



UNIVERSITY OF THE WITWATERSRAND

MASTERS THESIS

Out-Of-Plane Action Unit Recognition Using Recurrent Neural Networks

Author:

Christine TREWICK - 0608835F

Supervisor:

Dr. Hima VADAPALLI

May 20, 2015

Abstract

The face is a fundamental tool to assist in interpersonal communication and interaction between people. Humans use facial expressions to consciously or subconsciously express their emotional states, such as anger or surprise. As humans, we are able to easily identify changes in facial expressions even in complicated scenarios, but the task of facial expression recognition and analysis is complex and challenging to a computer. The automatic analysis of facial expressions by computers has applications in several scientific subjects such as psychology, neurology, pain assessment, lie detection, intelligent environments, psychiatry, and emotion and paralinguistic communication. We look at methods of facial expression recognition, and in particular, the recognition of Facial Action Coding System's (FACS) Action Units (AUs). Movements of individual muscles on the face are encoded by FACS from slightly different, instant changes in facial appearance. Contractions of specific facial muscles are related to a set of units called AUs. We make use of Speeded Up Robust Features (SURF) to extract keypoints from the face and use the SURF descriptors to create feature vectors. SURF provides smaller sized feature vectors than other commonly used feature extraction techniques. SURF is comparable to or outperforms other methods with respect to distinctiveness, robustness, and repeatability. It is also much faster than other feature detectors and descriptors. The SURF descriptor is scale and rotation invariant and is unaffected by small viewpoint changes or illumination changes. We use the SURF feature vectors to train a recurrent neural network (RNN) to recognize AUs from the Cohn-Kanade database. An RNN is able to handle temporal data received from image sequences in which an AU or combination of AUs are shown to develop from a neutral face. We are recognizing AUs as they provide a more fine-grained means of measurement that is independent of age, ethnicity, gender and different expression appearance. In addition to recognizing FACS AUs from the Cohn-Kanade database, we use our trained RNNs to recognize the development of pain in human subjects. We make use of the UNBC-McMaster pain database which contains image sequences of people experiencing pain. In some cases, the pain results in their face moving out-of-plane or some degree of in-plane movement. The temporal processing ability of RNNs can assist in classifying AUs where the face is occluded and not facing frontally for some part of the sequence. Results are promising when tested on the Cohn-Kanade database. We see higher overall recognition rates for upper face AUs than lower face AUs. Since keypoints are globally extracted from the face in our system, local feature extraction could provide improved recognition results in future work. We also see satisfactory recognition results when tested on samples with out-of-plane head movement, showing the temporal processing ability of RNNs.

Declaration

I declare that this dissertation is my own, unaided work. It is being submitted for the degree of Master of Science at the University of the Witwatersrand, Johannesburg. It has not been submitted for any degree or examination at any other University.



(Signature of candidate)

20 day of MAY 2015

Acknowledgements

I would firstly like to thank my supervisor, Dr Hima Vadapalli for her constant support, knowledge and assistance over the last two years.

I would also like to thank the School of Computer Science as a whole for providing me with the knowledge and guidance required to allow me to get to the point where I am today. There have been several academic staff over the course of the years who have inspired me and continuously pushed me to work harder and aim higher.

Lastly, I would like to thank my family for their support and assistance to allow me the opportunity to complete my tertiary education.

Publications

Some ideas and figures have appeared previously in the publication listed below.

Title: Temporal Classification of FACS AUs using SURF Descriptors

Authors: Christine Trewick, Hima Vadapalli

Conference: Pattern Recognition Association of South Africa (PRASA) / RobMech 2014

Date: 27 - 28 November 2014

Status: Accepted

Contents

Abstract	i
1 Introduction	1
1.1 Research Hypothesis	2
1.1.1 Research Questions	2
2 The Facial Action Coding System	3
2.1 Introduction	3
2.2 Action Units	3
2.3 Scoring	4
2.4 FACS AUs and Pain Classification	4
2.5 Databases	5
2.5.1 FACS AU Encoded Databases	5
2.5.2 Emotion Labelled Databases	6
2.6 Conclusion	7
3 Literature Review	9
3.1 Introduction	9
3.2 Facial Expression Recognition Overview	9
3.2.1 Input Data	10
3.2.2 Face Detection	10
3.2.3 Feature Extraction	10
3.2.4 Expression Classification	10
3.2.5 Challenges Experienced in Facial Expression Recognition	10
3.2.6 Applications of Facial Expression Recognition Systems	11
3.3 Feature Extraction	11
3.4 Appearance Based Feature Extraction	12
3.4.1 Principle Component Analysis	12
3.4.2 Independent Component Analysis	13
3.4.3 Linear Discriminant Analysis	14
3.4.4 Gabor Filters	14
3.4.5 Two-Dimensional PCA	15
3.4.6 Haar Filters	16
3.4.7 AdaBoost	16
3.5 Geometric Based Feature Extraction	18
3.6 Comparison Of Feature Extraction Methods	18
3.7 Feature Classification	19
3.8 Spatial Approach	19
3.8.1 Neural Networks	19
3.8.2 Support Vector Machines	20
3.9 Spatial-Temporal Approach	21

3.9.1	Hidden Markov Model	21
3.9.2	Recurrent Neural Networks	21
3.10	Feature Detectors and Descriptors	22
3.10.1	Scale Invariant Feature Transform	22
3.10.2	Speeded-Up Robust Features	23
3.11	Out-Of-Plane Head Movement Techniques	23
3.11.1	Out-of-plane Face Detection	23
3.11.2	Head Pose Detection	24
3.11.3	3D Face Modelling Using Multiple 2D Images	25
3.11.4	Alternate Angle Face View Generation	25
3.12	Conclusion	26
4	Speeded Up Robust Features	27
4.1	Introduction	27
4.2	Keypoint Detectors	27
4.3	Keypoint Descriptors	28
4.4	Systems Using SURF	29
4.4.1	JAFFE Database	29
4.4.2	FERET Database	30
4.4.3	SURF and SVMs	30
4.4.4	SURF and PCA	31
4.5	Conclusion	31
5	Recurrent Neural Networks	32
5.1	Introduction	32
5.2	Feedforward Neural Networks	32
5.3	First-order Recurrent Neural Networks	33
5.3.1	Elman Recurrent Neural Network	34
5.3.2	Jordan Recurrent Neural Network	34
5.4	Backpropogation	35
5.5	Backpropogation Through Time	35
5.6	RNN Architecture	37
5.7	Applications	39
5.8	Conclusion	40
6	Methodology	41
6.1	Introduction	41
6.2	Research Hypothesis	41
6.2.1	Research Questions	41
6.3	Assumptions	42
6.4	Databases	42
6.5	Methodology Overview	42
6.6	Obtaining Feature Vectors - Cohn Kanade Database	43
6.7	Recurrent Neural Network Structure	43
6.8	Optimal Number of Keypoints	47
6.9	Feature Vectors - UNBC-McMaster Database	47
6.10	Challenges	48
6.11	Conculsion	50

7	Results	51
7.1	Introduction	51
7.2	Correlation - Cohn Kanade Database	51
7.3	Receiver Operating Characteristic Curves	52
7.4	Results of Phase I Testing (Cohn-Kanade database)	52
7.4.1	Results for AU1	53
7.4.2	Results for AU2	56
7.4.3	Results for AU4	57
7.4.4	Results for AU5	57
7.4.5	Results for AU6	61
7.4.6	Results for AU7	61
7.4.7	Results for AU9	64
7.4.8	Results for AU15	67
7.4.9	Results for AU17	67
7.4.10	Results for AU20	68
7.4.11	Results for AU25	71
7.4.12	Results for AU27	71
7.5	Results of Phase II Testing (UNBC McMaster Pain Database)	71
7.5.1	Results for AU4	77
7.5.2	Results for AU6	78
7.5.3	Results for AU7	78
7.5.4	Results for AU9	80
7.6	Comparison With Other Methods	80
7.7	Conclusion	81
8	Conclusion	83
9	Future Work	85
	References	85
A	FACS AUs and their descriptions	97
B	Detailed Results	100

List of Tables

7.1	Correlation between AUs in the Cohn Kanade Database.	52
7.2	Sample set sizes - Cohn Kanade database.	53
7.3	Results of Phase I Testing.	55
7.4	Correlation between AUs in the UNBC-McMaster database.	74
7.5	UNBC-McMaster sample set.	74
7.6	AU intensity sample numbers.	77
7.7	Results of Phase II Testing.	77
7.8	Correctly identified samples for each intensity.	77
7.9	Positive output means.	77
7.10	Positive output means - correctly classified samples.	77
7.11	Comparison with other methods	81
A.1	Lower face AUs.	98
A.2	Upper face AUs.	98
A.3	Head Positions.	98
A.4	Eye Positions.	99
A.5	Lip Parting and Jaw Opening.	99
A.6	Miscellaneous.	99
B.1	Results for AU1 - Cohn Kanade database	101
B.2	Results for AU2 - Cohn Kanade database	101
B.3	Results for AU4 - Cohn Kanade database	102
B.4	Results for AU5 - Cohn Kanade database	102
B.5	Results for AU6 - Cohn Kanade database	103
B.6	Results for AU7 - Cohn Kanade database	103
B.7	Results for AU9 - Cohn Kanade database	104
B.8	Results for AU15 - Cohn Kanade database	104
B.9	Results for AU17 - Cohn Kanade database	105
B.10	Results for AU20 - Cohn Kanade database	105
B.11	Results for AU25 - Cohn Kanade database	106
B.12	Results for AU27 - Cohn Kanade database	106
B.13	Results for AU4 - UNBC McMaster database	107
B.14	Results for AU4 - UNBC McMaster database	107
B.15	Results for AU7 - UNBC McMaster database	108
B.16	Results for AU9 - UNBC McMaster database	108

List of Figures

2.1	Range of AU intensity scoring.	4
2.2	Examples of extracts from some of the sequences from the UNBC-McMaster database.	6
3.1	Facial expression recognition process overview.	9
3.2	17
5.1	Elman RNN	35
5.2	BPTT with the network unfolded to a depth of $k = 3$	37
6.1	Image sequence showing a neutral face in the first frame and the development of AUs 6, 7, and 12 in the final frame with 60 keypoints extracted in each frame.	44
6.2	Learning rate of 0.1.	46
6.3	Learning rate of 0.01.	46
6.4	True positive rates with differing numbers of extracted keypoints.	47
6.5	False positive rates with differing numbers of extracted keypoints.	48
6.6	Image sequence showing the onset of AUs 4c, 6a, 7d, 43 and 50, and a PSPI score of 8, and with 60 keypoints extracted in each frame.	49
7.1	Results of AU1 testing	54
7.2	Neutral face (left) and face activating 1+4+7+11+20+25 (right).	56
7.3	Neutral face (left) and face activating 1+4+6+12+20+25 (right).	56
7.4	Neutral face (left) and face activating 1+2+5+25+27 (right).	57
7.5	Results of AU2 testing.	58
7.6	Results of AU4 testing.	59
7.7	Results of AU5 testing.	60
7.8	Results of AU6 testing.	62
7.9	Results of AU7 testing.	63
7.10	Neutral face (left) and face activating 4+7+9+17 (right).	64
7.11	Neutral face (left) and face activating 7+15+17+20+25+43 (right).	64
7.12	Results of AU9 testing.	65
7.13	Results of AU15 testing.	66
7.14	Neutral face (left) and face activating 1+4+7+15+17 (right).	67
7.15	Neutral face (left) and face activating 1+2+15+17 (right).	68
7.16	Results of AU17 testing.	69
7.17	Results of AU20 testing.	70
7.18	Results of AU25 testing.	72
7.19	Results of AU27 testing.	73
7.20	Results of AU4 testing.	75
7.21	Results of AU6 testing.	75
7.22	Results of AU7 testing.	76
7.23	Results of AU9 testing.	76
7.24	Extract showing 4d+6a+7e+10a+26a+43.	79
7.25	Extract showing 4d+6c+7c+9c+12c+43.	79

7.26	4+7+9+17 from Cohn Kanade database.	80
7.27	6c+9b+12b from UNBC-McMaster database.	80
A.1	Some of the AUs represented in the Cohn-Kanade database.	97

Chapter 1

Introduction

Facial expressions are the result of changes and deformations on a human face due to muscle movement. These are considered facial changes to portray a human's intentions, feelings, emotional states, or to accompany communication. Humans use their face in conjunction with facial expressions to consciously or subconsciously express their emotional states, such as sadness or surprise, and to exchange thoughts with or without words. Darwin looked at the facial expressions of humans and their genetically determined aspects of behaviour in his 1872 book, *The Expressions of the Emotions in Man and Animals*, and since then facial expression recognition and analysis has been an active and important research topic across many different scientific subjects [Darwin 1872]. Suwa *et al.* [1978] presented a preliminary research on automatically analyzing facial expressions. This was considered the first step towards automatic facial expression recognition. They accomplished this by tracking the motion of twenty identified spots on image sequences. Since then, significant progress has been made in the research and development of systems to recognize and analyze facial expressions. Facial expression recognition (FER) systems attempt to automatically recognize and analyze facial movement and changes from visual information in the form of static, single images or video sequences.

The recognition and analysis of facial expressions by computing systems has applications in many different and diverse scientific subjects. These include psychology, neurology, pain assessment, lie detection, intelligent environments, psychiatry, and emotion and paralinguistic communication. It also has applications in everyday life such as driver monitoring systems [Brandt *et al.* 2004] [Ji and Yang 2002], and automated tutoring systems [Whitehill *et al.* 2008]. As humans, the task of identifying changes in facial expressions comes naturally and easily to us, even in complicated situations. However, for a computer, this task can be challenging and computationally expensive. These automatic FER systems must share the same robustness, accuracy and speed as a human so that these systems can be used in a real-world scenario, such as in an airport security application.

The Facial Action Coding System (FACS) is based on a system originally developed by Varl-Herman Hjortsjo, a Swedish anatomist [Hjortsjö 1969]. It was adopted and adapted by Ekman and Friesen [1978] and later joined by Hager to publish an updated version in 2002 [Ekman *et al.* 2002]. The objective of FACS is to measure and categorize facial movements of a human by their appearance on the face. Contractions of specific facial muscles are related to a set of units called Action Units (AUs). FACS makes use of AUs and combinations of AUs to describe facial expressions.

Speeded Up Robust Features (SURF) is a local feature detector and descriptor first presented by Bay *et al.* [2006]. It was partly inspired by another feature detector and descriptor method, the Scale-Invariant Feature Transform (SIFT) [Lowe 1999]. SURF detects keypoints (or interest points) on the face, and then facial feature vectors are generated from keypoint descriptors. SURF provides smaller sized feature vectors than other commonly used feature extraction techniques. SURF is comparable to or outperforms other methods with respect to distinctiveness, robustness, and repeatability [Bay *et al.* 2008]. It is also much faster than other feature detectors and descriptors. The SURF descriptor is scale and rotation invariant and is unaffected by small viewpoint changes or illumination changes

We introduce a system to recognize FACS AUs on the face of a subject using SURF descriptors to de-

scribe facial features. A recurrent neural network (RNN) is used to classify features, which incorporates temporal data to classify features in a sequence of images. The use of RNNs has been tested by Vadapalli [2014], who made use of RNNs to recognize FACS AUs. The system used Gabor Filters to extract features. However, the use of Gabor filters can suffer from the high dimensionality problem. We propose the use of SURF to extract keypoints from the face in a sequence of images where an AU (or combination of AUs) are formed from a neutral face. We use these keypoints to generate SURF descriptors which form feature vectors. The feature vector obtained from SURF is far smaller than feature vectors obtained from Gabor filters and other common feature extraction techniques. An RNN is a spatial-temporal approach of classification. The RNN allows us to take advantage of temporal data received from image sequences, which depict a neutral face to the formation of an AU or AUs. Although Vadapalli [2014] were successful in using an RNN to classify AUs in image sequences, the use of SURF and RNNs to classify AUs where the face moves out-of-plane during the sequence has not yet been tested. Therefore, once the RNNs are trained to recognize AUs using SURF feature vectors, we recognize pain in human subjects from video sequences where the face moves out-of-plane for some part of the sequence. The presence of pain is considered to be the combination of AUs 4, 6 or 7, 9 or 10, and 43.

1.1 Research Hypothesis

The following research hypothesis is reached:

1. **RNNs will be able to classify FACS AUs in an image sequence using SURF to extract features and SURF descriptors as feature vectors.**
2. **SURF descriptors and RNNs can be used to recognize FACS AUs in an image sequence where the face moves out-of-plane by taking advantage of the temporal processing abilities of RNNs.**

1.1.1 Research Questions

1. Can SURF descriptors extract enough information from the face to recognize AUs?
2. Is it possible to train a recurrent neural network to recognize AUs using the SURF feature vectors?
3. How many keypoints need to be extracted from the face in order to optimally recognize AUs?
4. Can SURF descriptors and RNNs be used to recognize FACS AUs when the face moves out-of-plane?

Chapter 2

The Facial Action Coding System

2.1 Introduction

Interaction between humans often does not only consist of the generation of a particular facial expression to communicate a message. Rather, it could be a subtle change in one of the face regions that conveys the message and the related expression. The Facial Action Coding System (FACS) is based on a system originally developed by Val-Herman Hjortsjo, a Swedish anatomist [Hjortsjö 1969]. FACS measures and categorizes facial movements by the appearance changes they produce on the human face. It was then adopted by Ekman and Friesen [1978] and later joined by Joseph C. Hager to publish a significantly updated version of FACS in 2002 [Ekman *et al.* 2002]. FACS encodes movements of individual muscles of the face from the slightly different instantaneous changes in facial appearance they produce. It has become a common and widely used benchmark to categorize the physical expression of emotions, and it has proven useful to psychiatrists [Heller and Haynal 2002], psychologists [Keltner and Bonanno 1997] [Ekman and Rosenberg 1997], doctors [Rosenberg *et al.* 2001], animators and lie detectors.

2.2 Action Units

Contractions of specific facial muscles are related to a set of units called Action Units (AUs). The FACS makes use of AUs to describe facial expressions. Instead of using muscle movement, FACS uses AUs as a measurement unit. This is due to two reasons. Firstly, for some appearances, one AU consists of more than one muscle. This is because the appearance changes these muscles produce cannot be separated or distinguished. Secondly, one muscle can produce appearance changes separated into two or more AUs. Different parts of a muscle can represent relatively independent actions. Therefore, the mapping between facial muscles and AUs is not necessarily one-to-one. Some AUs are represented by more than one muscle, and some AUs represent separate and distinct movements of the same muscle.

The AUs are grouped based on the location and/or the type of action involved. Two major groups are the upper face AUs and the lower face AUs. Each AU has a unique number. Originally there were 44 AUs in the FACS with numbers ranging from 1 to 46, where numbers 3 and 40 were omitted [Ekman and Friesen 1978]. In the updated version of the FACS, an additional twelve AUs are added [Ekman *et al.* 2002]. These additional AUs include movements of the eyeball and of the head. The additional AUs are numbered from 51 upward. The full list of FACS AUs is available in Appendix A. The number of each AU as well as a short description of the AU is given. The AUs are grouped by location on the face or type of movement. Figure A.1 shows images of the AUs which we will be using for the purposes of this research.

2.3 Scoring

Describing facial movements using FACS is called scoring. FACS includes a system to describe the strength of the action of AUs, which result in variations in the intensity of the appearance change. Multiple FACS codes which are present in a particular instance are strung together using a plus (+) sign. The intensity of each AU is added after the letter, but before the plus sign. Figure 2.1 displays the range of AU intensity scoring. For example, 9a is the trace action of the nose wrinkler. For an intensity of E (maximum intensity), the letter is often omitted. For example, 1+2 raises the inner and outer portions of the eyebrow at maximum intensity. This equates to raising the entire eyebrow.

Activation of AUs result in changes in facial appearance. For example, raising the eyebrow can produce horizontal wrinkles on the forehead. These are considered additional indicators that AUs are active, and FACS scorers use these indicators to determine which AUs are activated in a particular scene. Scoring is a challenging task. Since the muscles of the face are covered by skin, they cannot be directly seen. This is why these additional indicators are important in both FACS scoring and automatic facial analysis. Another challenge is that several AUs can produce similar appearance changes when activated. For example, AU6, AU7 and AU12 can all result in the formation of wrinkles at the outer corners of each eye. Additionally, the appearance change of one AU can effect or even hide the appearance change caused by another AU. For example, activating AU4, which causes furrowing of the eyebrows, can reduce the visible amount of sclera of the eyes. When activating AU5, which raises the upper eyelid, the eye is widened and the amount of visible sclera is increased. When AU4 and AU5 are performed together (4+5), they work opposingly. Therefore, when AU4 causes the furrowing of the eyebrows, it can potentially reduce the amount of sclera shown by raising the upper eyelids of AU5, and has the possibility of making it appear that the upper eyelid was not raised when it actually was. Because of these factors, a FACS scorer must know the AUs associated with each muscle movement, how each AU affects facial appearance, and how the appearance changes of each AU affect each other. For automatic facial expression recognition and analysis, these changes and how they affect each other can also be seen as a challenging task. For AU recognition by a computing system, these factors play a vital role in the design and development of systems.

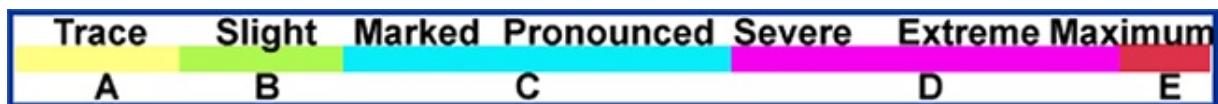


Figure 2.1: Range of AU intensity scoring.

2.4 FACS AUs and Pain Classification

FACS has also had applications in the classification of pain in human subjects. Prkachin [1992] found that four actions, represented by six AUs, carried most of the information about pain on a human face. These four actions are brow lowering (AU4), orbital tightening (AU6 and AU7), levator contraction (AU9 and AU10) and eye closure (AU43). Thereafter, it was confirmed and further researched by Prkachin and Solomon [2008]. They developed a pain intensity scoring system called The Prkachin and Solomon Pain Intensity metric, also known as the PSPI metric. This system is the sum of intensities of the four actions. The PSPI metric is defined as:

$$\text{Pain} = \text{AU4} + (\text{AU6 or AU7}) + (\text{AU9 or AU10}) + \text{AU43} \quad (2.1)$$

This equation shows that the sum of AU4, the highest in intensity of AU6 or AU7, the highest in intensity of AU9 or AU10, and AU43 can be yielded for a sixteen point pain scale. For the first image sequence in Figure 2.2, which is taken from the UNBC-McMaster pain database (discussed below), the peak frame here (frame 60) has been FACS coded as 6c+9b+43. AU6 is at intensity C, which gives a score of three. AU9 is at intensity B, which is a score of two. AU43 is active, which gives a score of one. This gives

a total PSPI score of six (3+2+1). AU43 score is not dependent on its intensity, and is considered a score of one if AU43 is active and a score of zero if AU43 is not active. Similarly, in the second image sequence in Figure 2.2, the peak frame has been FACS coded as 4a+6d+7d+12d+43. This gives a score of $1+4+1=6$. AU4 has an intensity of A, therefore a score of one. AU6 and AU7 both have intensity D, which is a score of four for both. The maximum of four is taken. AU43 is active and therefore has a score of one. For a PSPI score of zero, there is no pain present. For a PSPI score of greater than or equal to three, strong pain is considered to be present.

2.5 Databases

There are several databases which have been used in the field of automated facial expression analysis. When developing facial expression recognition systems, the expressions recognized generally cluster into two categories. The first group concentrates on recognizing the six basic emotions which are: anger, disgust, surprise, happiness, fear, and sadness. These six emotions were described by Ekman and Friesen [1971]. The second group concentrates on a more fine-grained means of measurement and description of facial expressions. These fine-grained means of measurement are usually based on FACS and the facial muscle movements caused by AUs. Because of this, facial databases and how they are encoded also tend to cluster into one of these two groups. Additionally, facial databases tend to either contain single, static images showing expressions, or image sequences (or video) showing the development of expressions. An important factor to consider when determining which database to use for facial expression recognition analysis is how the facial actions are elicited for data collection. Some databases contain images or image sequences of professional actors displaying desired behaviour, while other databases contain natural expression elicitation or by asking subjects to perform the desired actions. In databases created using actors or where the subjects are requested to perform an action, the timing of facial actions and their appearance may differ from spontaneously occurring behaviour. We discuss the two types of facial databases below.

