>
<td>98.67</td>
</tr>
<tr>
<td>Aligned faces (proposed)</td>
<td>99.00</td>
</tr>
</tbody>
</table>

Table 5.8: Confusion matrix for the two facial expressions. The entry at (row $r$, column $c$) is the percentage of facial expression $r$ that is classified as facial expression $c$.

<table>
<thead>
<tr>
<th>%</th>
<th>Smiling</th>
<th>Neutral</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smiling</td>
<td>98.00</td>
<td>2.00</td>
</tr>
<tr>
<td>Neutral</td>
<td>0.00</td>
<td>100.00</td>
</tr>
</tbody>
</table>

5.4 Results of feature selection

We applied two methods, described in the previous chapter, for feature selection. The classification rates are shown as follows.

- When all 560 features produced by Stage 2 were used, the CR was 96.2%.
- When feature selection method based on individual pdfs was used, the system achieved a CR of 96.2% using only 375 features.
- When feature selection method based on joint pdf was used, the system achieved a CR of 96.7% using only 373 features.

These results indicate that feature selection leads to better classification performance with significantly fewer features.

Figure 5.4 shows the face areas where the selected features are located. It seems features used for facial expression recognition are located near the cheek and the
5.4. Results of feature selection

Figure 5.3: A visual result of the automatic system: (a) original colour image, (b) grey scale image, (c) eye detection, (d) expression recognition.

area between the two eyes. Surprisingly, the mouth area plays a less significant role in the proposed FER system.

Figure 5.5 shows the classification rate on the test set versus the number of features that are selected via the training and validation set based on feature
5.4 Results of feature selection

Figure 5.4: Locations of selected features superimposed on a face image as yellow-red patches. The first four images correspond to features in the four directions, $\theta = 0^\circ, 45^\circ, 90^\circ, \text{ and } 135^\circ$. The last image shows the selected features, combined from all directions.

![Figure 5.4](image)

Figure 5.5: System performance using feature selection method one. Selection method-1. It uses individual pdf to compute feature mutual information. Figure 5.6 shows the system performance using feature selection method-2 that uses joint pdf to generate mutual information. The red horizontal line indicates the performance when all features are used (CR = 96.2%). These figures demonstrate that feature selection based on joint pdf is better than feature selection based on individual pdf.

![Figure 5.5](image)
5.5. Comparison with other FER methods

Table 5.9 shows the classification rates of several existing FER methods, tested on the JAFFE database using ten-fold validation. Guo and Dyer [54] compared several feature selection schemes: using all features, feature selection via linear programming (FSLP), feature selection via adaptive boosting (AdaBoost). Busiu et al. [106] used Gabor wavelets to extract image features and the linear SVM as a classifier. Zhang et al. [104] used 34 manually defined fiducial points for feature extraction, and two-layer feed-forward neural network for classification. Koutlas and Fotiadis [105] used 20 automatically defined fiducial points and feed-forward neural networks (MLP). The proposed system, which used DSI filters, SVM classifiers and feature selection based on joint pdf, achieved a classification rate of 96.7%. It performed better than the seven existing FER methods.
Table 5.9: Classification rates of FER methods on the JAFFE database.

<table>
<thead>
<tr>
<th>Method</th>
<th>CR (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSI filters + SVM + FS method 2</td>
<td>96.7</td>
</tr>
<tr>
<td>DSI filters + SVM + FS method 1</td>
<td>96.2</td>
</tr>
<tr>
<td>DSI filters + SVM + all features</td>
<td>96.2</td>
</tr>
<tr>
<td>DSI filters + Linear classifier</td>
<td>95.3</td>
</tr>
<tr>
<td>Gabor + Linear SVM [106]</td>
<td>95.2</td>
</tr>
<tr>
<td>Fiducial points + FSLP [54]</td>
<td>91.0</td>
</tr>
<tr>
<td>Gabor + MLP [105]</td>
<td>90.2</td>
</tr>
<tr>
<td>Fiducial points + two-layer MLP [104]</td>
<td>90.1</td>
</tr>
<tr>
<td>LBP + Coarse-to-Fine [20]</td>
<td>77.0</td>
</tr>
<tr>
<td>Fiducial points + AdaBoost [54]</td>
<td>71.9</td>
</tr>
<tr>
<td>Fiducial points + Bayes rule [54]</td>
<td>71.0</td>
</tr>
</tbody>
</table>

* DSI filter: Directional Shunting Inhibition filter.