2.5.1 FACS AU Encoded Databases

Cohn-Kanade

The Cohn-Kanade Database is a publically available facial expression database from Carnegie Mellon University [Kanade *et al.* 2000]. It contains AU coded image sequences from men and women of varying races, backgrounds, ethnicities and ages. This database contains image sequences from over one hundred subjects who perform a series of facial displays that include single AUs and combinations of AUs. The first frame of the image sequences show a neutral face and illustrate the development of AUs or combinations of AUs. The database was updated following the FACS updates in 2002 [Ekman *et al.* 2002]. In this database there is small head motion present and camera orientation is frontal. In forthcoming versions of the database, there will be some degree of out-of-plane head motion displayed. There are three variations in lighting. These are ambient lighting, single-high-intensity lamp, and dual high-intensity lamps with reflective umbrellas. Not only are the image sequences encoded using FACS, but they are also assigned emotion-specified labels. For example, 6+12 is considered happiness. The emotion expressions included in this database are happy, surprise, anger, disgust, fear, and sadness. The FACS is often used in FER systems for five reasons [FACS]:

- FACS can identify the slightest movements in facial musculature uncovering even the subtlest emotional reactions in subjects
- FACS can identify the exact emotion felt as well as the time frame including start/peak/end and duration of an emotional reaction
- FACS can identify all facially expressed emotional reactions across a given sequence or stimulus duration

- FACS can capture emotional reactions during both speaking and non-speaking occurrences
- FACS is a non-invasive tool allowing for subjects to remain unaware/impervious to the behavioural analysis

UNBC-McMaster

The UNBC-McMaster Shoulder Pain Expression Archive Database was developed by researchers at McMaster University and University of Northern British Columbia [Lucey *et al.* 2011]. It consists of videos captured of the faces of participants who were suffering from shoulder pain. To elicit these pain responses, subjects were asked to perform several active and passive range-of-motion tests to both their affected and unaffected limbs. Each frame from each video of this data was FACS coded by certified coders. In addition to FACS coded labels for each frame of the videos, a PSPI score is also included to show the level of pain the subject is experiencing. Participants gave a self-report to validate pain level. A total of 129 subjects (66 female and 63 male) were used to develop this database. These subjects were self-identified as having problems with shoulder pain. Two hundred image sequences across 25 subjects have been made publically available. Unlike the Cohn-Kanade database, which contains frontal face images, the UNBC-McMaster database contains image sequences where the face moves out-of-plane due to the effect of pain on humans. Figure 2.2 shows examples of extracts from the UNBC-McMaster database. From these examples, we can see that this database contains some instances of significant head movement.



Figure 2.2: Examples of extracts from some of the sequences from the UNBC-McMaster database.

2.5.2 Emotion Labelled Databases

The following are some examples of facial databases which are alternatives to Cohn-Kanade or UNBC-McMaster for facial expression analysis. These databases do not make use of FACS AUs and are coded with emotion labels such as surprise, anger, or happiness.

University of Maryland Database

The University of Maryland database is a facial expression database containing image sequences of 40 subjects [Black and Yacoob 1997]. The subject are of diverse cultural and racial backgrounds. The

subjects were recorded while displaying facial expressions of their choice. The subjects were told that they should move their heads, but not enough to result in profile views. The resulting image sequences were manually coded. The database contains 70 image sequences showing a total of 145 instances of the six basic emotions. Each image sequence shows one to three expressions. Instances of the six emotions were not balanced. Disgust, happiness, anger, and surprise showed up far more frequently than fear and sadness.

The AR Database

The AR Database is a publically available database developed at the Ohio State University [Martinez 1998]. It contains 4000 colour images showing expressions such as neutral, smiling, anger, and screaming. The images come from 126 men and women who were facing frontally, but who were allowed facial occlusions such as hats, glasses, sunglasses, and scarves. In addition, no restrictions were given with respect to make up and hair style. The images were taken under differing illuminations.

Japanese Female Facial Expression Database

The Japanese Female Facial Expression (JAFFE) database [Kamachi *et al.* 1998] contains 213 images of ten female Japanese subjects. The images were elicited while subjects were facing a semi-reflective mirror. Each female subject was recorded multiple times while displaying a neutral face and the six basic emotions. The camera trigger and resulting images were controlled and determined by the subjects. The images were then rated by 60 Japanese women to give a score on a five-point scale for each of the six emotions. The images are distributed along with the rating results.

GEMEP-FERA

The GEMEP-FERA database [Bänziger and Scherer 2010] consists of audio-video emotion portrayal recordings of ten subjects displaying a range of expressions, while making sounds. These sounds are uttering a meaningless phrase, or the word “ah”. Videos are approximately two to five seconds in length. There are five emotion categories which are anger, fear, joy, relief and sadness.

2.6 Conclusion

This chapter provides an overview of the FACS, as well as a description of AUs that make up the FACS. FACS provides a system which describes the subtle changes that occur in the face region. The Cohn-Kanade database is introduced, which is a database that illustrates FACS AUs in over 100 subjects. This database contains image sequences of the formation of AUs from a neutral face while the face is frontal. So much emphasis has been placed on using FACS in research because of the link between emotions and how they are “universally” expressed on the face. FACS has played an integral part in the breakdown and identification of the various muscles and muscle combinations involved in each of the universal facial expressions. As such, FACS has also been an integral part of the research into deception detection, interpersonal communication, consumer engagement, psychological states such as pain, computer animation and more. We use FACS, and in particular, the Cohn Kanade database to train and test our system due to its level of detail provided when describing facial expressions. It has become one of the expression description benchmarks in the field of FER, and several FER systems have made use of FACS and the Cohn Kanade database. FACS is not only able to measure the standard expressions generated by humans (such as anger or surprise) , but can also measure the changes that are produced randomly on the face. This makes it flexible to not only recognize the standard facial expressions, but also to recognize other expressions in which emotion labelled databases fail. The subjects in the Cohn kanade database are male and female, and of different ages, ethnicity, and race, allowing for the samples to be varied and more representative than other commonly used databases. The UNBC-McMaster database is also introduced, which shows the effect of pain on a human face and is labelled with FACS AUs as

well as PSPI scores. Unlike the Cohn-Kanade database, this database contains image sequences in which the face moves out-of-plane or has some degree of in-plane movement. We therefore make use of this database as we can test our FACS AU trained system to determine if an RNN can recognize AUs where the face is not frontal. This chapter forms the framework on which we will base our recognition of AUs.

Chapter 3

Literature Review

3.1 Introduction

This chapter provides a literature review and an overview of the facial expression recognition procedure. The steps taken to recognize facial expressions are introduced, as well as the different approaches that can be taken in each step. Two steps in the procedure are detailed, which are feature extraction and feature classification. An overview of the different approaches that can be taken as well as a summary of the approaches are provided. We also look feature detection and extraction, and the different techniques that have been developed to accomplish this. Thereafter, we look at some ways in which out-of-plane head movement has been dealt with or overcome in other facial expression recognition systems.

3.2 Facial Expression Recognition Overview

Facial expression recognition (FER) by a computing system is a complex and challenging task because of the variations in faces due to age, ethnicity, gender, race, etc. The complexity is further increased by variations in lighting, angles, and background noise. In general, the steps taken to recognize facial expressions are as follows: receive input data, detect the face, extract the features, and classify the features. Figure 3.1 shows the steps taken in the FER process.

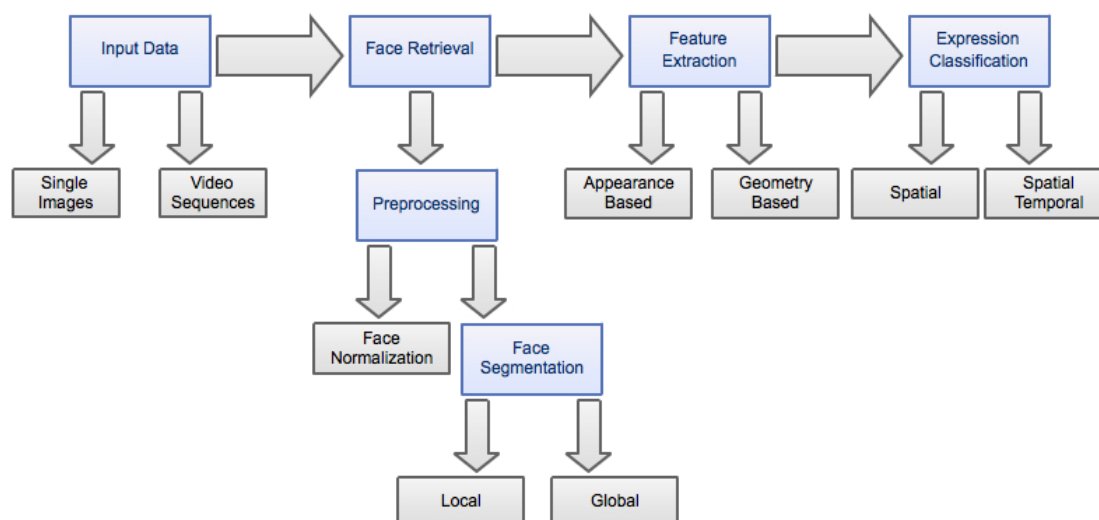


Figure 3.1: Facial expression recognition process overview.

3.2.1 Input Data

When capturing data for FER, the two main types of data are single static images, and video. A video is a sequence of images, and each image is known as a frame. Video sequences are more significant (and also more challenging to work with) as they provide temporal information from one frame to the next. Video sequences are usually evaluated frame-by-frame.

3.2.2 Face Detection

The face of the subject needs to be located in each image and be distinguished from the background. When the face is frontal, just the face region needs to be recognized. In the case of a rotated face which is out-of-plane, the head pose of the subject also needs to be recognized in some systems. There are two steps taken in the face detection process. These are face normalization and then face segmentation. Face normalization is required to standardize images. Thereafter, the face is segmented to provide data on a local or global level for expression classification.

3.2.3 Feature Extraction

The next step in the process is feature extraction. The two most commonly used methods in feature extraction process are an appearance based method, and a geometry based method. The appearance method looks at the appearance changes of the features on the face, such as furrows, dimples and wrinkles that appear when movement occurs on the face. Appearance based methods use the colour data of the image pixels to extract information about salient features to classify the facial expressions. These appearance features are either extracted from the whole face (known as global extraction) or on specific regions on the face such as the eye or mouth region (known as local extraction). Geometry based methods analyze the geometric relationships between fiducial points on the face. These fiducial points are considered key feature points of the face. Geometric features represent the geometric location and shape of facial elements such as the mouth, nose, eyes, and eyebrows. The extracted features in this step form feature vectors describing the face.

3.2.4 Expression Classification

Once features have been extracted, the next step is to recognize (classify) the face or facial expressions based on the extracted feature vectors. This step is known as expression classification. There are two main approaches taken when classifying facial expressions. These are the spatial approach and the spatial-temporal approach. The spatial approach uses only a single frame when classifying expressions. This can be a single, static image or a single frame taken from a video or image sequence. Expression classification of the image takes place with or without the use of a reference image (usually a neutral face image). The spatial-temporal approach uses the temporal information gained from image sequences (or video) to classify facial expressions of one or more images. This means that it takes into account previous frames when classifying the current frame.

3.2.5 Challenges Experienced in Facial Expression Recognition

One of the challenges faced when extracting features from a face is feature occlusion. This occurs when the presence of elements such as beards, glasses, hats or scars introduce a high level of variability. A face can be partially covered by objects, such as glasses, which makes extracting features and classifying the facial expressions more challenging.

Another challenge faced is pose variation. The ideal scenario for FER is when a face is facing frontally and only frontal images or video sequences are involved. This is very unlikely in the real world, and humans are naturally inclined to move their head as they express themselves. The performance of many algorithms used in FER is greatly deteriorated when there are large pose variations.

The quality of image or video sequence also plays a large role in the integrity of facial expressions recognized. Different imaging conditions, different types of cameras and technology, and varying types of lighting can play a role in image quality.

3.2.6 Applications of Facial Expression Recognition Systems

The field of robotics is advancing rapidly, and with the development of more and more sophisticated robots which interact with humans increasingly often and become more prominent in our every day lives, the need and urgency to develop a robust FER system is evident. Robots must become increasingly intelligent in order to understand our feelings and emotions and for them to respond correctly to these human emotions. FER systems can create an intelligent visual interface between man and computer.

Automating the analysis of facial expressions by computing systems has applications in many different and diverse scientific subjects. These include psychology, psychiatry, neurology, pain detection, lie detection, intelligent environments, and emotion and paralinguistic communication.

FER systems can also have applications in every day life. An example of this is a driver monitoring system which detects and analyzes the level of fatigue of a driver [Brandt *et al.* 2004] [Ji and Yang 2002]. It can also be applied to automated tutoring systems which determine the level of understanding or confusion of a student [Whitehill *et al.* 2008].

FER systems are also popular in the entertainment industry. Bartlett *et al.* [2003] used their FER system to develop an animated character that mimics the user's expressions. This is known as CU Animate. They also successfully developed the recognition system for Sony's Aibo Robot which was a robot dog that first appeared in 1999. This robot dog was one of the first examples of robotics designed for and targeted to regular consumers, and with an integrated camera for eyes, it would "interpret" what it saw. They also developed the recognition system for ATRs RoboVie which is an interactive humanoid robot [Ishiguro *et al.* 2001]. Anderson and McOwan [2006] developed the EmotiChat. This is a chat room application in which users log in and start chatting. There is an FER system connected to the chat application and it automatically (and in real-time) inserts emoticons which mimic the user's facial expressions.

An important application of FER systems is for security purposes. Previously, security systems concentrated on face recognition in places such as airports and high profile buildings. Recently, the emphasis has shifted to analyzing facial expressions too. Systems that recognize faces are often trained to detect neutral faces. However, a completely neutral face for the majority of the time is considered unrealistic in real life. FER systems can be used for security purposes where the face is shown to experience different expressions under differing conditions.

Affective computing is the research and development of systems that can recognize, interpret, process, and simulate human affects. It is an interdisciplinary field spanning computer science, psychology, psychiatry, and cognitive science [Tao and Tan 2005]. A computer should be able to understand and interpret the emotional state of humans and thus adapt its behaviour to result in an appropriate response for those emotions. FER plays an important role in recognizing and analyzing a human's affect and thus helps in building meaningful and responsive human-computer interaction interfaces.

3.3 Feature Extraction

Before feature extraction, the face must first be aligned and normalized to a standard face image. This can be done either manually or automatically. This is referred to as the preprocessing step. This step removes the effects of variations in facial motion, lighting, scale, and others. Then normalized facial feature measurements are obtained by using a reference image which is often a neutral face. Thereafter, features can be extracted in the feature extraction step. Feature extraction finds a representation of the face data that highlights the relevant information. The use of a particular feature extraction technique is to maximize performance and also depends on the classification method used, the structure of the data, the amount of noise and the ability of the extraction technique to handle the noise.

Extraction can either be global (also known as holistic) or local. The global approach treats the face as a whole and extracts features on the entire facial region without considering any special or prominent features. Global representation is the most commonly used approach in face or facial expression recognition. Feature extraction looks at the lexicographic ordering of pixel values, and for each image obtains one vector. This is the feature vector obtained for a single image, and for image sequence there is one feature vector per image. An image is considered to be a point in the high dimension feature space. The dimension depends on the pixel size of the image. For example, an image of 100x200 pixels is seen as a point in the 20 000 dimension feature space. This is considered an extremely high dimensionality. Learning cannot take place in such a high dimension feature space. This is known as the high dimensionality problem. To combat this, a dimension reduction technique can be used to deal with the high dimensionality problem. This will assist in taking the image into a lower dimension feature space, but will still ensure that the important types of variation of the data are preserved. The local approach treats different parts of the face as individual and the feature extraction process is applied to certain locations (such as mouth and eyes) on the image. These locations are called fiducial points. Some degree of invariance is required with respect to translation, scale, and rotation.

There are two methods commonly used for extracting features of the face. They are geometry based methods and appearance based methods. Geometric features represent the location and shape of facial elements. These facial elements, which include the mouth, eyes, eyebrows and nose, are extracted to form one feature vector per image to represent facial geometry. Essa and Pentland [1997] and Pantic and Rothkrantz [2000] make use of geometric feature extraction techniques. Appearance features are represented by the appearance changes on the face, such as wrinkles, dimples and furrows, when movement occurs on the face. Appearance based methods use the colour of the image pixels to extract salient features to classify the facial expressions. The appearance features can be extracted either locally or globally on face images. Bartlett *et al.* [2001], Littlewort *et al.* [2002] and Lyons *et al.* [1998] make use of appearance based feature extraction techniques. Some systems use a hybrid of the two approaches. Donato *et al.* [1999], Tian *et al.* [2001], Tian *et al.* [2002], Wen and Huang [2003], Zhang *et al.* [1998] make use of a hybrid system consisting of both geometric and appearance based techniques.

We will further detail appearance based methods and geometric based methods, as well as comparisons of the methods in terms of recognition accuracy rates.

3.4 Appearance Based Feature Extraction

Appearance methods use colour data in the pixels of the image to extract information about the facial features. Information about images is contained in their pixels, and the number of pixels in an image is important when obtaining this information. Many pixels contain information that is not relevant to FER. In some cases, pixel data stays the same from one facial expression to another, which makes analyzing these pixels redundant. Additionally, some pixels are completely dependent on their neighbours which make the extracted feature vectors irrelevant. Thus, these redundant pixels should be removed to improve performance and also reduce the dimensionality when extracting facial features from an image. Principle component analysis (PCA) and independent component analysis (ICA) are two of the most frequently used unsupervised methods to remove this redundant data and also decrease dimensionality.

Many algorithms and methods have been developed to extract features using an appearance based method. We will look at PCA, ICA, Linear discriminant analysis (LDA), Gabor filters, Haar filters and AdaBoost.

3.4.1 Principle Component Analysis

Principle Component Analysis (PCA) is an approach of feature extraction. It identifies patterns in data, and expresses the data such that similarities and differences within the data are highlighted.

PCA finds an orthogonal set of dimensions that account for the principal directions of variability in the data set. Principle components can be obtained by finding the eigenvectors of the pixelwise co-

variance matrix, S , of the δ -images, X . The eigenvectors can be obtained by decomposing S into the orthogonal matrix P and diagonal matrix D . Therefore $S = PDP^T$ [Donato *et al.* 1999]. Eigenfaces are a set of eigenvectors used in the FER process. The basis vectors (eigenfaces) computed by PCA are in the direction of the largest variance of the training vectors. The approach of using eigenfaces for recognition was developed by Sirovich and Kirby [1987] and later used by Turk and Pentland [1991]. A set of eigenfaces can be generated by performing PCA on a large set of images depicting different faces. It is considered one of the first successful examples of an FER system.

One of the goals of PCA is to reduce the number of dimensions of the feature space and to identify new, meaningful underlying variables, but to still keep principle features to minimize loss of information. The PCA method uses second-order statistics in the data. Since patterns in high dimension data can be hard to find, PCA is a powerful tool for analyzing data [Smith 2002]. When the original data is reproduced, the images have lost some of the information. The PCA compression technique is thus said to be lossy because the decompressed image is not exactly the same as the original. The data that is lost could be useful or relevant data and this is therefore the main disadvantage of using PCA.

Kirby and Sirovich [1990] applied PCA to face images. Their work showed that PCA is a compression scheme well suited to faces images, and that the mean squared error between the original images and the reconstructed images was minimized for all given levels of compression. They were among the first to use PCA as a feature extraction technique.

Padgett and Cottrell [1997] compared three representations of data based on PCA. These are eigenfaces, eigenfeatures, and local PCA. Local PCA produced the best recognition results of 75%. They also reported a performance of 70% when using eigenfaces for FER and a neural network for classification.

Bartlett *et al.* [1999] tested a global system with 50 principal components. The classification performance was tested for two scales of difference images. These are 66x96 and 22x32. Several quantities of principal components were evaluated in the network input. These are ten, 25, 50, 100, and 200. Increasing the amount of network input information and increasing the number of free parameters to be estimated results in a trade-off. The best performance of 89% was achieved using the first 50 principal components of the 22x32 difference images. Overparameterization can be a risk when working with such high dimensional networks. In addition, Bartlett *et al.* [1999] employed a region of interest analysis, which consisted of half of the face image. This was similar to the eigenfeature approach that gave Padgett and Cottrell [1997] better performance than the eigenvector approach.

3.4.2 Independent Component Analysis

Some of the most widely used and successful feature representations for FER, such as eigenfaces [Turk and Pentland 1991], holons [Cottrell and Metcalfe 1990], and local feature analysis [Penev and Atick 1996] are based on PCA. PCA only considers the second-order moments and thus it lacks information on higher-order statistics. In a task such as FER, important and relevant information can be kept in the high-order relationships amongst the image pixels. It can therefore be beneficial to consider not only the second-order relationships, but also the higher-order relationships between pixels. Independent Component Analysis (ICA) accounts for higher order statistics.

ICA is said to be more powerful than PCA with respect to data representation because it gives an independent rather than uncorrelated image representation and decompression [Karhunen *et al.* 1997]. ICA is considered unsupervised because it outputs a set of maximal independent component vectors. It is characterized by its ability to identify statistically independent components based on input distribution. When using ICA for feature extraction, output class information is included in addition to input features. Performing ICA on data provides N -dimension vectors showing the directions of independent components in the feature space. There is an additional input feature in the form of the output class, giving an $(N + 1)$ -dimension space. After applying ICA to this, $(N + 1)$ -dimension vectors are obtained. The main disadvantage of ICA is that it does not guarantee useful information because of its unsupervised learning nature.

Draper *et al.* [2003] compared PCA and ICA for the task of face recognition. PCA calculations used two different distance measures. They also implemented ICA with two types of architecture. PCA I

used a city-block distance metric. PCA II used a Euclidean distance metric. ICA I used statistically independent basis images where the images were variables and the pixels provided the observations for the variables. ICA II used statistically independent coefficients that represented the input images in the subspace defined by the basis images. For ICA II, input was transposed from ICA I, i.e. pixels were the variables and the images were observations. Results showed that the average recognition rates for facial actions were 88% for ICA I, 76% for ICA II, 79% for PCA I and 85% for PCA II. This shows the ICA using statistically independent basis images provides the best overall recognition rate.

3.4.3 Linear Discriminant Analysis

PCA only looks at training image variances to construct the subspace. In contrast to this, linear discriminant analysis (LDA) considers the training image class-membership information when finding a subspace. This is considered by some to be an improvement on PCA. LDA can be used as a linear classifier to reduce dimension since it finds a linear feature combination which characterizes two or more classes. PCA is often considered to be more appropriate for data compression, and LDA is more appropriate for classification purposes [Chen *et al.* 2000].

LDA maps high-dimension samples onto a low-dimensions space by finding the set of vectors which are of the most discriminant projection. The main disadvantage of using LDA is that it could potentially encounter the small sample size problem. Whenever the number of samples is smaller than the dimension of the samples, this problem will arise. This can cause the sample scatter matrix to become singular, which will result in computational difficulty when using LDA [Chen *et al.* 2000].

3.4.4 Gabor Filters

Gabor wavelets can be used to extract the facial appearance changes. A Gabor filter is a linear filter used for edge detection. The input image is filtered with a Gabor filter tuned to a particular orientation and frequency. The Gabor filter may be applied to the face image locally or globally. Gabor wavelets demonstrate two advantageous characteristics: orientation selectivity and spatial locality [Jain and Li 2005]. For the purposes of FER, multiple Gabor filters can be used to extract features where each feature is tuned to a characteristic frequency and orientation [Bhuiyan and Liu 2007]. An important factor to consider in the use of Gabor filters for FER is the choice of filter parameters. The combined response is called a Gabor jet.

A Gabor filter is considered a complex exponential modulated by a Gaussian function. The Gabor filter is defined as:

$$\psi(x, y, \varpi, \theta) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x'^2+y'^2)}{(2\sigma^2)}} [e^{i\varpi x'} - e^{-\frac{\varpi^2\sigma^2}{2}}] \quad (3.1)$$

$$x' = x \cos \theta + y \sin \theta \quad (3.2)$$

$$y' = x \sin \theta + y \cos \theta \quad (3.3)$$

where (x, y) is the pixel's position in the spatial domain, ϖ is the radial centre frequency, θ is the orientation of the Gabor filter, and σ is the standard deviation of the round Gaussian function along the x -axis and y -axis.

Bartlett *et al.* [2001] applied Gabor wavelets with eight orientations and at five spatial frequencies to different images. They applied the Gabor filters to the face globally. The Gabor output's dimension was too large to train the classifier, so they first used PCA to reduce the Gabor output to 100 dimensions per image. They used a representation where the outputs of two of the Gabor filters were squared and summed to provide robustness to image shift and lighting conditions.

Wen and Huang [2003] applied Gabor wavelets at two spatial frequencies and six orientations. Instead of applying the Gabor filters globally, they extracted appearance features at eleven face regions. This limited the effects of individual variation and noise. Thereafter, they used the feature points weighted average as each region's final feature vector. For each of the eleven regions, a twelve dimension feature vector was obtained.