5.6 Chapter summary

In this chapter, we presented the experimental results of the proposed FER system tested on the JAFFE and MMI databases. The results of both multi-class classification and two-class classification were discussed. We compared our FER system with several existing FER methods tested on the same database, and our system achieved a better performance.
Automatic facial expression recognition is an active research area in facilitating human-computer interactions. In this thesis, we proposed an automatic FER system that is based on fixed, directional filters and adaptive filters arranged in a hierarchical structure. Different classifiers have been integrated to improve the system performance. The proposed system has been tested on the standard benchmarks, JAFFE and MMI databases, which are widely used for evaluating FER systems. This chapter is organised as follows: Section 6.1 summarises the research contributions, which have been discussed in this thesis; Section 6.2 presents future research directions; Section 6.3 provides concluding remarks.

6.1 Research summary

In this section, we summarise the thesis in terms of research activities and contributions.
6.1. Research summary

- We provided a comprehensive review of the state-of-the-art technologies of FER system. Four types of facial feature extraction method have been analysed, which are based on geometric features, appearance features, fusion features, and spatio-temporal features, respectively. To reduce the amount of features used for classification, several methods had been introduced, such as feature selection and dimensionality reduction. We also discussed the commonly used approaches for facial expression classification.

- We developed an eye detection and face alignment method. To detect the eye positions, we first used the OpenCV Adaboost-based method to detect faces and then introduced an eye-filter-based method to extract eye candidates. This eye-filter consisted of two types of filter: Gabor filter and circular filter. A template matching technique was employed to select the accurate eye positions among all the candidates. Then the face image was aligned using a geometric face model created based on the eye positions. This method achieved 100% eye detection accuracy, which is higher than several existing methods tested on the JAFFE database.

- We proposed a novel FER system. This system used fixed, directional filters and adaptive filters arranged in a hierarchical structure to extract facial features. The linear classifiers and SVMs were implemented for facial expression classification. The proposed method was evaluated on the JAFFE and MMI databases and achieved classification rates of 96.2% and 98.7%, respectively.

- To train the adaptive filters, we utilised two training methods, namely, the
LM algorithm and Rprop algorithm. The LM algorithm was used to train the JAFFE database and the Rprop algorithm could cope with large databases, such as the MMI database.

- The feature selection technique had been integrated into the system to improve the FER accuracy. We selected salient features based on the mutual information between each feature pair. Two methods were proposed to generate the mutual information. System performance tested on the JAFFE database was increased to 96.7%.

- We outlined the experimental results and compared them with the results generated by several existing FER systems. We used the JAFFE database to evaluate the FER system with different classifiers. Experimental results demonstrated that the classification rates derived using SVMs are better than that generated using linear classifiers.

6.2 Future work

Possible research directions can be summarised as follows:

- Train the SVM in the classification stage simultaneously with the adaptive filters to reduce the computational expenses.

- Develop new feature selection method to cope with large databases and improve the overall system efficiency.

- Apply the proposed FER method on video sequences to evaluate its real-time performance and improve recognition accuracy.
Combine with processing audio to improve the emotion recognition accuracy.

6.3 Conclusion

In this thesis, the automatic eye detection and face alignment approach is developed, which uses an eye filter combining with a template matching approach. The eye filter consists of a Gabor filter and a circular filter which jointly can generate possible eye candidates. The template matching approach uses a face template to select the accurate eye points which are then used to rotate the face image to the upright position based on the facial geometric characteristics.

For feature extraction and classification, we use the fixed, directional filters to extract primitive edge features in Stage 1, whereas, in Stage 2, we train the adaptive filters to extract more complex features, which is classified by different classifiers. To demonstrate the robustness of the extracted features, we first use linear classifiers to conduct expression classification on the JAFFE database and the classification accuracy is 95.9%. To improve the system performance, we then utilise the SVMs to classify different facial expression categories. The classification rate is increased to 96.2% which is comparable to the existing FER methods. Experimental results evaluated on the MMI database using linear classifiers is 98.7% and 99.5% using SVMs.

Generally, the number of features extracted from Stage 2 is very large. To eliminate the irrelevant facial features, we introduce techniques to find essential facial features before conducting final classification. We use mutual information as a selecting criterion. Two methods have been implemented for calculating
the mutual information for each feature. With the selected facial features, the recognition rate of the JAFFE database is increased to 96.7%.
References


[28] F. Bourel, C. C. Chibelushi, and A. A. Low, “Robust facial expression recognition using a state-based model of spatially-localised facial dynamics,” in


References


[61] S. Muller, F. Wallhoff, F. Hulsken, and G. Rigoll, “Facial expression recog-


[68] G. A. Ramirez and O. Fuentes, “Face detection using combinations of clas-


[75] F. H. C. Tivive, A. Bouzerdoum, S. L. P. Phung, and K. M. Liftekharuddin,


[82] F. H. C. Tivive and A. Bouzerdoum, “Efficient training algorithms for a class
References