Tian *et al.* [2002] looked at Gabor coefficients and geometric features for AU and AU combination recognition in image sequences. They used 480 Gabor coefficients, as used by Zhang *et al.* [1998], at twenty upper face locations. They also used 432 Gabor coefficients at eighteen locations on the lower face. It was found that the Gabor method works well for single AU recognition on similar subjects displaying no head motion. For recognition of combinations of AUs in dissimilar subjects displaying some small head motion, the results were poor. There are many factors which are said to account for the differing results. Firstly, studies from the past used only similar subjects. Zhang *et al.* [1998] used only Japanese subjects, Donato *et al.* [1999] used Euro-Americans, while Tian *et al.* [2002] used subjects of diverse ancestry such as European, African, and Asian. Secondly, studies from the past recognized emotion-coded expressions (such as anger or surprise) or only single AUs. Tian *et al.* [2002] tested the Gabor method on not only single AUs, but also AU combinations, and combinations where the occurrence of one AU changes the appearance of another. Thirdly, studies from the past manually cropped and aligned face images while Tian *et al.* [2002] did not crop or align face images at all.

Kotsia and Pitas [2007] developed a system to recognize facial expressions that are partially occluded. They used three methods of feature extraction: Gabor wavelets, a discriminant non-negative matrix factorization (DNMF) algorithm, and by geometric displacement vectors extracted using a Candidate tracker. As classifiers, they used multiclass SVMs and a multi-layer perceptron. This system was tested on both the Cohn Kanade database and the JAFFE database. Recognition accuracy when tested on the JAFFE database showed that Gabor wavelets received 88% accuracy and DNMF got 85% accuracy. When tested on the Cohn Kanade database, Gabor wavelets got 92% accuracy and DNMF got 87% accuracy.

High Dimensionality Problem

We have seen examples of applying Gabor filters tuned to x frequencies and y orientations. This results in xy phase and magnitude response maps which are of the same size as the image. For example, a Gabor filter applied to an image which is tuned to five frequencies and eight orientations results in 40 phase and magnitude response maps. Therefore, each pixel is now a 40 dimension feature vector describing the pixel's response of Gabor filtering. An over complete and highly localized response is obtained at each location on the image. Lades *et al.* [1993] combatted this by placing a coarse rectangular grid over the image, and then taking the response only at the nodes of the grid. This puts a limit on the dimension of the feature vectors. To describe the entire face image using Gabor features, some methods only look at the response at certain image locations, such as looking at only important facial landmarks.

The pixel responses can be concatenated into a global feature vector. However, a global image vector results in an incredibly high dimension problem. For example, if an image of size 200x200 is tuned to the Gabor filter described above, it would result in a $40 \times 200 \times 200 = 1\,600\,000$ dimension feature vector. This is prohibitively high and will result in a computationally expensive system. Kernel PCA can be used to reduce dimension, known as Gabor-KPCA [Liu 2004]. Some systems have selected only significant points by using a method such as Adaboost [Wu *et al.* 2002]. However, a global feature description in this case would still result in a high dimension feature vector. For example, Ma *et al.* [2002] selected 32 points in an image and applied a Gabor filter tuned to five frequencies and eight orientations. This still resulted in high dimensional vector of $32 \times 40 = 1280$ dimensions. This is referred to the high dimensionality problem, which the use of Gabor filters often suffer from.

Abdulrahman *et al.* [2014] proposed an approach for FER using a Gabor wavelet transform. They first used Gabor filters as a preprocessing step to extract the feature vectors. Thereafter, because of the high dimensionality of Gabor responses, they reduced dimensionality by using PCA and local binary patterns (LBP) [Ojala *et al.* 1994]. They tested these dimensionality reduction techniques on the JAFFE database, and found that LBP outperformed PCA with an average recognition rate of 90%.

3.4.5 Two-Dimensional PCA

Yang *et al.* [2004] proposed an image representation technique which is said to overcome the high dimen-

sionality problem, named two-dimensional principal component analysis (2DPCA). 2DPCA is based on two-dimensional image matrices instead of one-dimensional vectors used in PCA. This means that a feature vector does not need to be obtained before feature extraction. Instead, using the original image matrices, an image covariance matrix is directly constructed. Image feature extraction uses eigenvectors. Yang *et al.* [2004] tested 2DPCA and evaluated its performance. They performed a series of experiments on three face image databases, the ORL database [Samaria and Harter 1994], the AR database, and the Yale face database [Georghiades 1997]. Across all trials, the recognition rate was higher when using 2DPCA than for PCA. Experimentation also showed that using 2DPCA to extract features is more computationally efficient than when using PCA.

3.4.6 Haar Filters

Haar filters are an alternative to Gabor Filters, but which do not face the disadvantage of high dimensionality faced by Gabor filters. The advantage of using a Haar feature over most other feature representations is its calculation speed. Haar filters use integral images, which allow for fast and efficient generation of the sum of the pixel values in a rectangular subset of an image. Thus, a Haar feature of any size can be calculated in constant time. [Viola and Jones 2001] also prominently used integral images to reduce the computation time.

The one-dimensional Haar decomposition of an array of size n is computed by recursively using averaging and differencing. If we have an array of four elements $[x_1, x_2, x_3, x_4]$, after the first iteration of averaging and differencing we have:

$$\left(\frac{x_1 + x_2}{2}, \frac{x_3 + x_4}{2}, x_1 - \frac{x_1 + x_2}{2}, x_3 - \frac{x_3 + x_4}{2} \right) \quad (3.4)$$

After the second iteration of averaging and differencing we have:

$$\left(\frac{x_1 + x_2 + x_3 + x_4}{4}, \frac{x_1 + x_2 - (x_3 + x_4)}{4}, \frac{x_1 - x_2}{2}, \frac{x_3 - x_4}{2} \right) \quad (3.5)$$

The two-dimensional Haar decomposition of an image can be obtained by generalizing the one-dimensional Haar decomposition using standard or non-standard decomposition. In standard decomposition, the transform is applied initially to each row and then each column of the input matrix. In non-standard decomposition, the transform is applied alternatively to rows and columns at every recursive level of the transform.

For a square image with n^2 pixels, the two-dimensional decomposition consists of n^2 wavelet coefficients, where every wavelet coefficient corresponds directly to a Haar wavelet. The initial wavelet is the mean of the pixel intensity of the image. The other wavelets are the difference in mean intensity values of squares that are diagonally, vertically, or horizontally adjacent. Therefore, for an image n^2 pixels, the Haar decomposition contains n^2 coefficients, implying that the two-dimensional decomposition is exactly complete. [Papageorgiou *et al.* 1998] shifted the wavelet basis at four times the density of the standard transform. The result was Haar coefficients which allowed for object recognition at an even finer resolution than when doing the standard Haar approach. Viola and Jones [2001] also modified the standard Haar approach by including an additional Haar wavelet which contained three sub-regions instead of one, two, or four.

3.4.7 AdaBoost

AdaBoost is a machine learning algorithm developed by Freund and Schapire [1995]. AdaBoost is often used in conjunction with several other machine learning algorithms to improve its performance. AdaBoost is considered adaptive because classifiers subsequently built are favourably adjusted by considering the previously misclassified instances. AdaBoost is less susceptible to the problem of over-fitting faced by other classifiers. However, AdaBoost is sensitive to noisy data and to outliers.

AdaBoost repeatedly calls a new, weak classifier in each of a series of rounds $t = 1, 2, \dots, T$ from a total of T classifiers. The classifiers it uses can be weak meaning they can display a substantial error rate. They will improve in the final model, provided that their performance is even slightly better than random. For each call of a weak classifier, a weight distribution D_t is updated, indicating the importance of samples for classification in the data set. In each round the weights are increased for every incorrectly classified sample, and the weights are decreased for each correctly classified sample. This results in the new classifier focusing on the samples which have so far been incorrectly classified. The AdaBoost algorithm is shown in Figure 3.2

Owusu *et al.* [2014b] developed a system which used Gabor filters and AdaBoost to provide fast and accurate FER. They used Gabor filters to extract features, but because Gabor representations are of an extremely high dimension, they used AdaBoost to then only select a few features which would be used for FER. They used a support vector machine classifier. The system was tested on the JAFFE database Kamachi *et al.* [1998] and Yale database [Georghiades 1997] and was shown to perform faster than other commonly used methods. The JAFFE database contains images showing the six basic emotions and the Yale database shows a neutral face, happiness, sadness, and surprise. The JAFFE database achieved an average recognition rate of 98%, while the Yale database achieved 92%. Owusu *et al.* [2014a] then also tested the system using a three-layer neural network with backpropagation. The JAFFE database achieved an average recognition rate of 97%, while the Yale database achieved 92%.

Algorithm AdaBoost(\mathbf{Z}, T)

Input: l examples $\mathbf{Z} = \langle (\mathbf{x}_1, y_1), \dots, (\mathbf{x}_l, y_l) \rangle$

Initialize: $w_1(\mathbf{z}_i) = 1/l$ for all $i = 1 \dots l$

Do for $t = 1, \dots, T$,

1. Train classifier with respect to the weighted sample set $\{\mathbf{Z}, \mathbf{w}^t\}$ and obtain hypothesis $h_t : \mathbf{x} \mapsto \{\pm 1\}$
2. Calculate the training error ϵ_t of h_t :
$$\epsilon_t = \sum_{i=1}^l w^t(\mathbf{z}_i)(h_t(\mathbf{x}_i) \neq y_i), \quad (3.6)$$

abort if $\epsilon_t = 0$ or $\epsilon_t \geq \frac{1}{2} - \Delta$, where Δ is a small constant
3. Set
$$b_t = \log \frac{1 - \epsilon_t}{\epsilon_t}. \quad (3.7)$$
4. Update weights \mathbf{w}^t :
$$w^{t+1}(\mathbf{z}_i) = w^t(\mathbf{z}_i) \exp \{-b_t(h_t(\mathbf{x}_i) - y_i)\} / Z_t, \quad (3.8)$$

where Z_t is a normalization constant, such that $\sum_{i=1}^l w_{t+1}(\mathbf{z}_i) = 1$.

Output: Final hypothesis

$$f(\mathbf{x}) = \sum_{t=1}^T c_t h_t(\mathbf{x}), \quad \text{where } c_t := \frac{b_t}{\sum_{t=1}^T |b_t|} \quad (3.9)$$

Figure 3.2: The AdaBoost algorithm.

3.5 Geometric Based Feature Extraction

The geometric method of feature detection looks at the geometric relationships between key facial feature points, also known as fiducial points. Facial features are extracted by using relative positions and sizes of the important components of face. Fiducial points are located along the eye, eyebrows, forehead and mouth. The geometric approach of facial feature extraction concentrates on the detection of edges, directions of important components, and regions in the images containing important components. Thereafter building feature vectors from these edges and directions. Filters such as the Canny filter or Gabor filters, which are edge detection operators, are used to detect the fiducial points such as the eyes or mouth regions of the facial image.

The relative distance approach uses location and displacement of facial features and converts them directly into a feature vector. Lien *et al.* [1998] developed a system to automatically recognize single upper face AUs or AU combinations by using hidden Markov Models. They used three approaches to extract facial information: high gradient component detection, dense flow tracking with PCA, also known as furrow detection, and facial feature point tracking. The feature vectors were calculated by the displacement of six fiducial points located at the upper boundaries of the left and right eyebrows. The displacement of each fiducial point in each frame was calculated by measuring the difference in its position in the neutral video frame from its position in the current frame. This forms the feature vector. The recognition rates for the upper face AUs using high gradient component detection was 85%, dense flow tracking was 93%, and feature point tracking was 85%.

Cohn *et al.* [1999] developed an automatic facial expression analysis system which made use of the geometric approach of feature extraction. One hundred subjects were recorded while performing a series of facial displays. They selected fifteen AUs and AU combinations that occurred 25 times or more to be used for automatic analysis. They located 37 fiducial points in the initial frame and tracked each of them in the image sequence using a hierarchical algorithm. They were then classified using discriminate analysis. The measurements were normalized and then image sequences were randomly selected to be split into a training set as well as a cross-validation set. They did discriminant function analysis on the feature point measurements. The results were compared to those acquired by manual FACS coding. They agreed 92% or more for AUs in the eyebrow, mouth, and eye areas of the training set. They agreed 88%, 91%, and 81% for AUs in the eye, eyebrow, and mouth regions, respectively in the cross-validation set.

Niese *et al.* [2012] developed an FER system based on geometric and optical flow features from colour images. Geometric feature vectors were acquired by extracting 3D features from each image in an image sequence. Then optical flow motion detection was carried out on subsequent images in the sequence, which resulted in temporary features. They used a neural network and support vector machine classifier and found that the colour data from the images resulted in a more advanced representation of facial features, making the task of FER more robust and accurate.

3.6 Comparison Of Feature Extraction Methods

Zhang *et al.* [1998] compared a geometric approach and appearance approach to feature extraction from face images. The geometric approach looked at the positions of a set of 34 facial fiducial points. The appearance approach extracted multi-scale and multi-orientation Gabor wavelets from the face image at the fiducial point locations. A total of 612 Gabor coefficients were extracted from the face image at 34 fiducial points. Their system composed of a two-layer perceptron. They compared the recognition results of the two approaches and found that the recognition rates for the six emotions (joy, anger, etc) were significantly higher when using the Gabor coefficients.

Donato *et al.* [1999] compared several methods of feature extraction to recognize six upper face AUs and six lower face AUs. These AUs were single, i.e. they were not in combination with any other AUs. The techniques compared were: facial motion analysis through estimation of optical flow, global spatial analysis (such as PCA and ICA), local feature analysis, LDA, and local filters (such as Gabor wavelets). All of these systems had a manual preprocessing step which aligned every input image to a standard face

image. A simple nearest neighbour classifier was used. The recognition rates were the highest for both Gabor wavelets and ICA, which both achieved 96% accuracy when classifying the six upper face and six lower face AUs. The results show the success in using local filters and high spatial frequencies for AU classification.

Geometric and appearance based methods can only be combined into a hybrid approach. Bartlett *et al.* [1999] compared three approaches: global analysis, explicit measurement of features such as wrinkles, and estimation of motion flow fields. These three approaches were then combined into a hybrid system to classify six upper face AUs. Manual face alignment took place. The hybrid system saw a recognition rate of 91%. The hybrid system also out-performed human non-experts and trained experts.

These are just a few of many studies undertaken to compare feature extraction techniques. In general, appearance based techniques see more successful recognition rates than geometric approaches. Since these early studies, research has focused on improving speed and accuracy of both the extraction and classification stages.

3.7 Feature Classification

Classification is the final step in an automatic FER system. Once features have been extracted, the next step is to recognize the facial expression (or AUs) using the extracted features. There are two main classification approaches that can be taken: the spatial approach and the spatial-temporal approach. The spatial approach looks at only a single image. This image could be part of an image sequence of video. Usually, systems using a spatial approach will make use of a standard face image which is known as a reference image. The spatial-temporal approach uses the temporal information in image sequences to classify features (or AUs). This means that it takes into account previous frames or images in the sequences to classify the current image.

There are many different classifiers that can be used in this step. The performance of a classifier often depends on the feature extraction procedure. We will be further detailing the two main approaches of classification and some methods within those approaches.

3.8 Spatial Approach

This approach use only a single, static image and does not use any temporal information for classification purposes. It uses only the information from the current input image, and can be used in conjunction with a reference image. The reference image is often a neutral face image. The static image can be taken from a video or image sequence, however it is treated independently from other images within the video or sequence. Several methods are used in the spatial approach of FER such as neural networks [Tian *et al.* 2001] [Tian *et al.* 2002] [Tian *et al.* 2000] [Zhang *et al.* 1998] [Wen and Huang 2003], support vector machines [Littlewort *et al.* 2002], linear discriminant analysis (LDA) [Cohn *et al.* 1999], Bayesian network [Cohen *et al.* 2003a] [Cohen *et al.* 2003b], and rule-based classifiers [Cohn *et al.* 2001]. We will be looking at neural networks and support vector machines.

3.8.1 Neural Networks

A neural network (NN) is a method to solve problems by simulating a neuron's activities. NNs are a computational model which has the ability to learn, adapt, generalize, and to cluster and organize data. This is one of the most popular data-driven techniques used in machine learning. A NN is inspired by and roughly replicates the behaviour of neurons in a human brain to process information.

A neuron within a NN is a single processing unit that accepts the input from other neurons or an external source and computes their weighted sum. There are different types of neurons based on their location within the NN: input neuron, hidden neuron, and output neuron. The NN trains itself with the data that is available in the training set and tests using the test set. The network learns by updating the weights associated with each neuron such that the weighted sum of the inputs in converging towards

some known output. The reliability and robustness of a NN is dependent on the data source, the range, quality and quantity of data. NNs have had many applications in the real world such as in handwriting and speech recognition and have proven successful in the task of FER.

Tian *et al.* [2000] used a NN to develop an automatic system to detect eye-state AUs based on FACS. They used Gabor wavelets to extract features in nearly frontal image sequences. Three eye-state AUs were detected: AU41, AU42, and AU43. The system tracked the corners of the eyes in each image of the sequence, and then eye data was extracted at three points of each eye using Gabor wavelets. The normalized Gabor wavelets were then used as inputs into an eye-state AU detector NN. The average recognition rate of this system was 83%.

Tian *et al.* [2001] developed a system to automatically analyze facial expressions, which included permanent features and temporary features. Permanent features include the eyes and mouth, and temporary features include furrows when activating certain muscles. The input images were nearly-frontal face images. The system detected the face in the first frame and was able to handle small head motion as well as faces partly occluded by glasses and hair. They used three-layer NNs consisting of one hidden layer and standard backpropagation. Separate NNs were used for the upper and lower face. The inputs to the network were the extracted features, and the output were the classified AUs. This system achieved a 96% recognition rate for the upper face AUs, and a 97% recognition rate for lower face AUs.

3.8.2 Support Vector Machines

Support vector machines (SVMs) have been one of the most widely used techniques in feature classification and in many other real world applications. The objective of this method is to maximize the distance between two classes in the input space which one wishes to classify. The data is assumed to be linearly separable. This principle was highly successful in classification and regression problems surpassing many other state-of-the-art methods after being introduced in 1992 [Boser *et al.* 1992]. SVMs are considered supervised learning models because they take a set of input data and predict, for each given input, which of two given possible classes forms the output. One of the main advantages of using SVMs is their ability to handle high dimensional feature vectors without it affecting their training time.

Littlewort *et al.* [2002] used SVMs in their system to recognize neutral expressions and six emotion-specified expressions. The system detected the face in each frame. This system was capable of only recognizing frontal faces and a neutral face reference image was needed. The system consisted of two stages. In the first stage, each SVM in this system was trained to distinguish between two emotions. In the next step, they converted the representation produced by the initial stage into a probability distribution over the neutral face and six emotion-specified expressions. They tested several approaches to accomplish this. These approaches included multinomial logistic ridge regression, nearest neighbour, and a simple voting scheme. The highest recognition rate of 92% was achieved by the SVM and multinomial logistic ridge regression model.

SVMs are often used in conjunction with Gabor wavelets for feature extraction. Although SVMs have the advantage of handling high dimensional feature vectors without it becoming computationally expensive, Gabor wavelets can suffer from the high dimensionality problem. For some time, Gabor coefficients and SVMs were considered the state-of-the-art method for FER. Whitehill and Omlin [2006a] evaluated the recognition rates of an AdaBoost and Haar wavelet approach for FACS AU recognition. They compared it to the commonly used approach of Gabor filters with SVMs. They tested on the Cohn-Kanade database and results showed that the Haar and Adaboost method achieved AU recognition rates similar to that of the Gabor and SVM method. An average recognition rate of 92% was achieved across all AUs tested. The Gabor and SVM method, as tested by Bartlett *et al.* [2002] and Donato *et al.* [1999], had an average recognition rate of 91% across all AUs tested. However, the Gabor and SVM approach is computationally expensive due to the high redundancy of the Gabor representation.

Zavaschi *et al.* [2013] presented a method for FER using an SVM as a classifier. They used a combination of two feature extraction techniques as inputs to the SVM in an ensemble approach. Gabor filters and LBP were used to extract features. A multi-objective genetic algorithm was employed to find the most successful ensemble which minimized the error rate and size. They tested on both the JAFFE

and Cohn Kanade databases. The ensemble approach saw recognition rates improve by between 5% and 10% over approaches which use only one feature extraction technique.

3.9 Spatial-Temporal Approach

This approach takes advantage of the temporal information of image sequences or video to classify features. This approach uses previous frames when classifying one or more frames. To take advantage of the temporal information, many techniques can be used such as hidden Markov Models (HMMs) [Bartlett *et al.* 2001] [Cohen *et al.* 2003b] [Cohn *et al.* 1999] [Lien *et al.* 2000], recurrent neural networks (RNNs) [Kobayashi and Hara 1993] [Rosenblum *et al.* 1996], and rule-based classifiers [Cohn *et al.* 2001] to classify facial features. We will be looking at hidden Markov Models and recurrent neural networks in further detail.

3.9.1 Hidden Markov Model

A Hidden Markov Model (HMM) is a finite set of states. Each state is usually associated with a multi-dimensional probability distribution [Rabiner 1989]. The transition from one state to another is governed by a set of probabilities known as the transition probability. For a given state, an outcome is associated with the probability distribution. This outcome from a state is visible to the user but not the state themselves, hence the name hidden Markov model is given to these systems. A set of assumptions are made when using HMMs. These are:

- The Markov assumption: It is assumed that the next state is only dependent on the current state.
- The stationary assumption: It is assumed that the state transition probabilities are independent of the actual time at which the transition takes place.
- The output independent assumption: It is assumed that the current observation is independent of the previous observation.

Bartlett *et al.* [2001] used Gabor wavelets and SVMs to recognize AUs. Then, to deal with the AU dynamics, they used HMMs. They applied the HMMs in two ways. Firstly they took the Gabor representations as input, and secondly they took the SVM outputs as input. They used PCA to reduce the Gabor coefficients to 100-dimensions in each image. The reduced Gabor representations were then used as inputs to train the HMMs. Two HMMs were trained and tested using leave-one-out cross validation, one for blinks and one for non-blinks. The system was able to handle large head motion. However, a neutral face was needed to classification and the system was only able to recognize single AUs (no AU combinations). The highest recognition rate achieved of 96% was when using three Gaussians and five states. A recognition accuracy of 98% was achieved for distinguishing between blinking and not-blinking when using SVM outputs as HMM inputs. Recognition rates for the three brow region categories, which included pulling the brow up, pulling the brow down, and not moving the brow, was 70% when using HMMs trained with PCA-reduced Gabors. Recognition accuracy was 67% for HMMs trained with SVM outputs.

3.9.2 Recurrent Neural Networks

There are two major types of neural networks. These are feedforward neural networks and recurrent neural networks. In a feedforward network, activation is sent from input units through to output units. In contrast, a recurrent neural network (RNN) has at least one connection that feeds back into the network. This allows activation to flow in a directed loop. RNNs use context units to store the previous time step data. The context units and the feedback connections allow the RNN to do temporal processing and learn sequences. RNNs are discussed further in Chapter 5.

Vadapalli [2014] used RNNs for the recognition of six upper face and five lower face AUs using Gabor filters for feature extraction. They applied different dimensionality reduction techniques including frequency scale selection, local Gabor filters and PCA, and studied their effect on the classification performance of RNNs. Results showed that local Gabor filters performed the best (85% average performance) for the six upper face AUs, whereas the use of all the Gabor filters performed the best (82% average performance) for the five lower face AUs. Network regularization through weight decay improved the classification performance (86% average performance for six upper face AUs and 85% for the five lower face AUs).

3.10 Feature Detectors and Descriptors

This section aims to look at feature descriptors and detectors and methods that have been developed to detect and describe features. Feature detection is a method that aims to look at points in an image and determine whether or not there is an image feature at a particular point. The features that are obtained from the image are a subset of the original image, often as points, connected regions, or curves. Feature detection is also known as interest point detection or keypoint detection.

There is no precise definition for a feature, and defining a feature is often dependent on the application. Features are considered “interesting” parts of an image. Features are often used as the first step for subsequent algorithms, and the success of the overall algorithm is usually dependent on the success of the feature vector. The most important property of a feature detector is its ability to be repeated. This is whether or not the same feature will be found in other images. In FER, an “interesting” part of the image could be furrows or wrinkles caused by the movement or contraction of facial muscles. These furrows could be related to the contraction of specific muscles that relate to an AU being present.

Feature detection is usually an initial image operation, and is therefore considered to be a low-level image processing operation. It examines each pixel in an image to determine if there is a keypoint present at that pixel. In instances where feature detection is computationally expensive, a higher-level algorithm can be used to assist with detecting features. This will ensure that only some areas of the image are searched for features, as opposed to the entire image region.

Once keypoints have been detected using the feature detector, local image areas around the keypoints can be extracted and described. The result is known as a feature descriptor. The feature descriptor is also known as the feature vector. Feature description methods include Scale-Invariant Feature Transform (SIFT) and Speeded-Up Robust Features (SURF). In addition to the feature vector, the feature detection process may also provide accompanying indicators, such as edge orientation and gradient magnitude.

3.10.1 Scale Invariant Feature Transform

The Scale Invariant Feature Transform (SIFT) is an algorithm first published by Lowe [1999] to detect and describe local features in images. SIFT can be used as an image descriptor for image matching and recognition. Originally, SIFT was geared towards the interest point detection and description of greyscale images where computations of local gradient directions of image intensities are gathered to describe the local neighbourhood around the points. Thereafter the descriptor can match interest points in different images. However, the SIFT descriptor has since been extended from use in greyscale images to the use in colour images [Bosch *et al.* 2006] [Van De Weijer and Schmid 2006] [Burghouts and Geusebroek 2009] [Van De Sande *et al.* 2010], and from 2D images to video [Laptev and Lindeberg 2006]. This means that SIFT can be used in the spatio-temporal realm. The SIFT descriptor has also been successfully used to categorize objects, classify texture, and to align images and bio-metrics [Bosch *et al.* 2006].

SIFT takes an image and converts it into a group of local feature vectors. Each of these vectors is scale, rotation, and translation invariant. Previous approaches which detect and describe local features lacked invariance of scale. SIFT features are found using a staged filtering approach. The initial stage finds keypoints by looking for locations that are maxima or minima of a difference-of-Gaussian function.

For each keypoint, a feature vector is generated, which describes the local area sampled relative to its scale-space coordinate frame.

For each keypoint, a nearest neighbour approach indexes potential object models. All keypoints that agree on a model pose are then identified through a Hough transform hash table, and then a least squares fit to obtain a final model parameter estimate. If three or more keypoints agree on a model parameter, there is strong indication that there is an object at that particular location. There may be several keypoints in the image of objects (such as face), therefore it is possible to have significant occlusion on the image, but high reliability levels are retained. Thus, SIFT can identify objects robustly even when there is partial occlusion. This is because SIFT feature descriptors are scale and orientation invariant, and are also somewhat invariant to distortion and illumination changes.

3.10.2 Speeded-Up Robust Features

Speeded Up Robust Features (SURF) is a robust local feature detector, originally published by Bay *et al.* [2006]. It is in part inspired by the SIFT descriptor [Lowe 1999]. Similar to SIFT, SURF finds “interesting” points within an image and describes the neighbourhood around those points to produce feature vectors.

SURF uses integral images which allows for fast and efficient implementation of box type convolution filters. By relying on integral images for image convolution, SURF achieves distinctiveness and robustness. SURF aims to find salient regions in near constant time because of its use of integral images and box filters. The SURF detector makes use of the Hessian matrix, which has seen good performance in computational time and accuracy.

SURF differs from SIFT as it is based on Haar wavelets rather than derivative approximations on an image pyramid, making it faster than SIFT. To detect keypoints, SURF uses the determinant of the Hessian operator, while SIFT uses the Laplacian operator. The SURF method of keypoint detection and description is further discussed in Chapter 4.

3.11 Out-Of-Plane Head Movement Techniques

In the real world, out-of-plane head motion is a common occurrence. When humans communicate, we have a natural tendency to move our heads. Therefore, in image data for the goal of FER, the human subject is not always facing a camera front-on, and changes and movements generated when the face is out-of-plane can contain valuable information for FER. Face detection methods should not only be able to detect frontal images, but also non-frontal faces. Different head pose detection techniques can be used to determine the face view. A non-frontal face image can be changed to a frontal image by 3D face modelling. In some cases, non-frontal faces can be reconstructed or normalized to a frontal view for expression analysis by generating an alternative view. We look at some methods of face detection and head pose detection that have been used when the face is out-of-plane. We also look at an approach which models a 3D face that is out-of-plane using 2D images. Additionally, an alternate angle face view method is outlined.

3.11.1 Out-of-plane Face Detection

Many face detection algorithms depend on the face having a frontal view. To combat the problem of non-frontal view, many detection methods have been developed to detect faces in out-of-plane instances. Heiselet *et al.* [2001] developed a trainable, component based system to detect faces that were near frontal in static, grey images. The system consisted of a two-level hierarchy SVM classifier. On the first level, classifiers detected components of a face. On the second level, a single classifier checked if the detected components matched a geometrical face model. The method used 3D head models to automatically learn components. This system was fully automatic and no manual intervention was required.

Pentland *et al.* [1994] developed a method that could handle varying head positions. It used a view-based and modular eigenspace method. This system could run real-time and yielded higher recognition

rates than other techniques used for out-of-plane head motion. It was also considered a more robust system for face recognition than other systems at the time.

Schneiderman and Kanade [2000] developed a method that could reliably detect faces that were out-of-plane. It looked at 3D objects and the statistics of each object appearance using the product of histograms. Each histogram represented a subset of wavelet coefficients and the object position.

3.11.2 Head Pose Detection

If a head moves out-of-plane, head pose can be estimated to determine the head geometry and the orientation of the head. This allows for systems to determine the degree at which the head is rotated to assist with possible reorientation. In general, the two types of head pose estimation methods used are 2D image-based methods or 3D model-based methods.

2D Image-Based Method

A 2D image-based method looks at 2D images to determine the head pose or orientation of the face. To handle head motion which often occurs in the process of FER, some systems detect the head instead of the face, as opposed to common face detection techniques. The head is first distinguished from the background, and then the head silhouette is segmented by background subtraction. Intuitively, segmentation into parts usually happens at the silhouette's negative curvature minima points. Tian *et al.* [2003] used a three-layer NN for head pose estimation. The inputs to the NN were the processed head image after being converted to greyscale, histogram equalized, and resized to the desired resolution. The outputs of the NN were the three head poses: frontal or almost frontal view, profile, and others. These included behind the head or hidden face. The Tian *et al.* [2003] system performed well even when receiving very low resolution face images as input.

In frontal or almost frontal views of a face, both the eyes and the corners of the lips can be seen. In a profile view of a face, only one corner of the mouth and only one eye can be seen. Additionally, a nose profile is visible. Many systems can only analyze facial expressions on faces that are frontal, or almost frontal, and largely out-of-plane faces are often discarded. Alternatively, profile views can be reconstructed to obtain an alternative view of the face, such as a frontal face view. This generation of an alternative view of the face is discussed below.

3D Model-Based Method

The 3D model-based method looks at 3D head models to determine the head pose. Zhao *et al.* [2002] presented a real-time system for determining the head orientation of a human based on visual information. Two NNs were trained to approximate the functions that map an image of the head to the orientation of the head. An electromagnetic tracking device was worn by subjects to obtain training data. Experimental results showed orientation accuracy within ten degrees, with the subject free to move at distances of approximately three to ten feet from the camera.

The Tao and Huang [1999] system used a three-dimension wire frame facial model which tracked facial features that were defined by the model. Initially, they fit the model to the first image of an image sequence. They selected landmarks on the face and then manually aligned the image and the model. These landmarks included the eyes and the mouth. The standard facial model was fitted to the selected features on the face. This consisted of sixteen locations. A two-step process was used to estimate the head motion. Firstly, the 2D image motion was tracked by matching templates on different images at several resolutions. Thereafter, from the 2D changes of the points on the facial model, they could estimate the 3D head movement by solving a system of equations of the projective motion in the least-squares sense as used by [Essa and Pentland 1997]. Following this, Bartlett *et al.* [2001] used a canonical wire mesh facial model which assisted in estimating the facial geometry and 3D head pose from manually labelled feature points.

In the system developed by Xiao *et al.* [2003], head motion could be identified in image sequences in real-time using a cylindrical head model. The cylindrical head model was automatically mapped to the face, which was acquired by face detection, and then used as the first facial template. Then, for each image of the sequence, the template for that particular image was the face image from the previous image that was placed onto the model. The template was then used to determine the head motion. They used certain techniques to deal with nonrigid motion and occlusion, and they dynamically updated the template to deal with lighting changes. This allowed the system to work even if the face was mostly occluded.

3.11.3 3D Face Modelling Using Multiple 2D Images

Another method that has been used to handle out-of-plane head movement is to model a 3D view of the face using multiple 2D images. This allows for a frontal view of the face to be acquired. Parke [1972] pioneered the field of modelling a human face in 3D and since then there have been several algorithms developed to model faces geometrically [Parke 1974]. The 2D-based methods discussed above do not consider the different structures of human faces, and thus when tested on profile images, result in poor performance. Lam and Yan [1998] used face images with out-of-plane motion and modelled these sample images into frontal faces based on a cylindrical facial model. This used an analytic-to-holistic method which identified faces at several variations in perspective. However the drawback of this system was that it required heavy manual labelling work.

Wang *et al.* [2005] developed an efficient, automatic method for reconstructing realistic 3D faces by using several 2D facial images which could be of any degree of rotation. An algorithm to align multi-view faces, developed by Hu *et al.* [2003], was used to find the local feature points on the faces from the images. Their Syncretized Shape Model reconstructs the 3D facial geometry. Then the pose and shape of the face were determined by an algorithm. The correspondence between the contour points and their vertices in the 3D facial models were found. Texture was determined using their Syncretized Texture Model, which makes use of the Texture Confidence Function. Results showed realistic reconstructed 3D face models. This method has many advantages. It is efficient, there is no manual interaction required, and it is robust to pose variation. The reconstructed 3D facial model obtained from this method was said to be more realistic than many other reconstruction models of the time because it used co-enhancement of the several images

3.11.4 Alternate Angle Face View Generation

Given a view of a face at a particular angle, an alternative view may be required, which can be retrieved by developing a 3D view of the face. At least three images are needed of an object to obtain the precise 3D geometry of an image. However, novel views can be generated from only a single image. Blanz and Vetter [1999] pioneered the method of a 3D algorithm to gain alternative views of the face from just a single image. The disadvantage of this approach was that it was computationally expensive as shown by Blanz and Vetter [2003] and Blanz *et al.* [2005]. Because of the large cost, applications where real-time processing was required, such as for airport security, were limited.

Ni and Schneiderman [2005] proposed an algorithm for the accurate generation of a face showing a different orientation using a single input image. Two poses were represented by stacking pixel and location information into a combining feature space. The test vector consisted of the input image and the missing generated image. The missing part was then determined by maximizing the probability of the test vector. The approach uses the “distance-from-feature-space” and “distance-in-feature-space” to maximize the test vector’s probability by minimizing the weighted distance sum. This difference between this approach and other alternate view algorithms is that this approach does not need a 3D model or 3D data to train, and does not correspond images. The algorithm has a low computational cost, and a face can be generated in only four to five seconds. This algorithm has also shown to be more accurate than other common approaches.

The method of generating new, alternate angles of a face using only a single input face image is beneficial to FER to overcome the variations of pose. Large out-of-plane head motion often makes FER an incredibly challenging task.

3.12 Conclusion

This chapter provides an overview of the facial expression recognition procedure and the steps involved. We look at an overview for each step, however there is a detailed literature review for feature extraction and feature classification techniques. The two main approaches for extraction and classification are detailed, as well as a comparison of the approaches. We also look at methods that have been used to deal with out-of-plane head movement. Some of these systems use a technique which detects the head pose to determine if the image is frontal enough to be used for analysis. Often, side or profile views of the face are discarded for the use in FER. Alternatively, to combat the problem of profile views, methods have been developed which can obtain a frontal view of the face by reconstruction or alternative angle view generation. We will be looking to recognize AUs where the face moves out-of-plane for some part of an image sequence. Instead of using one of the methods explained above, we look at the ability of RNNs to handle out-of-plane head movement because of their temporal processing abilities. RNNs are briefly addressed in this chapter, however they are further detailed in Chapter 5.

Chapter 4

Speeded Up Robust Features

4.1 Introduction

Speeded Up Robust Features (SURF) is a robust local feature detector, originally published by Bay *et al.* [2006]. It has had applications in both object recognition and 3D reconstruction of images [Bay *et al.* 2008]. It is partly inspired by the Scale Invariant Feature Transform (SIFT) descriptor. SIFT detects and describes features in images [Lowe 1999]. SURF provides feature vectors of size 64, which is much smaller than other commonly used feature extraction techniques. SURF is comparable to or outperforms other methods with respect to distinctiveness, robustness, and repeatability. It is also much faster than other feature detectors and descriptors. The SURF descriptor is scale and rotation invariant and is unaffected by small viewpoint changes or illumination changes

The first step in the SURF method is to find the keypoints (or interest points) of an image and the second step describes the keypoints. The keypoint detector locates the keypoints in the image, such as a wrinkle created during activation of an AU, and the keypoint descriptor then describes the local neighbourhood of the keypoints. The descriptors result in the feature vectors of an image. There is a feature vector of length 64 for each keypoint of an image. SURF operates on the integral image using a Hessian matrix approximation to detect the keypoints, which significantly lowers the computation time. An integral image is a data structure and algorithm for the efficient generation of the sum of pixel values in a rectangular subset of an image. It is also often used to calculate the average intensity of an image. It was first prominently used within the object detection framework of Viola and Jones [2001]. To compute the descriptor, the first-order responses of Haar wavelets are found in the x and y directions, which describes the intensity distribution of the neighbourhood of a keypoint. SURF features have an indexing scheme which uses the sign of the Laplacian. SIFT feature vectors consist of 128 dimensions and are thus larger than those of SURF. Each feature vector can then be normalized to the unit length, creating a probability density function (PDF) descriptor.

4.2 Keypoint Detectors

The Harris corner detector is a widely used keypoint detector [Harris and Stephens 1988], which is based on the eigenvalues of the second-moment matrix for its corner decisions. Harris corners are invariant to image rotation, but are not invariant to image scale. To overcome this, Lindeberg [1998] introduced automatic scale selection where keypoints are detected in an image, and where each keypoint has its own characteristic scale. Lindeberg [1998] also experimented with the determinant of the Hessian matrix and the Laplacian to detect blob-like structures. Blob-like structures are considered regions of interest. Mikolajczyk and Schmid [2004] then improved this method by developing a scale-invariant feature detector which is robust and has the benefit of high repeatability. They called this the Harris-Laplace and Hessian-Laplace. For the Harris-Laplace, they used a Harris measure to find the location, and the Laplacian to choose the scale. Similarly, for Hessian-Laplace, they used the determinant of the Hessian matrix to find the location. Lowe [1999] approximated the Laplacian of Gaussian by a Difference of Gaussians

filter in order to improve speed. SURF follows on this by using the determinant of Hessian matrix approximation as the base of the detector. Integral images are used during Hessian matrix approximation for the fast and efficient evaluation of box filters. The integral image representation J of an image I is defined as:

$$J(x, y) = \sum_{i=0}^x \sum_{j=0}^y I(i, j) \quad (4.1)$$

Given an image I , and a point $X = (x, y)$, the Hessian matrix $H(X, \sigma)$ in X with scale σ is defined as:

$$H(X, \sigma) = \begin{bmatrix} L_{xx}(X, \sigma) & L_{xy}(X, \sigma) \\ L_{xy}(X, \sigma) & L_{yy}(X, \sigma) \end{bmatrix} \quad (4.2)$$

where $L_{xx}(X, \sigma)$ is the second order Gaussian derivative convolution, $\frac{\partial^2 g(\sigma)}{\partial x^2}$, where the image is at point $X = (x, y)$ (similarly for L_{yy} and L_{xy}). These are considered the Laplacian of Gaussians. A set of 9x9 box filters, which lowers computation time, is used as the approximations of a Gaussian with $\sigma = 1.2$. The approximations are denoted by D_{xx} , D_{yy} and D_{xy} . The weights applied to the rectangular regions are simple for computational efficiency. This results in:

$$\det(H_{approx}) = D_{xx}D_{yy} - (wD_{xy})^2 \quad (4.3)$$

where w is the energy conservation weight between the approximated Gaussian kernels. $w \approx 0.9$. Scale space must be considered to detect keypoints using the determinant of Hessian. SURF uses a pyramid scale space.

4.3 Keypoint Descriptors

SURF uses a basic Hessian Matrix approximation. Integral images are used which reduce the computation time and to increase efficiency, and is thus known as the Fast-Hessian detector. The descriptor describes a Haar-wavelet response distribution within the keypoint neighbourhood. Additionally, only 64 dimensions are used, as opposed to SIFT's 128 dimensions, reducing feature computation and matching time, and simultaneously increasing robustness.

A feature of a keypoint is described using the Haar wavelet response sums. The use of Haar wavelets increases robustness and decreases computation time. The first step of keypoint description constructs an oriented square region around and centered at the keypoint. The region is then equally divided into 4x4 square sub-regions. For each sub-region, at 5x5 regularly spaced points, the Haar wavelet responses are determined. The Haar wavelet response is called d_x horizontally, and d_y vertically. Then the sum of d_x and d_y is found in each sub-region and this results in the first few elements in the feature vector. The absolute values of the responses is found and summed. Each sub-region now has a four-dimension vector v describing the neighbourhood and showing its underlying intensity structure

$$v = (\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|) \quad (4.4)$$

The descriptor vector v consists of four values from each sub-region of the image, resulting in a vector of length $4^3 = 64$.

A keypoint descriptor represents the gradient magnitude and orientation at each keypoint in a region around a keypoint location. Each keypoint descriptor (i.e. feature vector) can then be normalized to the unit length and a PDF descriptor is generated.

In some cases, keypoint-based feature extraction can result in a large number of false positive detections. If this is the case, it can be overcome by using hypothesis rejection methods, such as the Random Sample Consensus (RANSAC) [Fischler and Bolles 1981].

4.4 Systems Using SURF

The following are face or facial expression recognition systems that make use of SURF detectors and descriptors. These systems are tested on different databases, and make use of different classifiers than our proposed method.

4.4.1 JAFFE Database

Huang and Tai [2012] proposed a new method for FER. Facial feature vectors are generated from key-point descriptors using SURF. Each facial feature vector is then normalized and the PDF descriptor is obtained. The distance between two PDF descriptors is calculated using Kullback Leibler divergence:

$$k(y_1, y_2) = \frac{1}{n} \sum_{i=1}^n y_1(x_i) \log\left(\frac{y_1(x_i)}{y_2(x_i)}\right) \quad (4.5)$$

where n is the number of bins and x_i is the i th bin. The minimum distance between the two PDF descriptors is found using:

$$D_j = \arg \min_{y_j} k(y_i, y_j) \quad (4.6)$$

where y_i is a descriptor in the i th image, y_j is one of the m descriptors in the second image, and D_j is a descriptor in the j th image.

The recognition among the classes in a space can be measured using class separability. The PDF descriptors are utilized as a class separability measure. The recognition tally for each descriptor y_i from class A is defined as:

$$T(y_i) = \frac{\sum_{B=1}^{C-1} \left(N_B \sum_{j=1}^{N_B} k(y_i, D_j) \right) / C - 1}{N_A \sum_{j=1}^{N_A} k(y_i, D_j)} \quad (4.7)$$

where C is the amount of classes and class B has N_B samples. The denominator represents the minimum PDF descriptor of class A , such as happy. The numerator represents the minimum PDF descriptor of the other classes, such as angry.

Each training image is divided into a uniform grid. Depending on the number obtained from the recognition tally, that number of largest PDF descriptors of each grid are extracted from the training image of each class. These PDF descriptors are treated as recognition patterns of the test image.

Following feature extraction using the SURF method, the 4x4 uniform grids for each image and the largest PDF descriptors depending on the tally are chosen for each expression class. These PDF descriptors are used to discriminate facial expression F for each grid. The central part of a face contains most of the important information. A Gaussian mask G on the grids is selected and the central part of the grids is assigned a heavier weight. Classification depends on the cumulative matching scores of each grid. The total scores of each AU for all grids are calculated to determine the highest score of the AU in each image.

This method by Huang and Tai [2012] was tested on the JAFFE database and achieved a recognition rate of 95%. The JAFFE database contains images of ten people showing six facial expressions which were recognized by this system. The expressions are happiness, surprise, fear, sadness, anger, and disgust. The implementation was replaced by SIFT and the difference in performance measured. The average recognition rate using SIFT was 72%, which is significantly lower than the results when using SURF.

4.4.2 FERET Database

Du *et al.* [2009] exploited the performance benefits of SURF to develop a method of face recognition. After detecting 30 to 100 SURF keypoints on the face, a feature vector is created. To do feature matching, they introduced geometric limits into point matching with SURF features. Similarly to FER, face recognition usually uses frontal and normalized faces so that corresponding points in different images have locations which are near to each other. Therefore, for a keypoint (x, y) , the search area for the corresponding keypoint in another image is within a rectangular box with centre (x, y) . A candidate matching pair is found when the pair with the smallest distance between descriptors is found. Verification is performed by finding the next minimum distance pair that contains the same point as the one in the search image. The other images is then searched wholly. There is a pre-defined threshold which much be greater than the ratio of the two distances. If so, the point pair is considered to be matched. This method avoids mismatching because of the introduction of location information and the minimum distance approach.

Du *et al.* [2009] define a similarity measure (*sim*) for face recognition. This is the amount of matching points, the mean value of the Euclidean distance, and the mean distance ratio of all of the matching points.

$$sim = \begin{cases} (DisAvg + RatioAvg)/2 & \text{if } N \geq 10 \\ (DisAvg + RatioAvg)/2 + 1 & \text{if } N < 10 \end{cases} \quad (4.8)$$

$$DisAvg = \frac{1}{N} \sum_n MinDis \quad (4.9)$$

$$RatioAvg = \frac{1}{N} \sum_n DisRatio \quad (4.10)$$

N is the amount of matching points from the two face images, $MinDis$ is the Euclidean distance between the two matching points, and $DisRatio$ is the distance ratio of the matching points, where $n = 1, 2, \dots, N$.

Experiments were performed on the FERET [Phillips *et al.* 1996] database to evaluate the success of the algorithm. The datasets are standard testing subjects from the FERET database. The results were compared with that of using SIFT. Results showed that the SURF features perform better than SIFT in terms of recognition rate, and there is also a large improvement in matching speed.

4.4.3 SURF and SVMs

Kim and Dahyot [2008] presented a feature based method to determine if principle points belong to objects on the face or if they belong to the background. They used SURF descriptors as feature vectors and SVMs as classifiers. The system consisted of a two-layer hierarchy of SVM classifiers. A single classifier on the first layer checks if a feature belongs to face images or not. Component labeling is operated on the second layer using each component's classifier such as eyes, mouth, and nose.

Test data sets consisted of three subsets of high resolution, lower resolution, and lowest resolution face images. The high resolution subset of faces comprised of 100 randomly selected face images taken from the AR database [Martinez 1998] and Caltech face database [Angelova *et al.* 2005]. These images were then resized to obtain the lower and lowest resolution data sets.

In the high resolution and lower resolution test data, the classification results were very accurate for the eyes. For high resolution, left and right eye recognition was 97%. For lower resolution, left eye and right eye recognition was 88% and 93% respectively. Mouth recognition was also promising with 93% for high resolution images. Nose recognition was 72% for high resolution. However, for the lowest resolution, detection results were below 50% for all components. When comparing to the recognition rate Kim and Dahyot [2008] received using OpenCV face detector, the recognition rate for the lowest resolution was only 25%.

4.4.4 SURF and PCA

Lin *et al.* [2012] wanted to develop a robust face recognition system which overcomes problems such as light changes, expression changes, head movements and accessory occlusion. SURF was used to extract the feature vectors with scale invariance and pose invariance from face images. Then PCA was introduced for projecting the SURF feature vectors to the new feature space as PCA-SURF local descriptors. Finally, k-means was applied to clustering feature points, and the local similarity and global similarity were then combined to classify the face images.

This was tested on two databases. The first is the CAS-PEAL-R1 database, which is a Chinese face image database where face images are constrained in many different conditions [Gao *et al.* 2008]. In testing, two test image sets with expression and accessory variation were chosen to perform face recognition. The second database is the ORL which contains 40 people and each one has ten images with different orientations and facial expressions [Samaria and Harter 1994].

Results on the CAS-PEAL-R1 database showed that the PCA-SURF method received a higher recognition rate than other tested methods in both the accessory and expression datasets. The other tested methods are local binary patterns [Ojala *et al.* 1994], SIFT and PCA-SIFT. The PCA-SURF method by Lin *et al.* [2012] received an average of 95% recognition rate on both the expression and accessory datasets.

Results on the ORL database showed that PCA-SURF also received a higher recognition rate than other tested methods. These methods were based on SIFT. The PCA-SURF method received a recognition rate of 97%.

4.5 Conclusion

This chapter introduces the SURF method of feature detection and extraction. Keypoint detectors are introduced as well as keypoint descriptors. In the original system developed by Bay *et al.* [2006], there is a method of image matching included to match objects of differing scale in different images. We are not matching images, but rather using SURF descriptors as inputs to a recurrent neural network. The SURF descriptors describe the local area of keypoints. We are using SURF to obtain feature vectors because the SURF descriptors are far smaller than many other common feature extraction techniques. For each keypoint, a SURF feature vector is only of length 64. The number of keypoints in an image will determine the size of the total feature vector, which we will use to train our RNNs.

Chapter 5

Recurrent Neural Networks

5.1 Introduction

There are two main types of neural networks. These are feedforward neural networks and recurrent neural networks. In a feedforward network, activation is sent from input units through to output units. In contrast, a recurrent neural network (RNN) has at least one connection that feeds back into the network. This allows activation to flow in a directed loop. RNNs use context units to store the previous time step data. The context units and the feedback connections allow the RNN to do temporal processing and learn sequences. RNNs can take many different forms, but all forms share a common feature: there is a multi-layer perceptron as a subsystem. The Hopfield [1982] network is not necessarily considered to be an RNN, but is designed to process sequential patterns. Hopfield was the first to add feedback connections to neural networks, where the outputs are fed backwards into the inputs. It demonstrated that feedback connections can hold memory.

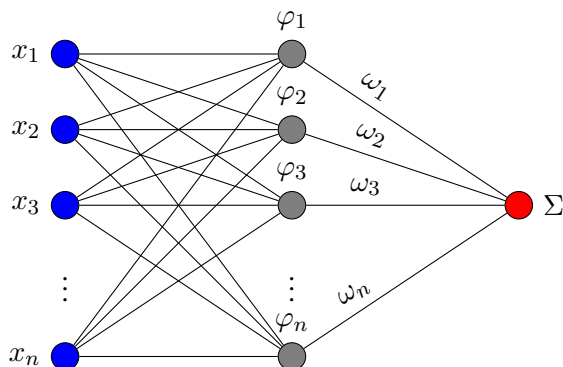
There are two types of learning that are used for machine learning purposes: supervised and unsupervised. In supervised training of RNNs, the data used to train the RNN is either observed or constructed time series data consisting of an input and output pair. This represents samples of the behaviour that the model is expected to exhibit. The training data is used to “teach” an RNN so that it reproduces the training data, so that ultimately the RNN can then generalize to novel inputs. In other words, when the trained RNN receives an input similar to the input from the training data set, it should give an output similar to the output that was originally given. This chapter details RNNs, which will be used as a classifier in our AU recognition system.

5.2 Feedforward Neural Networks

A neural network (NN) can be thought of as a directed graph, where the neurons are nodes, and the synaptic weights are the connections of the graph. In a feedforward NN only neurons of adjacent layers are interconnected with synaptic weights [Hornik *et al.* 1989]. Each layer of the NN has connections to the next layer and there are no backward connections, thus there is no memory or time data incorporated. The feedforward NN begins with an input layer which may be connected to a hidden layer or directly to an output layer. The input layer, which is the first layer, receives the input data to the network. The final layer is the output layer which produces the output or prediction of the network.

The processing of a feedforward NN begins when an external input pattern is copied to the input layer. The neurons of the input layer communicate the pattern to the subsequent layers through synapses. The pattern is then received by neurons of non-input layers and modulated by the weight of their connections. Weights are denoted as W_{jk} , where j is the neighbour neuron, and k is the neuron. Each neuron receives stimulation from other neurons, except the input layer which captures the pattern. Once the inputs are modulated, they are integrated and an activation value is determined. Often the activation value is just the integration of the modulated inputs, but may also be a function $F_i(a_i(t-1), net_i(t))$, where $a_i(t-1)$ is the activation at the previous time step and $net_i(t)$ is the integration of modulated inputs.

The feedforward algorithm is shown in Algorithm 1. The activation is fed forward through the NN. The most important thing to note from this algorithm that there are no backward connections. An RNN consists of at least one feedback connection. A feedforward NN is shown below. The NN takes input x_1, x_2, \dots, x_n and outputs σ , with hidden layer neurons $\varphi_1, \varphi_2, \dots, \varphi_n$.



Algorithm 1: The Feedforward Algorithm

Input: Set of inputs from the environment

Output: Set of output values calculated by the feedforward neural network

- 1 **foreach** layer from the first non-input layer to the output, **do**
 - 2 **foreach** unit on the current layer, **do**
 - 3 Set the accumulated input value for this unit to zero;
 - 4 **foreach** input connection to this unit, **do**
 - 5 Compute the modulated input across this connection;
 - 6 Add the modulated input to the accumulated input;
 - 7 Convert the accumulated input to its corresponding output;
 - 8 Store the output value for the unit in the layer structure;
 - 9 Return the output values from the top-most layer structure;
-

5.3 First-order Recurrent Neural Networks

First-order RNNs have been applied to many real world applications such as speech recognition, handwriting recognition, and financial forecasting. A first-order RNN uses context units to store the previous time step data. The hidden unit activations for a first-order RNN at time $t + 1$ is given as:

$$S_j^{t+1} = g \left(\sum_{i=1}^K V_{ij} I_i(t) + \sum_{i=1}^N W_{ij} S_i(t) \right) \quad (5.1)$$

where K is the number of input units and N is the number of hidden units. V_{ij} and W_{ij} are the weights associated with the input and hidden neurons respectively. $g()$ is the transfer function. I_j and S_j are the output values of the input and hidden neurons at time t .

There are several commonly used variations of first-order RNNs. These include the Elman [Elman 1990] RNN and the Jordan RNN [Jordan 1986], which differ based on the values that the context units accept. In the Elman RNN, the context layer accepts a copy of the output of the hidden neurons in the hidden layer. Thereafter, the values of the context neurons are used as an additional input to the hidden layer in the next time step. In the Jordan RNN, the values of the output neurons are stored in the context layer.

5.3.1 Elman Recurrent Neural Network

The Elman RNN is structured similarly to a regular feedforward network [Elman 1990]. Therefore all the neurons in one layer have a forward connection to all the neurons in the next layer. Figure 5.1 shows the structure of an Elman RNN. From this image, we see that there is an exception in the form of the context layer. In this layer, the context neurons each carry a copy of the output of the neurons of the hidden layer, known as the hidden neurons. The context neuron's value is then used as an additional input to the hidden layer in the next time step. Therefore, an Elman RNN has memory of one time unit. In Elman architecture, the context units take the output values of the hidden units with a time lag, and are then used along with the inputs as inputs to the hidden units in the next time step. Thus, in Elman networks, the number of context units is equal to the number of hidden units.

At each time step, t , the input is propagated in a standard feedforward way and thereafter an updating rule is applied which adjusts parameters. This updating rule will take into account the information received from the previous time step, $t-1$, hence making the network recurrent. This allows the networks to make sequence predictions, which cannot be achieved by standard feedforward networks.

In an Elman network, the strength of each connection between neurons is determined by a weight. Initially, all weights are chosen randomly and then optimized throughout the training process. In an Elman RNN, the weights of the connections from the hidden layer to the context layer are usually set to one and are fixed. They are fixed because the output of the hidden layer must be copied exactly to the context layer.

For input $x(t)$ at time t , we have hidden layer $h(t)$ and output layer $y(t)$. The Elman RNN can be represented as:

$$h(t) = f(Ux(t) + Vh(t-1)) \quad (5.2)$$

and the sigmoid function used at the hidden layer is:

$$f(x) = \frac{1}{1 + e^{-x}} \quad (5.3)$$

where U and V are weight matrices used to adjust the weights between the input and hidden neurons, and between the context and hidden neurons.

At the output layer, we also have a weight matrix:

$$y(t) = g(Wh(t)) \quad (5.4)$$

where

$$g(z_m) = \frac{e^{z_m}}{\sum_k e^{z_k}}$$

Elman RNNs have predominantly been used in the recognition of facial expressions and AUs. The time lag of one is considered sufficient when classifying AUs. The works of Vadapalli [2014], Tai and Chung [2007], and Wan *et al.* [2012] use Elman RNNs and have achieved promising results using this type of RNN.

5.3.2 Jordan Recurrent Neural Network

The Jordan RNN is very similar to the Elman RNN. The only difference comes in where a copy of the output neurons is stored in the context layer, rather than a copy of the hidden neurons as in an Elman RNN. Thus, the hidden layer can be represented as:

$$h(t) = f(Ux(t) + Vy(t-1)) \quad (5.6)$$

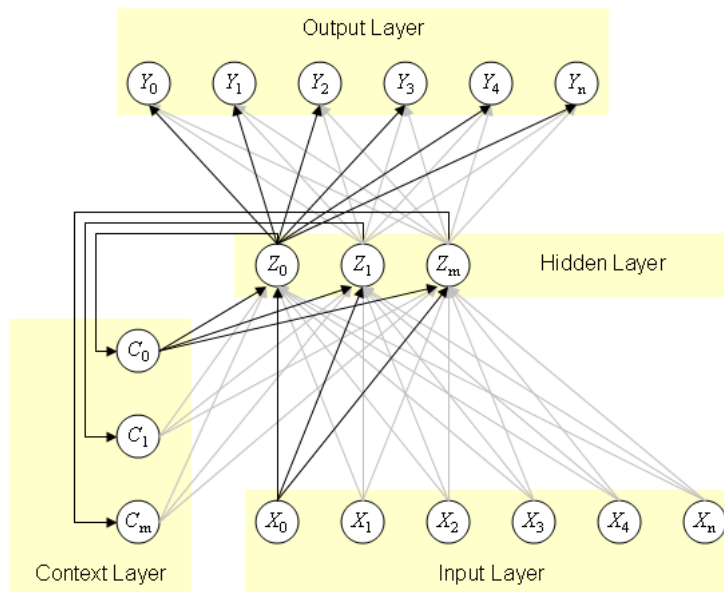


Figure 5.1: Elman RNN

5.4 Backpropagation

Backpropagation is a method of supervised learning [Rumelhart *et al.* 1988]. It is often used to train feedforward NNs. To use the backpropagation algorithm, it must be provided with both example inputs and target outputs. The outputs generated by the NN are then compared against the target outputs of the given example. Using the target outputs, the backpropagation training algorithm then calculates error and adjusts the weights of the various layers backwards from the output layer to the input layer. Backpropagation works by, for each modifiable weight, calculating the gradient of a cost function and then adjusting it accordingly [Park *et al.* 1993]. The most frequently used cost function is the summed squared error [Krogh *et al.* 1995].

Backpropagation is often used to train feedforward NNs, however it can be used to train other types of networks, and likewise feedforward networks may be trained with other methods. The algorithm for backpropagation is shown in Algorithm 2. Any network structure can be trained with backpropagation when desired output patterns exist and each function that has been used to calculate the actual output patterns is differentiable.

The Elman RNN network can be trained with gradient descent backpropagation and optimization. Backpropagation can have some shortcomings in many applications. This is because it is not guaranteed to find the error function's global minimum. The gradient descent can stop at a local minima, where it could stay indefinitely, prohibiting the global minimum from being found. Additionally, training sessions often consist of a large amount of iterations in order to find a desired weight solution, due to the difficulties inherent in gradient descent optimization.

Errors can be backpropagated even further. This is known as backpropagation through time (BPTT). All recurrent weights are duplicated spatially for a random number of time steps.

5.5 Backpropagation Through Time

Backpropagation through time (BPTT) is a gradient-based technique used to train some RNNs. It is an extension of the standard backpropagation algorithm where the output values of the hidden units at time $t - 1$ are regarded as additional input units at time t , and the error generated at the hidden units is used to modify the weights to these additional input units. The error can be propagated even further than one unit of time. The basic principle of BPTT is to unfold the network. In supervised learning methods, the role

Algorithm 2: The Backpropagation Algorithm

Input: Set of examples E

- 1 **foreach** *example e in a set of examples E* **do**
- 2 Calculate $O(e)$ for $I(e)$ with feedforward (refer to Algorithm 2);
- 3 Call function **CalculateOutputDeltas**($O(e), T(e)$);
- 4 Call function **CalculateInternalDeltas**;
- 5 Call function **UpdateWeights**;
- 6 **CalculateOutputDeltas**($O(e), T(e)$):
- 7 Get output values $O(e)$ from the output layer neurons;
- 8 **foreach** *individual output value $O(e)_i$* **do**
- 9 Calculate error ϵ as $O(e)_i - T(e)_i$;
- 10 Calculate $\delta_{O(e)_i} = \partial f(O(e)_i) \times \epsilon$;
- 11 Add $\delta_{O(e)_i}$ to set of deltas Λ
- 12 **CalculateInternalDeltas**:
- 13 Let Λ_{i+1} be the next layer's set of deltas;
- 14 **foreach** *non-output layer i* **do**
- 15 **foreach** *neuron j in this layer* **do**
- 16 Initialize error ϵ as 0.0;
- 17 **foreach** *neuron k of the next layer* **do**
- 18 Calculate ϵ as $\epsilon + \Lambda_{i+1,k} W_{ijk}$;
- 19 $\Lambda_{i,j} = \partial f(\epsilon \times \text{neuron } j\text{'s output})$;
- 20 **UpdateWeights**:
- 21 /* η is the learning rate */
- 22 **foreach** *layer i* **do**
- 23 **foreach** *neuron j in this layer* **do**
- 24 **foreach** *neuron k of the next layer* **do**
- 25 Calculate ΔW_{ijk} as $\Lambda_{i,j} \times \text{neuron } j\text{'s output}$;
- 25 $W_{ijk} \leftarrow \eta \times \Delta W_{ijk}$;

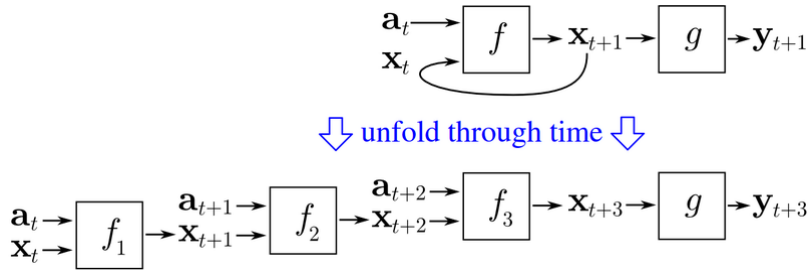


Figure 5.2: BPTT with the network unfolded to a depth of $k = 3$

of the training algorithm is to adjust the weights such that the output values at the output neurons equal target values at a specific time. It can be used to train Elman networks. We use BPTT when training the RNNs for our SURF feature vector system.

The algorithm was independently derived by numerous researchers [Rumelhart *et al.* 1985] [Mozer 1995] [Robinson and Fallside 1987] [Werbos 1988]. Elman used truncated backpropagation [Elman 1990]. This means that $y_j(t - 1)$ was regarded as an additional input. Any error at the state layer, $\delta_j(t)$ was used to modify weights from the additional input slot.

To train an RNN using BPTT, training data is required. The training data is a set of input-output pairs, such that each input has a corresponding output, $((a_0, y_0), (a_1, y_1), \dots, (a_{n-1}, y_{n-1}), (a_n, y_n))$ where a_i is the input value at time i and y_i is its corresponding output value. x_i is the input to the hidden neurons from the context neurons at the previous time step. An initial value must be given to x_0 . BPTT unfolds a network through time as shown in Figure 5.2. The RNN in the image has two regular feedforward networks, f and g . As the network unfolds, the unfolded network contains k instances of f and one instance of g . In the example shown in Figure 5.2, the network has been unfolded to a depth of $k = 3$. As the RNN trains, it does so in a similar manner as a feedforward NN with backpropagation, except that for each epoch, all observations y_t must be run through in sequential order.

Each training pattern consists of $(x_t, a_t, a_{t+1}, \dots, a_{t+k-1}, y_{t+k})$.

Each action for k time steps are required because the network has inputs at each level. Backpropagation updates the weights as each training pattern is given. After each pattern is given, the weights are updated and then each instance's weights $f(f_1, f_2, \dots, f_k)$ are averaged so that the weights are all the same. x_{t+1} is a function of the previous time step such that $x_{t+1} = f(x_t, a_t)$.

5.6 RNN Architecture

An important factor to consider when designing the RNN is the hidden layer architecture. The hidden layer consists of hidden neurons, and has an activation function applied to it. This layer falls between the input and output layer. The first question that arises is how many hidden layers have to be used when dealing with a complex problem. If the data is linearly separable, then there is no need to use a hidden layer as the activation function can be implemented to the input layer. However, if the data is not linearly separable, then one has to use a hidden layer. Also, the number of neurons that should be kept in each hidden layer need to be determined. If the number of neurons is too small for the complexity of the problem, then underfitting may occur. Underfitting occurs when there are too few neurons in the hidden layers to adequately detect the signals in a complicated data set. If too many hidden neurons are present in the network then overfitting may occur. Several methods are used to determine both the number of hidden layers and hidden neurons in each layer, but none of these can provide an optimal solution for the number of hidden layer as well as number of neurons in each hidden layer.

Karsoliya [2012] found that one hidden layer will be used when any function that contains a continuous mapping from one finite space to another. Two hidden layer can represent an arbitrary decision boundary to arbitrary accuracy with rational activation functions and can approximate any smooth mapping to any accuracy. NNs that consist of many layers can represent deep circuits, however training these

deep networks has always been seen as a challenging task. Empirical research has found that deep networks generally performed no better, and often worse, than NNs with one or two hidden layers [Bengio and LeCun 2007]. Thus, one hidden layer is considered sufficient in the task of FER.

The hidden neurons within the hidden layer can greatly influence the error of the output neurons, to which they are connected. The stability of any RNN is estimated by error. A smaller error reflects better stability, and a larger error reflects worse stability. Adding more neurons to the hidden layer is not necessarily better, as too many hidden neurons can result in overfitting. This means that the RNNs have overestimated the complexity of the target problem and greatly degrades the generalization capability and results in incorrect predictions [Ke and Liu 2008]. If there are too few hidden neurons, a high training error will result, and high generalization error due to high statistical bias and underfitting. If there are too many hidden neurons, a low training error will result, but a high generalization error due to high variance and overfitting can occur. Geman *et al.* [1992] detail how the number of hidden neurons affects the bias and variance tradeoff. The quality of the predictions made by a network is measured in terms of the generalization error. Generalization performance varies over time as the network adapts during training.

There have been several studies to determine the optimal number of neurons in the hidden layer. Early work included that of Sartori and Antsaklis [1991], Arai [1993], and Li *et al.* [1995]. Hagiwara [1994] presented a method to find the optimal number of hidden neurons. The drawback to this system, like many other systems, is that there is no guarantee that the network with a given number of hidden neurons will find the correct weights. According to the statistical behaviours of the output of the hidden units, if a network has a large number of hidden neurons, a linear relationship is obtained in the hidden neurons. Fujita [1998] proposed a method to determine the number of hidden neurons depending on the output error. Hidden neurons are added one by one until an acceptable error is reached. Keeni *et al.* [1999] presented a method to determine the number of hidden neurons which was used in the prediction of cancer cells. Training starts with a large amount of hidden neurons and then the neurons are pruned once the network is trained. However, pruning does not always improve generalization. The initial weights for the input to hidden layer and the number of hidden neurons are determined randomly. The disadvantage of the system is that there is no optimal solution. A method proposed by Onoda [1995] obtains the minimal errors when increasing the number of hidden neurons. Islam and Murase [2001] found that the generalization ability of network may be degraded when the number of hidden neurons is too large, since the hidden neurons may have some spurious connections. Zhang *et al.* [2003] developed a method for determining the number of hidden neurons a three-layer neural network, based on the number of input neurons. They found that the output error decreases when hidden neurons are increased. However, the number of hidden neurons should not be too large for a heuristic learning system. Jiang *et al.* [2008] presented the lower bound on the number of hidden neurons. The lower bound can accelerate the learning speed, and the upper bound gives a stopping condition for a constructive learning algorithm. Ke and Liu [2008] found that the optimal number of hidden layers and hidden neurons depends on the complexity of network architecture, the number of input and output units, the number of training samples, the degree of the noise in the sample data set, and the training algorithm. The necessary numbers of hidden neurons approximated in hidden layer using a multilayer perceptron were found by Trenn [2008]. It is important that the network is simple, scalable, and adaptive. They found that the number of hidden neurons is between the number of inputs and number of outputs. It is found in many system that a tradeoff exists between the number of neurons and the stability of the network, such that if the number of hidden neurons becomes too large, the output of neurons becomes unstable, and if the number of hidden neurons becomes too small, the hidden neurons becomes unstable again.

The optimal number of hidden neurons depends on the number of input units, the number of output units, the number of training cases, the amount of noise, how complex the classification is, the network architecture, the hidden unit activation function, the training algorithm, and regularization. Many methods exist, however, there is often no guarantee of finding the optimal solution. The number of hidden neurons seems to be dependant on the type of problem that is being solved. Many systems use a method of trial and error, which tests differing hidden neuron numbers until a minimum error is achieved. The

rule of thumb is that hidden neurons must fall between the number of input and output neurons. The disadvantage of adding more hidden neurons is a longer training time. Other existing techniques consist of network growing and network pruning [Wang and Huang 2011], [Zeng and Yeung 2006]. The growing algorithm allows the adaptation of the network structure. It begins with a low amount of hidden neurons, and then adds neurons, or grows. The disadvantage is that this system is time consuming and there is no guarantee of finding the optimal amount of hidden neurons. For our system, we will be determining the optimal number of hidden neurons by trial and error, which consists of training networks with differing hidden neuron numbers and then estimating the generalization error of each.

When training the RNN, the learning rate applies a greater or lesser portion of the respective adjustment to the old weight. If the factor is set to a large value, then the RNN may learn faster, but if there is a large variability in the input set then the network may not learn well or at all. In standard backpropagation, too low a learning rate makes the network learn very slowly. Too high a learning rate makes the weights and objective function diverge, so there is no learning at all. If the objective function is quadratic, as in linear models, good learning rates can be computed from the Hessian matrix [Bertsekas and Tsitsiklis 1995]. If the objective function has many local and global optima, as in typical feedforward NNs with hidden units, the optimal learning rate often changes dramatically during the training process, since the Hessian also changes dramatically. Finding the optimal learning rate is usually a tedious process requiring much trial and error. Many algorithms try to adapt the learning rate, but any algorithm that multiplies the learning rate by the gradient to compute the change in the weights is likely to produce erratic behaviour when the gradient changes abruptly [Riedmiller and Braun 1993], [Kandil *et al.* 1993]. Often, the learning rate is determined by trial and error. Experiments are run with different learning rates. If the weights and errors change very slowly, a larger learning rate is used. If the weights fluctuate wildly and the error increases during training, a lower learning rate is used. The learning rate is between 0 and 1, and often a learning rate of 0.1 is initially used.

5.7 Applications

RNNs are a type of network where the connections between the units form a directed cycle. This directed cycle creates an internal state that allows the RNN to exhibit temporal behaviour. The ability of RNNs to treat and store time dependent information enables them to learn space-time relationships. This ability makes RNNs useful in speech recognition, where the examples are space-time patterns [Ahmad *et al.* 2004]. The goal of speech recognition is to design automatic systems which are capable of interpreting the vocal signs coming from a human speaker in terms of linguistic categories. The task of speech recognition is divided into two steps: firstly, feature extraction, where the features are extracted from the stream of data and, secondly, design of a system to model the extracted features. Feature extraction is important as the speech sequences contain irrelevant information like background noise.

Another application of a RNN is signature verification [Tiflin and Omlin 2003]. These systems are designed for authentication of a signature. The verification system is composed of two units. The first unit is the pre-processing unit which extracts the feature of a signature which include the timing information and the positioning of the pen point when making signatures. The second unit is a modelling unit which learns the extracted features. The system is designed to detect the forged signature among the genuine ones. For this, the system is first trained on a set of sample signatures which contain both the genuine and forged signatures. Once the training is complete the system is tested on the test data, where it should be able to detect the forged one and a genuine one.

RNNs have been successful in financial forecasting. Tino *et al.* [2001] used time series data and other factors and fed it to the RNN so that the network can capture the rules of the how the currency exchange rates changes. The trained network is then able to forecast the exchange rates between different foreign exchanges.

The use of RNNs was successful for the dynamic recognition of facial expressions and AUs [Butko *et al.* 2011]. Vadapalli [2014] used RNNs for the recognition of six upper face and five lower face AUs using Gabor filters for feature extraction. They applied different dimensionality reduction techniques

including frequency scale selection, local Gabor filters and PCA, and studied their effect on the classification performance of RNNs. Results showed that local Gabor filters performed the best (85% average performance) for the six upper face AUs, whereas the use of all the Gabor filters performed the best (82% average performance) for the five lower face AUs. Network regularization through weight decay improved the classification performance (86% average performance for six upper face AUs and 86% for the five lower face AUs).

5.8 Conclusion

This chapter serves as an introduction to RNNs, which will be used to develop our system to recognize AUs. Our system will make use of an Elman RNN to recognize FACS AUs using SURF descriptors as feature vectors, which are the input to the RNN. An Elman RNN is used as it provides memory of one time unit, which is sufficient for facial expression recognition. A key point to note from this chapter is that RNNs make use of temporal data. This means that the output from a previous time step is used as the input to the next time step. This will be beneficial when classifying AUs that have shown to develop over time in an image sequence or video. This will also be beneficial when trying to recognize AUs when the face has moved out-of-plane in image sequences.

Chapter 6

Methodology

6.1 Introduction

This chapter details the methodology taken for this research, which includes how the RNN is structured to recognize FACS AUs. Data sets are a sequence of images, which show the face from neutral to the depiction of an AU or AUs. Image sequences are split into a number of frames, each showing a face in a particular step of the creation of an AU. We use SURF to extract keypoints from the face in each frame, which we then use to create SURF descriptors. These SURF descriptors are used as feature vectors to train an RNN to recognize AUs. The structure of an RNN includes the number of hidden layers, as well as the number of neurons in each hidden layer. We also need to determine the optimal learning rate to be used for the RNN. Once we have determined how to structure the RNN, we run experiments on how many keypoints need to be extracted from a face to provide the best recognition rate. This chapter details the methodology.

6.2 Research Hypothesis

Vadapalli [2014] made use of RNNs to recognize FACS AUs. This system made use of Gabor Filters to extract features. However, the use of Gabor filters can suffer from the high dimensionality problem. We propose the use of the SURF method to extract keypoints from the face in a sequence of images where an AU (or AUs) are formed from a neutral face. We use these keypoints to generate SURF descriptors which form feature vectors. A recurrent neural network is used to classify features by taking advantage of the temporal data obtained from the image sequences. The feature vector obtained from the SURF method is far smaller than feature vectors obtained from Gabor filters and other common feature extraction techniques. The following research hypothesis is reached:

1. **Recurrent neural networks will be able to classify FACS AUs in an image sequence using SURF to extract features and SURF descriptors as feature vectors.**
2. **SURF descriptors and RNNs can be used to recognize FACS AUs in an image sequence where the face moves out-of-plane by taking advantage of the temporal processing abilities of RNNs.**

6.2.1 Research Questions

1. Can SURF Descriptors extract enough information from the face for recognizing AUs?
2. Is it possible to train a recurrent neural network to recognize AUs using the SURF feature vectors?
3. How many keypoints need to be extracted from the face in order to optimally recognize AUs?
4. Can SURF descriptors and RNNs be used to recognize FACS AUs when the face moves out-of-plane?

6.3 Assumptions

Some assumptions that are initially known are:

- The input type is an image sequence split into a number of frames of varying length.
- A sequence depicts the formation of an AU or group of AUs from a neutral face.
- This system will be recognising FACS AUs.
- The database used for Phase I of testing is the Cohn-Kanade database, which contains FACS AU annotated video sequences.
- Phase II of testing takes place on the UNBC-McMaster database, which contains image sequences where the face moves out-of-plane.

6.4 Databases

We discussed FACS AUs in Chapter 2 as well as databases that are FACS annotated. The RNNs will be developed and tested using the Cohn Kanade database. This database contains image sequences from 123 subjects who perform twenty-three facial displays that include AUs and AU combinations. The camera is frontally oriented and only very small head motion is allowed. In this database, AUs are mostly shown to be present at high intensity (such as intensity D and E). Once the RNN is developed it is trained using images from the Cohn Kanade database. Phase I of testing also takes place on this database to determine the false positive rate, true positive rate and recognition rate of AUs.

The UNBC-McMaster database consists of videos captured of participant's faces who were suffering from shoulder pain. To elicit pain, they performed a sequence of active and passive range-of-motion tests to both their affected and unaffected limbs. Each frame of the image sequences of this data was AU coded by certified FACS coders. Two hundred image sequences across 25 subjects have been made publically available. Unlike the Cohn-Kanade database, which contains frontal face images, the UNBC-McMaster database contains image sequences where the face moves out-of-plane due to the effect of pain on humans. Phase II of testing takes place on the UNBC-McMaster database to determine the recognition rate of AUs where the face moves out-of-plane for some part of the image sequence.

6.5 Methodology Overview

In Chapter 3 we outlined the facial expression recognition process. Features must be extracted from the face to form feature vectors, and these are used to train an RNN. In order to obtain feature vectors, SURF keypoints are extracted from the face. A variable number of keypoints are extracted depending on the sample quality and strength of keypoints. Thus a standard number of keypoints must be taken from each image in a sequence across all samples. To determine the number of keypoints extracted, experiments are run with differing numbers of keypoints on samples from the Cohn-Kanade database to determine the true positive rate and false positive rate with different AUs. The best performing number of keypoints are chosen. Keypoint descriptors are obtained, which are normalized to form descriptors and consequently feature vectors. Before we can use these feature vectors, we must determine the structure of the RNN. This includes the number of hidden layers, the number of neurons within the hidden layers, and the learning rate of the RNN. This is accomplished by further experimentation to optimize performance. Once these are determined, the experimentation process can begin to determine the recognition rate of AUs from the Cohn-Kanade database using SURF feature vectors. After the RNN is trained to classify AUs, we test the recognition rate using samples from the UNBC McMaster database. Below is a summary for the methodology undertaken:

1. Extract SURF keypoints from the face. The number of keypoints extracted differ depending on the requirement of the experiment. Samples contain data sets of 20, 30, 40, 50, 60, 70, 80, 90 and 100 extracted keypoints.
2. Determine the learning rate and hidden layer structure of the RNN with a sample set of 50 extracted keypoints.
3. Once RNN structure is determined, determine optimal number of keypoints by testing the RNN with differing extracted keypoint numbers.
4. Using the optimal number of keypoints, run experiments on the Cohn-Kanade database to determine recognition rate of twelve AUs.
5. Using the RNNs trained to recognize AUs, determine the recognition rate using samples from the UNBC-McMaster database.

6.6 Obtaining Feature Vectors - Cohn Kanade Database

We are testing six upper face AUs and six lower face AUs. The upper face AUs tested are AU1, AU2, AU4, AU5, AU6, and AU7. The lower face AUs tested are AU9, AU15, AU17, AU20, AU25, AU27. The reason for choosing these particular AUs is that there are a sufficient number of samples available for these AUs in the Cohn Kanade database.

An image sequence shows a subject which starts with a neutral face and over several frames develops into the positive depiction of an AU or group of AUs. The subjects are of different age, gender, race and ethnicity. In each frame in an image sequence, we locate the eyes and mouth of the subject in each frame, and crop the face accordingly such that the image contains only the face and does not have any background noise. Thereafter, the SURF keypoints are located in each frame by Hessian Approximation. The keypoints are ordered from highest response to lowest response. Response indicates the strength of the keypoint. Each keypoint data set has a coordinate, diameter of the keypoint, orientation angle, and response. Depending on the number of keypoints that is required for an experiment, the keypoints with the highest response are chosen. For example, if the experiment requires 50 keypoints, the 50 keypoints with the highest response are chosen and the keypoints thereafter were discarded. Then the keypoint descriptors are obtained for each keypoint in each image in the sequence. The descriptor for each keypoint is a vector of length 64. The descriptors of all keypoints in an image are concatenated to create the feature vector. Thus, for 50 keypoints, the feature vector is of size $50 \times 64 = 3200$. The size of the feature vector determines the number of input neurons to the RNN.

We use OpenCV as the starting point for the face detection and feature extraction system. OpenCV (Open Source Computer Vision) is a programming function library mostly used for real-time computer vision, and was developed by the Intel Russia research center, and is now under the support by Willow Garage and Itseez [Bradski 2000]. It is free, cross-platform, and operates under the open source BSD license. We customize the OpenCV source code to be optimized for the purposes of our experiments.

Figure 6.1 shows an extract of an image sequence from the Cohn Kanade database in which the subject starts with a neutral face and over the sequence of frames develops AUs 6, 7 and 12. In each image in the sequence, 60 keypoints are extracted and mapped. As the corners of the subject's mouth pull in subsequent frames, more keypoints are concentrated around the mouth area. AU12 is the lip corner puller. More keypoints concentrated around the lip corner area are an indication that AU12 is present. For each image sequence, we have keypoints extracted in each frame, and SURF descriptors are determined for each keypoint. These are considered the feature vectors to train and test our system.

6.7 Recurrent Neural Network Structure

When structuring the RNN, it must be determined how many hidden layers the RNN has, as well as the number of neurons in each hidden layer. Additionally, the learning rate for the RNN must be determined.

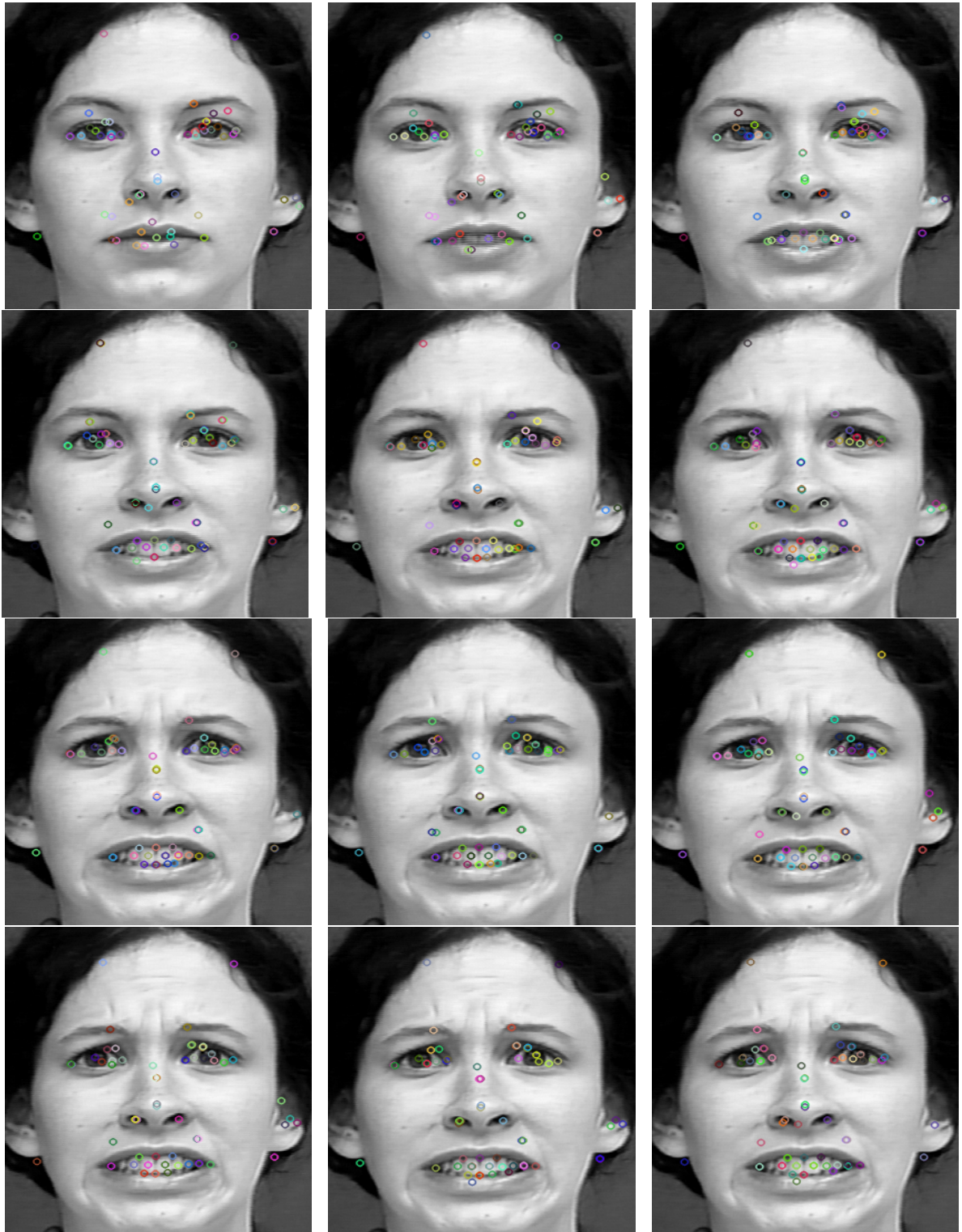


Figure 6.1: Image sequence showing a neutral face in the first frame and the development of AUs 6, 7, and 12 in the final frame with 60 keypoints extracted in each frame.

NNs that consist of many layers can represent deep circuits, however training these deep networks has always been seen as a challenging task. Empirical research has found that deep networks generally performed no better, and often worse, than NNs with one or two hidden layers [Bengio and LeCun 2007]. Thus we have one hidden layer in our RNN.

We need to determine the optimal number of neurons in the single hidden layer. The optimal number of hidden neurons depends on the number of input units, the number of output units, the number of training cases, the amount of noise, how complex the classification is, the network architecture, the hidden unit activation function, the training algorithm, and regularization. The only way to determine the optimal number of hidden neurons is to train networks with differing hidden neuron numbers and then estimate the generalization error of each. If there are too few hidden neurons, a high training error will result, and high generalization error due to high statistical bias and underfitting. If there are too many hidden neurons, a low training error will result, but a high generalization error due to high variance and overfitting can occur. Geman *et al.* [1992] detail how the number of hidden neurons affects the bias and variance tradeoff. Therefore we test our network with several different hidden neuron numbers and determine the error rate. We choose the local error minima.

When training the RNN, the learning rate applies a greater or lesser portion of the respective adjustment to the old weight. If the factor is set to a large value, then the NN may learn quicker, but if there is a large variability in the input set then the network may not learn well or at all.

We develop an RNN with the optimal learning rate, as well as the optimal number of neurons in the hidden layer. We test our RNN with several different learning rates and numbers of neurons on data sets that show the development of AUs. The weight initialization range is between -1 and 1. For standardization purposes, we extract 50 keypoints from each image in an image sequence across all samples.

We refer to true positive rate, false positive rate and recognition rate. These are defined as follows:

$$\text{True positive rate} = \frac{\text{True positives}}{\text{Total positive samples}} \quad (6.1)$$

$$\text{False positive rate} = \frac{\text{False positives}}{\text{Total negative samples}} \quad (6.2)$$

$$\text{Recognition rate} = \frac{\text{True positives} + \text{True negatives}}{\text{Total samples}} \quad (6.3)$$

We wish to maximise true positive rate and minimize false positive rate.

We test with a single hidden layer, and with one to twenty neurons in the single hidden layer. We test with learning rates 0.1 and 0.01. When testing with learning rate 0.001, the RNN does not converge (i.e. does not have an average error of less than 1%) after the maximum number of iterations allowed.

Figure 6.2 and Figure 6.3 show the results of testing with different numbers of neurons and a learning rate of 0.1 and 0.01. The x-axis shows the number of neurons in the hidden layer. Figure 6.2 shows the percentage of false positives and true positives with different numbers of neurons in the hidden layer and a learning rate of 0.1. Figure 6.3 shows the true positives and false negatives with a learning rate of 0.01. For each AU we test, training sample sets consist of 50% positive (i.e. AU is present) and 50% negative (i.e. AU is not present) samples. The RNN has two output classes, zero when the AU tested is not present, and one when the tested AU is present. The number of inputs depends on the number of keypoints extracted from the face. Each SURF keypoint descriptor is a vector of length 64. Therefore, for 50 keypoints, the input size for the RNN is $64 \times 50 = 3200$ inputs. Five-fold cross validation takes place to remove the effect of non-performing samples. This means that five tests are run, each training with 80% of the sample set and testing with 20% of the sample set. For each of the five experiments, different training and testing sets are used. The testing sample set, which is 20% the size of the sample set, also consists of 50% positive and 50% negative samples. The samples are randomly split into subsets, such that the correct ratio of positive to negative samples is maintained.

From Figure 6.3 we can see a large variability in results when a learning rate of 0.01 is used. As we add neurons, the true positive rate decreases and the false positive rate increases. Although a learning

Learning rate: 0.1

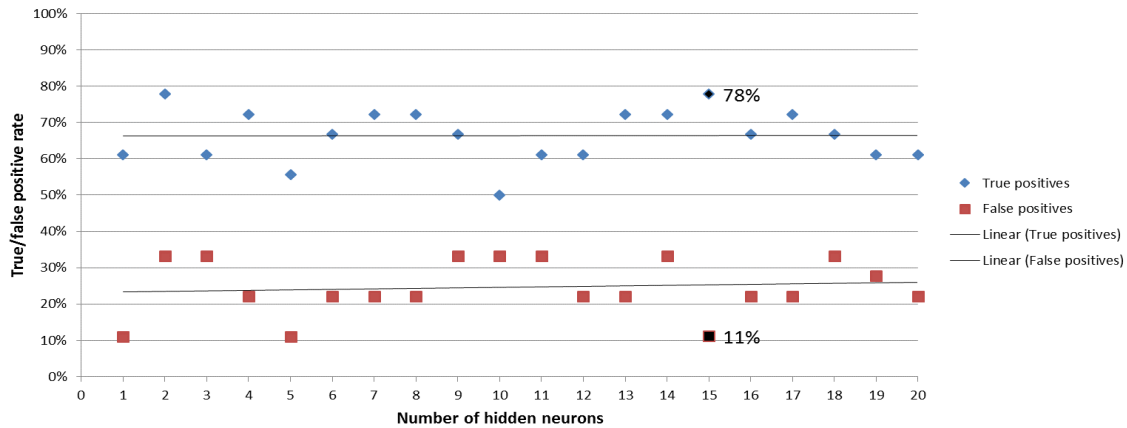


Figure 6.2: Learning rate of 0.1.

Learning rate: 0.01

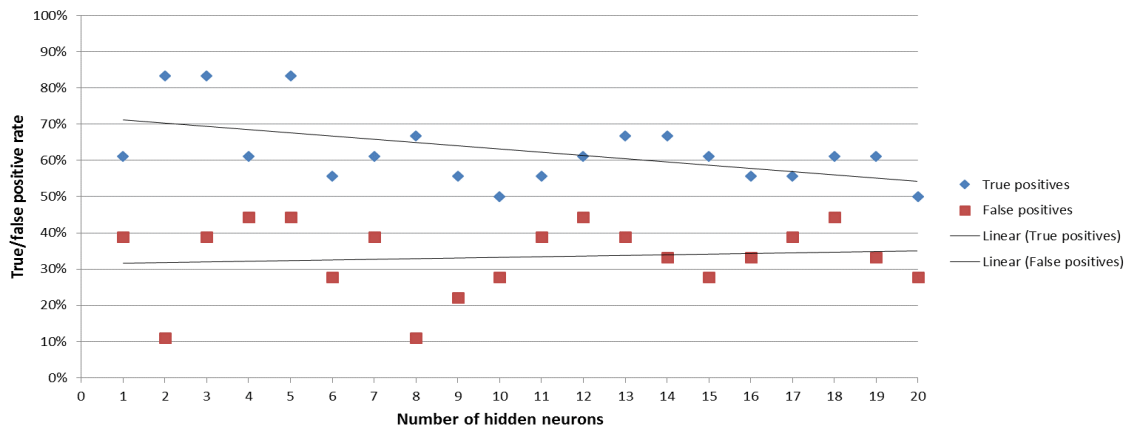


Figure 6.3: Learning rate of 0.01.

rate of 0.01 shows the highest rate of true positives in instances, the average rate of false positives is high with 31%. From Figure 6.2 we can see a smaller variability in results (true positive rate and false positive rate sees less change as more neurons are added), as well as a lower average rate of false positives (24%). The average rate of true positives with a learning rate of 0.1 (66%) is higher than the average rate of true positives with a learning rate of 0.01 (64%). Due to this, we choose to use a learning rate of 0.1. Within the results in Figure 6.2 we see that fifteen hidden neurons provides the highest rate of true positives (78%) and the lowest rate of false positives (11%). Thus, we choose to use fifteen neurons in the single hidden layer with a learning rate of 0.1.

For experimental purposes for this research, our RNN is structured with a single layer that consists of fifteen neurons and with a learning rate of 0.1. The weight initialization range is [-1, 1]. The maximum number of epochs, which is the presentation of all the training samples to the network, is set to 500. The training stops when the pre-defined system error is reached (1%) or the learning process continues until the maximum epochs are elapsed. A value of 0.5 or greater is taken as the target AU being present, and a value less than 0.5 is taken as the target AU not present.

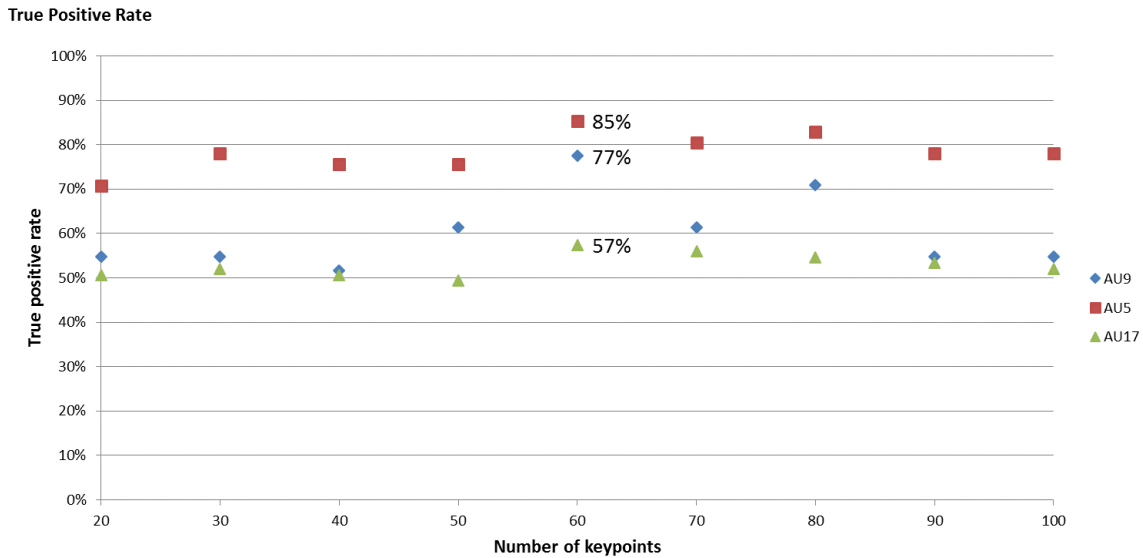


Figure 6.4: True positive rates with differing numbers of extracted keypoints.

6.8 Optimal Number of Keypoints

When extracting keypoints from an image, the number of keypoints extracted is variable and depends on the type of image, the image quality, and the intensity of changes in the face (such as furrows or wrinkles). For the development of the RNN, we initially used 50 keypoints extracted from the face as a starting point to determine the learning rate and number of hidden neurons. However, this number of keypoints may or may not provide the highest rate of true positives and lowest rate of false positives when testing on the Cohn Kanade database. Du *et al.* [2009] found that approximately 30 to 100 keypoints per image provide the best recognition rate on the JAFFE database.

We test how many keypoints extracted from a frame in an image sequence will yield the highest rate of true positives and lowest rate of false positives. After the optimal structure of the RNN is determined by using a standard number of 50 keypoints extracted per image, we test this RNN with differing keypoints numbers. We run experiments which test the rate of true positives and the rate of false positives when recognizing an upper face AU (AU5), a middle face AU (AU9), and a lower face AU (AU17) when different numbers of keypoints are extracted from the face. We test 20, 30, 40, 50, 60, 70, 80, 90, and 100 keypoints extracted from each frame in an image sequence, in line with the findings by Du *et al.* [2009].

The true positive rate results are shown in Figure 6.4 and the false positive rates are shown in Figure 6.5 for the three AUs tested. The x-axis contains the number of keypoints extracted. Results show that 60 keypoints provides the highest rate of true positives for AU5 (85%), AU9 (77%) and AU17 (57%). Additionally, 60 keypoints result in the lowest rate of false positives for AU9 (26%) and AU17 (35%). However, 60 keypoints does not provide the lowest rate of false positives for AU5. The tradeoff of choosing the number of keypoints (30 or 50) which provides the lowest rate of false positives for AU5 will result in a much higher rate of false positives for both AU9 and AU17. It also results in a significantly lower rate of true positives in AU5, AU9 and AU17. Thus, we chose 60 keypoints as it provides the best overall result of both true positives and false positives for all three AUs tested.

6.9 Feature Vectors - UNBC-McMaster Database

The UNBC-McMaster database contains image sequences where the subject is exhibiting pain, and due to the nature of pain on a human the face moves out-of-plane in the image sequences. The camera angle is frontal, however the head position of subjects shows some degree of movement out-of-plane. The

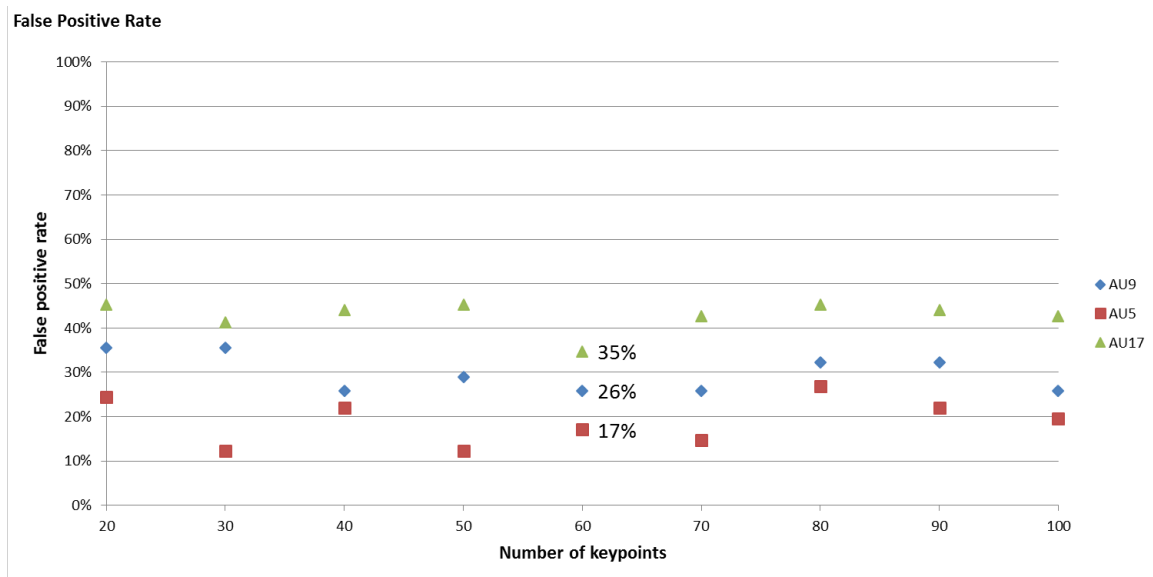


Figure 6.5: False positive rates with differing numbers of extracted keypoints.

images in this database are RGB, therefore they first have to be converted to greyscale and resized to be the same as the Cohn Kanade database (640x490). The images are made larger by resampling, which is a technique to create a new version of the image with a different width and/or height in pixels [Darwish *et al.* 1997]. New pixels are added based on colour values of existing pixels.

The database contains both a PSPI score and the FACS AUs present for each frame within an image sequence. We collect samples where no pain was present, i.e. no active AUs. We also collect samples of subjects exhibiting strong pain, i.e. a PSPI score of three or greater. For these a combination of AUs 4, 6, 7, 9, 10, and 43 are present at varying intensities (intensity A to intensity E).

Since we have shown that 60 keypoints provides the highest rate of true positives and the lowest rate of false positives on average on the Cohn Kanade database, we extract 60 keypoints from each image in image sequences collected in our sample set. These keypoints are then converted into SURF descriptors for our feature vectors.

Our sample set contains sequences where PSPI score is zero or three and greater. Where the PSPI score is three or greater, we look at the AUs present to give that score. We then develop sample sets for AU4, AU6, AU7, and AU9 containing the image sequences where those AUs are present, as well as image sequence where no pain and no AUs are present.

An extract of an image sequence is shown in Figure 6.7 with 60 keypoints extracted from each frame. This face shows some degree of out-of-plane head motion. The image sequence shows the onset of AUs 4c, 6a, 7d, 43 and 50. This gives a PSPI score of 8.

6.10 Challenges

There were several challenges encountered while designing and building the RNN. These came from the nature of the data sets, i.e. the image sequences, and the ability of the RNN to handle these data sets. The main challenge came from the lengths of the image sequences. Each image sequence is split into a number of frames. The number of frames is variable, which means that each sample consists of different numbers of frames, i.e. the number of image frames in an image sequence is not the same for each sample. This meant that the RNN had to handle cases of varying numbers of feature vectors. Each image in an image sequence has a feature vector, so the numbers of feature vectors were variable. However, the number of elements within the feature vector stays constant, and this is determined by the number of keypoints extracted from the face. The number of elements in the feature vector determines the number of inputs for the RNN.

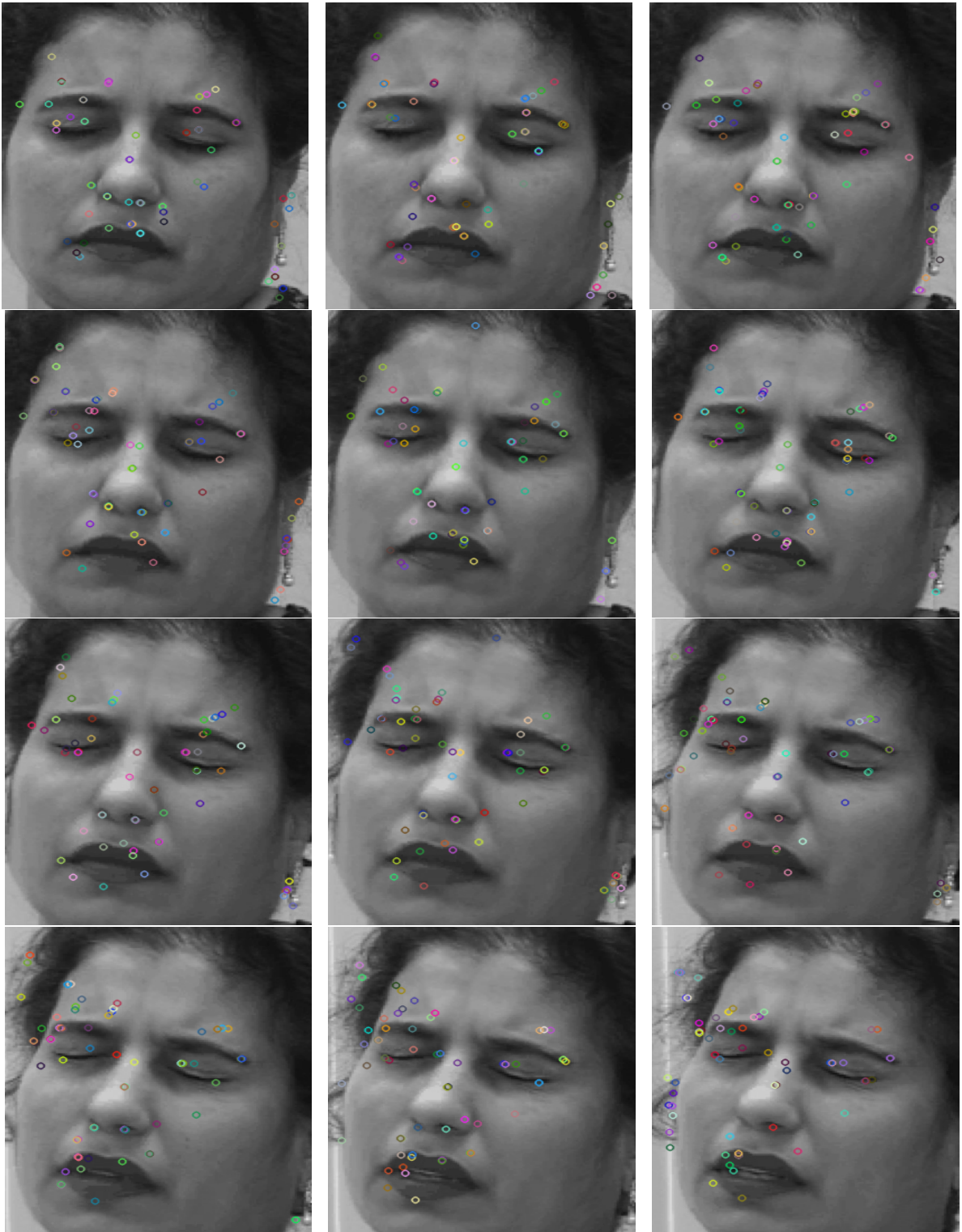


Figure 6.6: Image sequence showing the onset of AUs 4c, 6a, 7d, 43 and 50, and a PSPI score of 8, and with 60 keypoints extracted in each frame.

Other challenges included keypoint extraction from the face. In the Cohn Kanade database, the image quality of samples vary, therefore the varying quality made face detection challenging.

As discussed in Section 3.10, a keypoint (or a feature) is an area of interest in an image. This could be wrinkles or furrows related to the movement or contraction of specific muscles. These keypoints are concentrated around areas that see a change in the face while muscles are moving or contracting. The keypoints are also concentrated around areas of wrinkles or furrows from edges, such as the ear lobes. These keypoints often also have a high response, so are chosen as keypoints to train the RNN. However, these keypoints are not relevant to AU recognition. This means that some keypoints are extracted and chosen to train or test the RNN and they are not relevant to AU recognition. Since we choose keypoints by response, a method to overcome this problem would be to choose keypoints by area of the face only.

6.11 Conclusion

This chapter is an overview of how we get feature vectors from the faces in image sequences to use as inputs to the RNN. We also show how we determine the structure of the RNN in terms of hidden layers, hidden neurons and the learning rate. Once we have data sets and an RNN structure, we can begin experiments to determine the false positive rate, true positive rate and recognition rate of AUs when tested on the Cohn Kanade and UNBC-McMaster databases. From this chapter, it can be seen that 60 keypoints extracted from the face provides the best results in terms of true positive rate and false positive rate when tested on samples from the Cohn Kanade database. Therefore for experimentation purposes hereafter, 60 keypoints are extracted from each image in an image sequence in a sample set. Additionally, it is shown that a single layer RNN with fifteen neurons and a learning rate of 0.1 provides the best results when tested on the Cohn Kanade database. Experiments are now run using this structure and the results of experimentation are shown in Chapter 7.

Chapter 7

Results

7.1 Introduction

Once we have the structure of the RNN by determining the learning rate and number of hidden layer neurons, as well as the data sets that need to be obtained from each image in an image sequence, we can begin experimentation to determine the recognition rate of AUs when using SURF descriptors as feature vectors. We test the RNN on twelve AUs from the Cohn Kanade database, six from the upper face and six from the lower face. This is Phase I of testing. Once we train the RNNs for each AU, we test on an additional database called the UNBC-McMaster database. This is Phase II of testing. In the UNBC-McMaster database, the subjects show some degree of in-plane or out-of-plane head movement. In-plane movement is the movement of the face sideways such that the face is still facing frontwards (for example, tipping the head to the side), while out-of-plane movement is movement left or right such that the face is no longer facing frontwards (for example, a profile view). We detail the results of Phase I and Phase II testing in this chapter.

7.2 Correlation - Cohn Kanade Database

Some AUs are easier to detect and classify than others and recognition rates between AUs differ. Whitehill and Omlin [2006b] examined whether extracting features locally around the eyes, eyebrows, and mouth will result in higher AU recognition rates than when features are extracted globally. Results showed that local feature extraction did not improve recognition rates over global extraction. Global extraction even outperformed local extraction when recognizing AUs of the eyebrow and eye areas. This result was partly attributed to the high correlations between the AUs in the Cohn-Kanade database. We use global segmentation for our system, which means that the presence of AUs on different parts of the face can affect the recognition of other AUs.

Suppose that AU_i is more difficult to classify than AU_j . If it is known that AU_i is highly correlated with another AU_j , then the classifier for AU_i could attempt to classify AU_j instead. This is because the classifier associates AU_i with AU_j . If there are testing samples that contain AU_j , but not AU_i , the classifier could output a one (positive) incorrectly, i.e. a false positive, as it was trained to detect AU_i with a high presence of AU_j . We use global segmentation of the face, meaning that we do not segment into upper and lower face, or into parts of the face such as eyes and mouth. This means that AU_i and AU_j can even be on different parts of the face. By knowing which AUs occur together and analyzing their relationship, we can also determine if some AUs can be misinterpreted as others or if the combination of some AUs change the appearance of other AUs on the face. We can look at instances of false negatives, false positives, true positives and true negatives for each tested AU and by analyzing the correlation between AUs we can determine whether some AUs affect the recognition of others. Table 7.1 shows the relationship between AUs in the Cohn Kanade database. We look at the number of samples of AU_i that also contain AU_j . We see from Table 7.1 that 63% of AU_1 samples also contain AU_2 . We also

	AU1	AU2	AU4	AU5	AU6	AU7	AU9	AU15	AU17	AU20	AU25	AU27
AU1	1.00	0.99	0.33	0.88	0.07	0.16	0.00	0.38	0.24	0.52	0.35	0.88
AU2	0.63	1.00	0.05	0.83	0.01	0.01	0.00	0.07	0.05	0.16	0.28	0.88
AU4	0.35	0.09	1.00	0.15	0.26	0.85	0.88	0.35	0.61	0.63	0.17	0.03
AU5	0.48	0.71	0.08	1.00	0.02	0.05	0.00	0.01	0.01	0.18	0.25	0.74
AU6	0.05	0.01	0.18	0.03	1.00	0.30	0.37	0.01	0.09	0.15	0.26	0.00
AU7	0.12	0.01	0.58	0.06	0.31	1.00	0.86	0.07	0.38	0.39	0.13	0.00
AU9	0.00	0.00	0.28	0.00	0.17	0.40	1.00	0.01	0.24	0.03	0.03	0.00
AU15	0.20	0.05	0.20	0.01	0.01	0.05	0.02	1.00	0.46	0.01	0.01	0.01
AU17	0.26	0.09	0.63	0.03	0.14	0.57	0.76	0.97	1.00	0.06	0.02	0.00
AU20	0.24	0.12	0.27	0.15	0.10	0.25	0.04	0.01	0.03	1.00	0.23	0.00
AU25	0.71	0.91	0.33	0.95	0.74	0.37	0.20	0.03	0.04	0.97	1.00	1.00
AU27	0.47	0.74	0.01	0.72	0.00	0.00	0.00	0.01	0.00	0.00	0.26	1.00

Table 7.1: Correlation between AUs in the Cohn Kanade Database.

see that 99% of AU2 samples contain AU1. We have made values greater than 0.5 to be shown in bold (with the exception of the diagonal), as this indicates a large level of influence between AUs.

7.3 Receiver Operating Characteristic Curves

A Receiver Operating Characteristic (ROC) curve is a graphical plot which shows the performance of a binary classifier system as its discrimination threshold is varied. It plots the true positive rate versus the false positive rate at various threshold settings. The true positive rate is also known as sensitivity, and the false positive rate is (1 - specificity). Thus the ROC curve shows the tradeoff between sensitivity and specificity. The closer the curve is to the left and the top of the ROC space, the more accurate the test. There is a 45-degree line plotted on the ROC space. The closer the ROC curve comes to the 45-degree diagonal, the less accurate the test. For the purposes of our experimentation, results will be shown using ROC curves.

7.4 Results of Phase I Testing (Cohn-Kanade database)

We test six upper face AUs and six lower face AUs. The upper face AUs tested are 1, 2, 4, 5, 6, and 7. The lower face AUs tested are 9, 12, 15, 20, 25, and 27. We do five-fold cross validation testing on the sample sets to take away the effect of non-performing samples. For each AU which we are testing, we gather image sequences in which the AU tested is shown to develop from a neutral face to peak intensity in the final frame. We also gather the same number of image sequences where the AU is not present in the final frame. For training the RNN, the sample set has on average 50% positive (AU present) and 50% negative (AU not present) samples. The testing sample set contains image sequences with the same ratio of positive and negative samples. Therefore, if we are testing AU9, half of the sample set has samples containing AU9, and the other half of the samples have other AUs excluding AU9. Each sample set is randomly split into five smaller sample subsets. Each of the five subsets is used as a testing set, while the rest of the sample set is used for training purposes. The results are then averaged. The reason for five-fold cross validation is to examine which subsets perform better than others, and then investigate the reasons behind this. The average training and testing sample set sizes for each test in five-fold cross validation for each AU are shown in Table 7.2. For example, for AU1, each test in five-fold cross validation has a

	Average training sample size	Average testing sample size	Total sample size
AU1	132	32	164
AU2	96	24	120
AU4	120	30	150
AU5	72	18	90
AU6	88	22	110
AU7	80	20	100
AU9	52	14	66
AU15	64	16	80
AU17	120	30	150
AU20	40	10	50
AU25	164	40	204
AU27	88	22	110

Table 7.2: Sample set sizes - Cohn Kanade database.

training sample size of 132 and testing sample size of 32. For each test, a different subset of samples is used. The sample sizes shown in Table 7.2 are not a complete subset of the Cohn Kanade database. This is due to the inability of samples being used for training or testing purposes. Samples are unable to be used if 60 keypoints cannot be extracted from the face in image sequences. In some cases, there is too much noise for 60 keypoints to be extracted, or the quality of the images does not allow 60 keypoints to be detected and extracted. Additionally, we look at samples of AUs where the AU is shown to be at peak intensity (or almost peak intensity) in the final frame. For incidents where the AU is not at peak intensity in the final frame, we do not use this sample for training or testing purposes. The reason for not using samples which are shown at low AU intensity is because there are very few samples (if any) for each AU at low intensities. Therefore, there is not enough low intensity data from the Cohn Kanade database to provide enough information for training purposes.

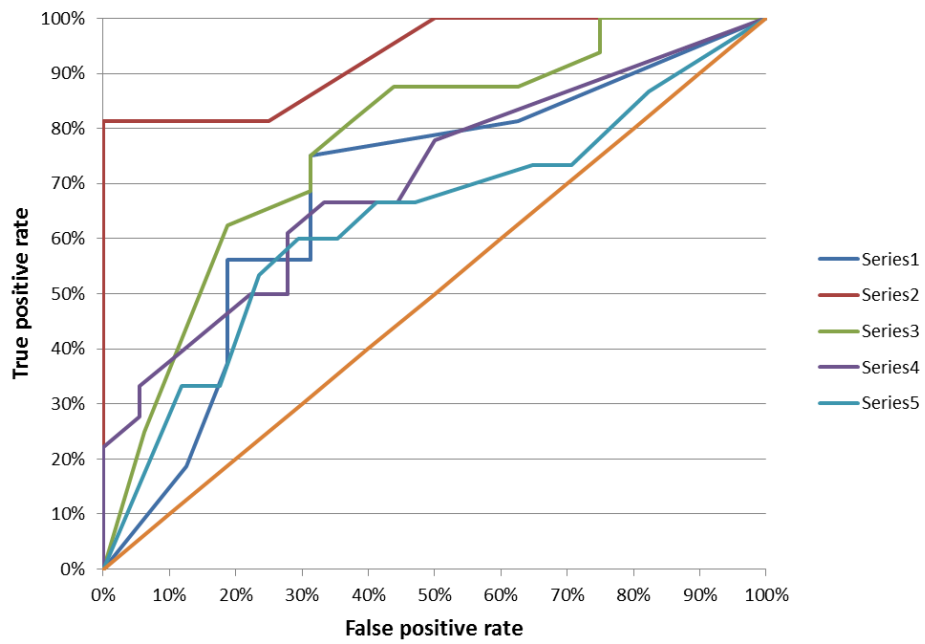
We show the ROC curve for each of the five tests taken in five-fold cross validation for each AU. We also show the average ROC curve for each AU. The recognition rate with a threshold of 0.5 is shown in Table 7.3. This means that for an output of less than 0.5, we consider it a negative result (AU not present). For an output of 0.5 or more, we consider it a positive result (AU is present). It can be seen, on average, that the upper face AUs have a higher true positive rate and lower false positive rate than the lower face AUs.

Detailed results of experimentation are shown in Appendix B. We show results at certain thresholds for each of the five tests, as well as the average results. Tables B1 to B12 contain the results of the Cohn Kanade database experimentation. The threshold (TH), is shown only at given intervals. For example, at a threshold of 0.05, test 1 achieved a 67% false positive (FP) rate, and a 93% true positive (TP) rate.

7.4.1 Results for AU1

AU1 is the inner brow raiser. From Table 7.3, AU1 sees a true positive rate of 65.43%, a false positive rate of 25.3% and a recognition rate of 70.01%. Detailed results at certain threshold settings are shown in Table B1. In Figure 7.1, we see the results of each test taken during five-fold cross validation as well as the average results. From the five tests, we see one particularly well performing test which is close to the top left corner of the plane (series 2). At a threshold of 0.5, this sample set sees a false positive rate of 6.25% and a true positive rate of 81.25%. There are two sample sets with poor false positive rates which increase the average false positive rates significantly (series 3 and series 5). The three sets with well performing false positive tests have false positive rates below 20%.

AU1 results



AU1 average

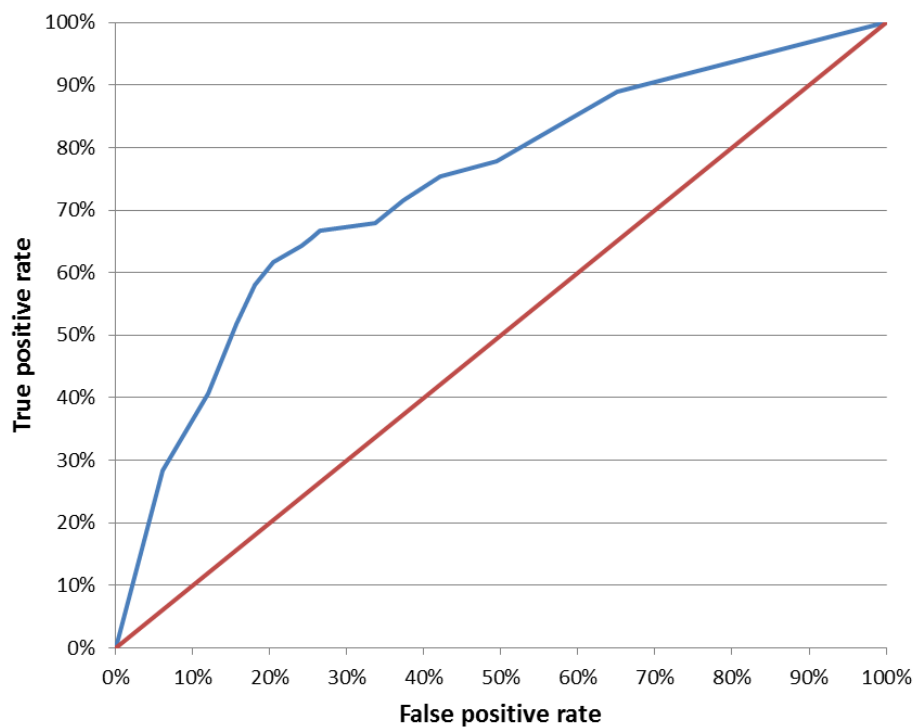


Figure 7.1: Results of AU1 testing

	True positive rate	False positive rate	Recognition rate
AU1	65%	25%	70%
AU2	80%	19%	81%
AU4	63%	32%	65%
AU5	88%	22%	83%
AU6	79%	15%	82%
AU7	69%	25%	72%
AU9	74%	29%	73%
AU15	62%	49%	56%
AU17	58%	30%	64%
AU20	56%	44%	56%
AU25	68%	45%	62%
AU27	86%	16%	85%

Table 7.3: Results of Phase I Testing.

Upon inspection of the poor and well performing false positive test sets, we see that in most cases the false positive samples contain AU25, while the true negative samples do not contain AU25. This implies that the AU1 classifier is classifying samples containing AU25 as positive when it should be negative. From Table 7.1, we see that the AU1 sample set has 71% instances of AU25. As a result, the classifier is being trained with 71% of the AU1 samples also containing AU25. This could alter the ability of the classifier to distinguish between AU1 and AU25.

Additionally, there are three sample sets with poor performing true positive rates between 50% and 60% (series 1, series4, and series 5) . The two test sets with high true positive rates are 81.25% (series 2) and 87.5% (series 3). Upon inspection of the false negatives, we see that there are combinations with AU1 which change the appearance of AU1. These combinations are some part of the reason that our RNN is outputting a false negative. The AU1 classifier does not correctly classify these samples as true. The first combination is 1+2+4. This combination pulls the brows upwards and together. The second combination is 1+2, which pulls the entire eyebrow upwards and results in the eyebrow shape becoming curved and arched. The combination 1+4 maintains the raising action from AU1, but draws the eyebrows together and the lowering effect of AU4 is counteracted or restricted to the outer portions of the brow. The last appearance change combination is 1+2+5, which raises the inner and outer portions of the brow and raises the upper eyelid. These combinations are found throughout our AU1 sample set. The most common misclassified combination is 1+4, which accounts for the most false negatives and thus the low true positive rate. From Table 7.1, we see that 35% of the AU1 sample set contains the combination of 1+4. Figure 7.2 and Figure 7.3 are examples of the effect of 1+4 on the face. Figure 7.2 shows a neutral face on the left, and the right image shows activation of 1+4+7+11+20+25. Figure 7.3 shows a neutral face and activation of 1+4+6+12+20+25. Figure 7.4 shows a neutral face on the left, and right image shows 1+2+5+25+27. The appearance of AU1 in these samples differs, and keypoints extracted also differ between samples with the combination of 1+4 and 1+2. Figure 7.2 and Figure 7.3 show the effect of the combination of 1+4, while Figure 7.4 shows AU1 without AU4. These figures show the first frame in the image sequence on the left, and the last frame of the image sequence on the right. The appearance change combinations cause for a different appearance of AU1 than when it is alone or with other AUs, and thus different keypoints are extracted from the images. This makes it challenging for the classifier to recognize AU1. We see this effect in the images in Figure 7.2 and Figure 7.3, which show a different appearance of AU1 than when it occurs in Figure 7.4.

When extracting keypoints from the face, in some cases if there is large mouth movement then more

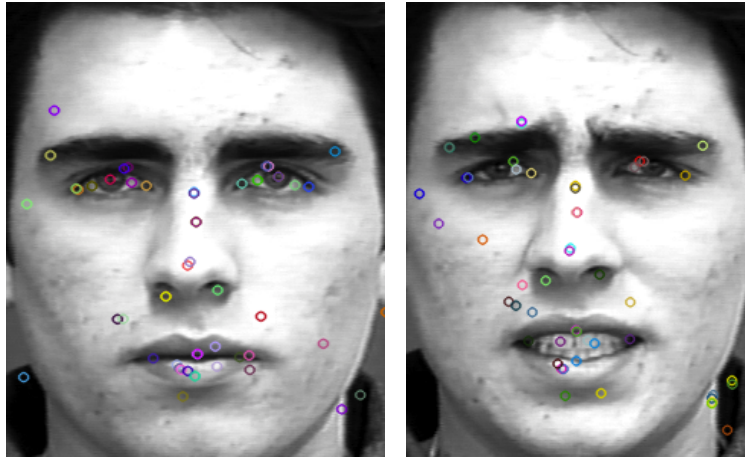


Figure 7.2: Neutral face (left) and face activating 1+4+7+11+20+25 (right).

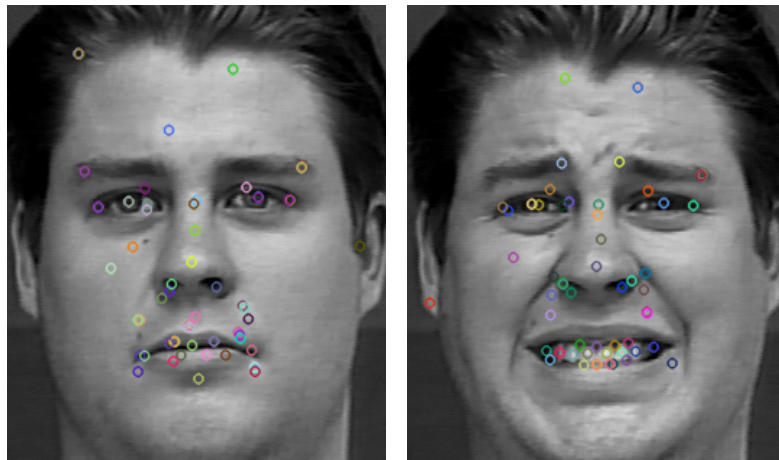


Figure 7.3: Neutral face (left) and face activating 1+4+6+12+20+25 (right).

keypoints concentrate around the mouth and fewer concentrate around the eyes. Thus, in cases like these we also see many incidences of false negatives as there is not enough information received from the eye area to adequately train the RNN to pick up the existence of AU1.

7.4.2 Results for AU2

AU2 is the outer brow raiser. It sees good performance with a true positive rate of 79.66%, a false positive rate of 18.64%, and an overall recognition rate of 80.51%. This is a promising result for AU2. Among the five experiment results seen in Figure 7.5, we see two sample sets with very low false positive rates (series 2 with 0%, and series 3 with 8.33%), as well as two samples with extremely high true positive rates (series 4 with 91.67% and series 5 90%). We also see one sample set with a low true positive rate of 58.33% (series 1), and a sample set with a high false positive rate of 33.33% (series 5). Table B2 shows detailed results at certain threshold settings. We investigate the reasons for the varied results.

From Table 7.1, we see that AU2 is highly correlated with AU1, AU5, AU25 and AU27. Therefore when training for RNN, the classifier sees the presence of those additional AUs as an indication that AU2 is present. Thus, the true positive samples almost always have the combination of 1+2+5+25+27. The false positive results we see often are the samples that have the combination of AU1, AU5, AU25 and AU27 without the presence of AU2. Additionally, the false negative samples often do not contain AU25. This implies that AU25 has a large influence on the recognition of AU2 since our sample set often contains a combination of the two AUs. The classifier associates AU2 with AU25 even though they

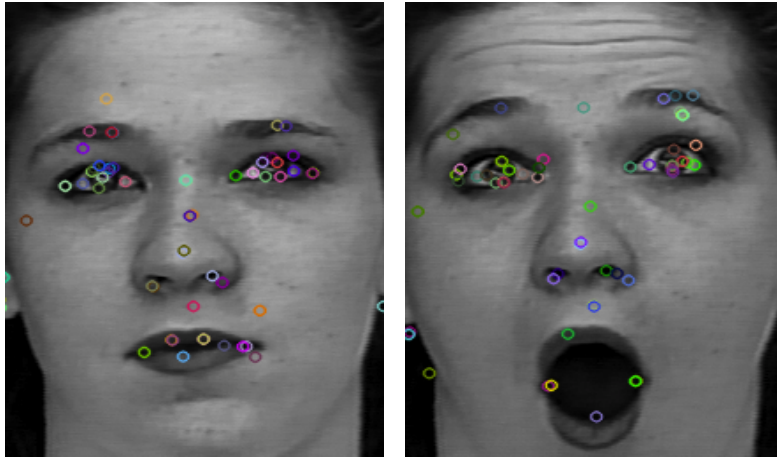


Figure 7.4: Neutral face (left) and face activating 1+2+5+25+27 (right).

are not related, do not influence each other, and are on different parts of the face.

7.4.3 Results for AU4

AU4 is the brow lowerer. From Table 7.3 we see that AU4 has a low recognition rate of 65.29%. This is due to a low true positive rate of 63.01% and a high false positive rate of 32.43%. Table B3 shows detailed results at certain threshold settings. From the results in Figure 7.6, we see that there is a sample set which performed particularly well (series 2). This sample set sees a false positive rate of 14.29% and a true positive rate of 78.57%. There is also a sample set, series 4, which sees a poor false positive rate of 47.06%.

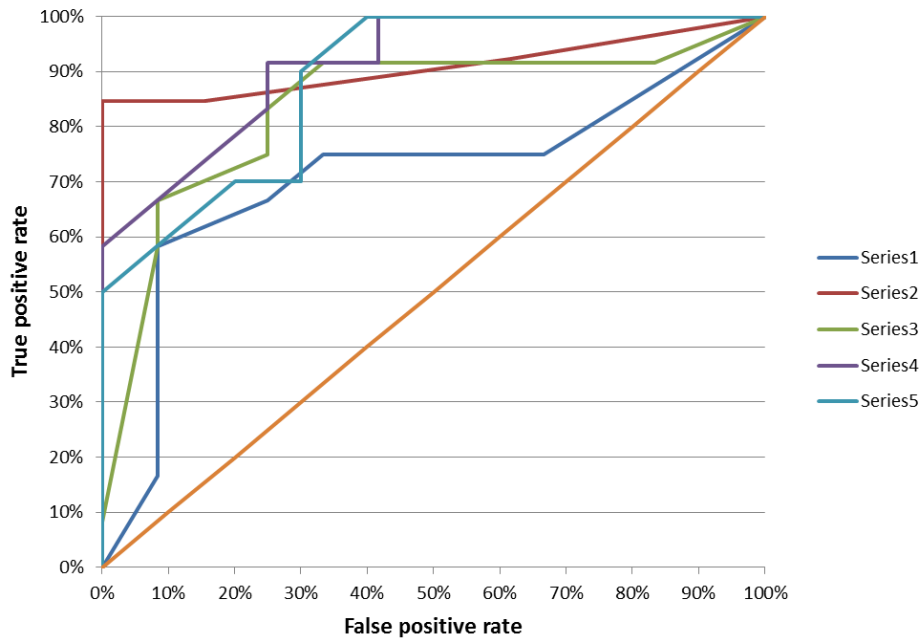
The combination of 4+5 combines the changes made by AU4 and AU5 (upper lid raiser), modifying the appearance changes due to AU4 alone or AU5 alone. The modified changes involve the appearance of the eye aperture, the upper eyelid and the amount of sclera exposed. The eye cover fold is pushed downwards by AU4 and may narrow the eye aperture, while AU5 widens the eye aperture. The resulting eye aperture from these opposing actions is a compromise between the two actions. Another appearance changing combination is 1+4. The combination of these two AUs maintains the raising action from AU1 with the drawing together action of AU4. The lowering effect of AU4 is counteracted or restricted to the outer portions of the brow. The combination of 1+2+4 also causes an appearance change. This combination of AUs pulls the eyebrows upwards and together, but neither of these changes is as significant as when found separately with AU4 alone or as with the 1+2 combination. The changes in appearance from the combination of 1+2+4 are not just the addition of those for the separate AUs, but are a different product of their joint action. This combination pulls the eyebrows together, but not as close together as when AU4 is activated alone. Many of the samples classified as false negative contain these combinations. Since these combinations cause an appearance change different to the appearance of AU4 alone or with other AUs, the classifier does not recognize these combinations with AU4 as positive.

Many of the false positive samples contain AU6, AU12 and AU17. From Table 7.1 we see that the AU4 sample set contains high instances of AU17 with 63%. Since we know that the training sample set contains many instances of 4+17, the classifier still sometimes classifies AU17 on its own as a positive instance of AU4. Additionally, there are many instances in the false positive samples of 6+12. The combination of 6+12 causes changes around the eye that could be misclassified as AU4.

7.4.4 Results for AU5

AU5 is the upper lid raiser which widens the eye aperture. Table B4 shows detailed results at certain threshold settings. From Table 7.3, AU5 has a true positive rate of 87.8%, a false positive rate of 21.95%, and a recognition rate of 82.93%. AU5 sees the highest true positive rate of all tested AUs. The false

AU2 results



AU2 average

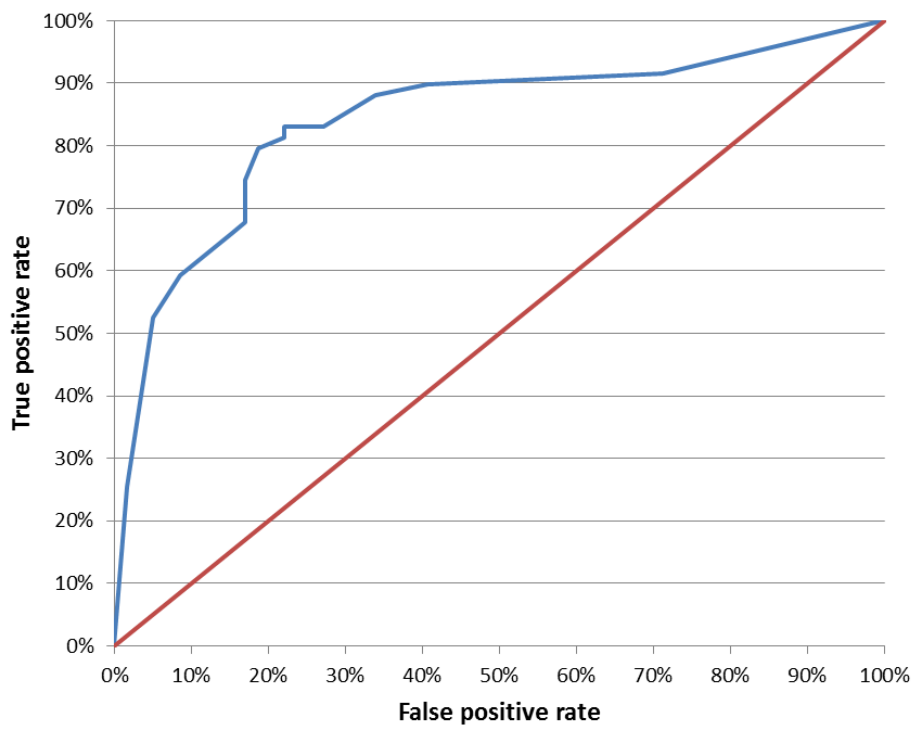


Figure 7.5: Results of AU2 testing.

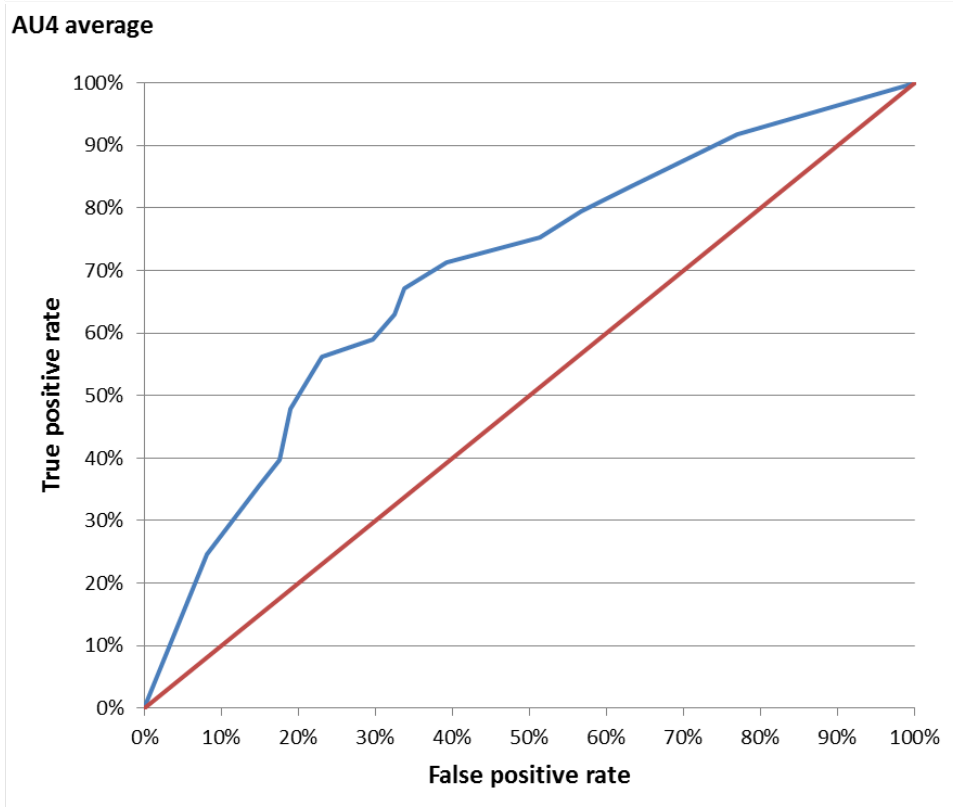
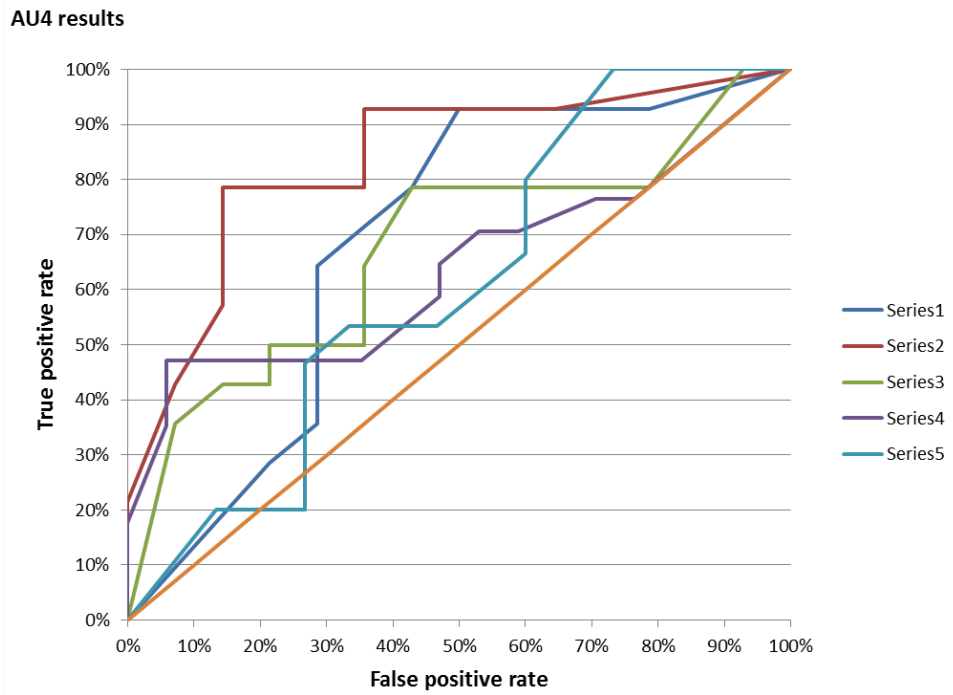


Figure 7.6: Results of AU4 testing.

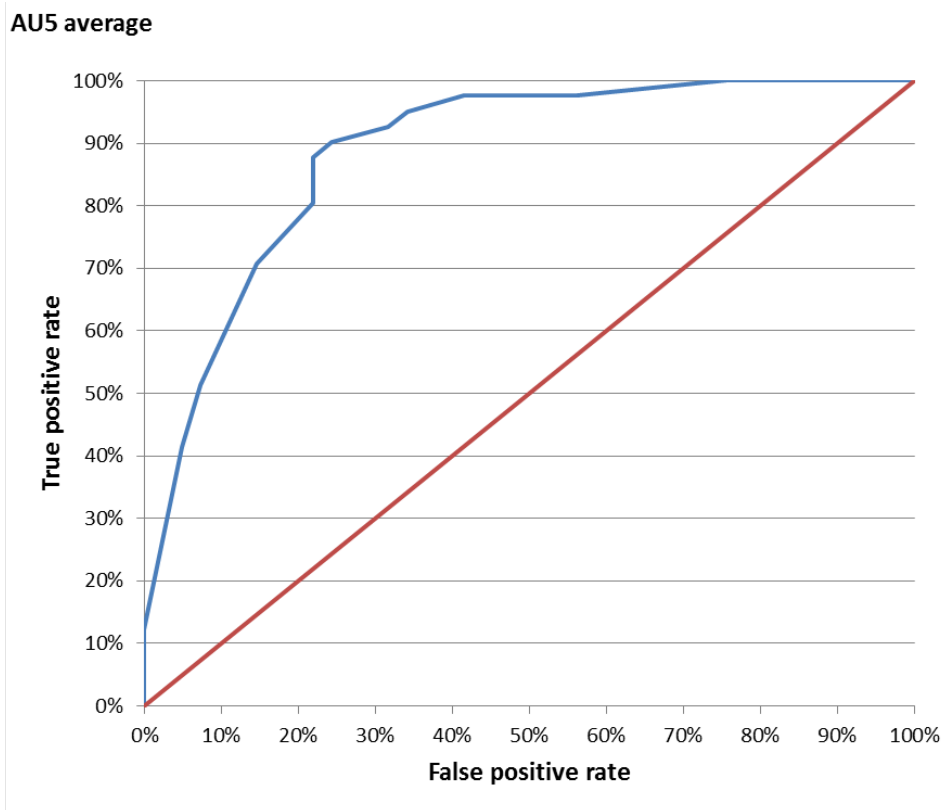
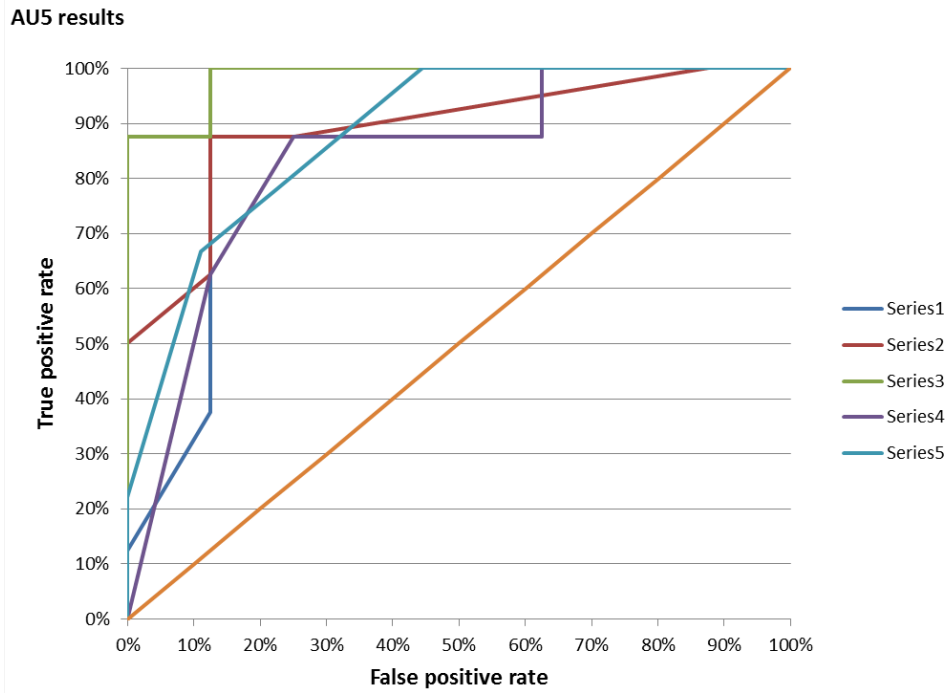


Figure 7.7: Results of AU5 testing.

positive rate of 21.95% lowers the overall recognition rate. Of the five tests performed as shown in Figure 7.7, three tests have very low false positive rates of 12.5% (series 1), 12.5% (series 2) and 0% (series 3). There are two poor performing false positive samples with 37.5% (series 4) and 44.44% (series 5).

Many false positive incidents contain AU4. We know that AU4 is also from the eye area (brow lowerer), and the appearance change caused by AU4 generally results in similar keypoints extracted from the appearance change caused by AU5.

There are very few false negatives, but all samples incorrectly classified as negative have the combination 1+2+5+25+27. The combination of 1+2+5 causes an appearance change which raises both the inner and outer parts of the brow and the upper eyelid, and causes the eyebrow to have a curved and arched shape. This appearance change could be the reason for the classifier being unable to classify AU5 when in combination with AU1 and AU2.

7.4.5 Results for AU6

AU6 is the cheek raiser and lid compressor. AU6 sees the lowest false positive rate of all AUs tested with 15.09%, a true positive rate of 79.25%, and an overall recognition rate of 82.08%. Table B5 shows detailed results at certain threshold settings. From the five sample sets tested as seen in Figure 7.8, four of the samples see very low false positive rates of 0% (series 1), 10% (series 3), 9.09% (series 4), and 18.18% (series 5). There is one sample set tested which has a high false positive rate of 40% (series 2), which increases the average false positive rate for AU6.

When looking at the samples which have a false positive result, we see that all of these samples contain AU17. Only 14% of AU6 samples contain AU17, therefore it is not a case that the classifier is associating AU6 with AU17. Thus the few instances of false positives cannot be attributed to a correlation with another AU.

All of the false negative samples contain the combination 6+12 and 6+7. AU12 is the lip corner puller. The combination of AU6 and AU12 causes an appearance change of the corners of the lips to pull back and upward, resulting in a smile shape. Additionally, the combination of 6+7 also causes an appearance change. AU7 is the lid tightener, and the combination of AU6 and AU7 causes changes in the upper eye lid depending on the shapes and position of the eye socket. Due to these two combinations which cause changes in the appearance of AU6 on the subject, some instances of 6+12 and 6+7 are not recognized.

7.4.6 Results for AU7

AU7 is the lid tightener. From Table 7.3 we see that AU7 has a low true positive rate of 68.75%, a false positive rate of 25%, and a recognition rate of 71.88%. Table B6 shows detailed results at certain threshold settings. From Table 7.1, 85% of AU7 samples contain AU4. Some samples perform well (curve closer to the top left corner of the plane) as seen in Figure 7.9. However, there are two particularly poor performing sample sets which affect the recognition rate, which are series 1 and series 5. These both cross the 45-degree line. Four of the five tested sample sets see low false positive rates between 10% and 22%, however one set has a high false positive rate of 55.56% (series 4). Upon inspection of the poor performing sample set, as well as the false positives from the other four sample sets, we see that the false positive samples contain AU25.

Another poor performing sample set sees a true positive rate of only 50% (series 1), while the other sample sets see true positive rates between 70% and 80%. All the samples that have false negative results have the combinations of 4+7+9, or 4+6. Although these are not combinations that cause appearance changes, these AUs do have an effect on each other. AU6 causes crows feet around the eyes, while AU7 causes crows toes (shorter wrinkles). Thus the presence of AU6 can hide the presence of AU7. AU9 is the nose wrinkler. AU9 can be strong enough to narrow the eye aperture. AU9 can therefore obscure the presence of AU7, and unless the actions of AU7 and AU9 are sequential in a motion record, it is difficult to see the signs of AU7, especially if AU9 is strong. AU4 is the brow lowerer, which in lowering the brow may also narrow the eye aperture. If AU4 is present, the lower lid must also be raised in order

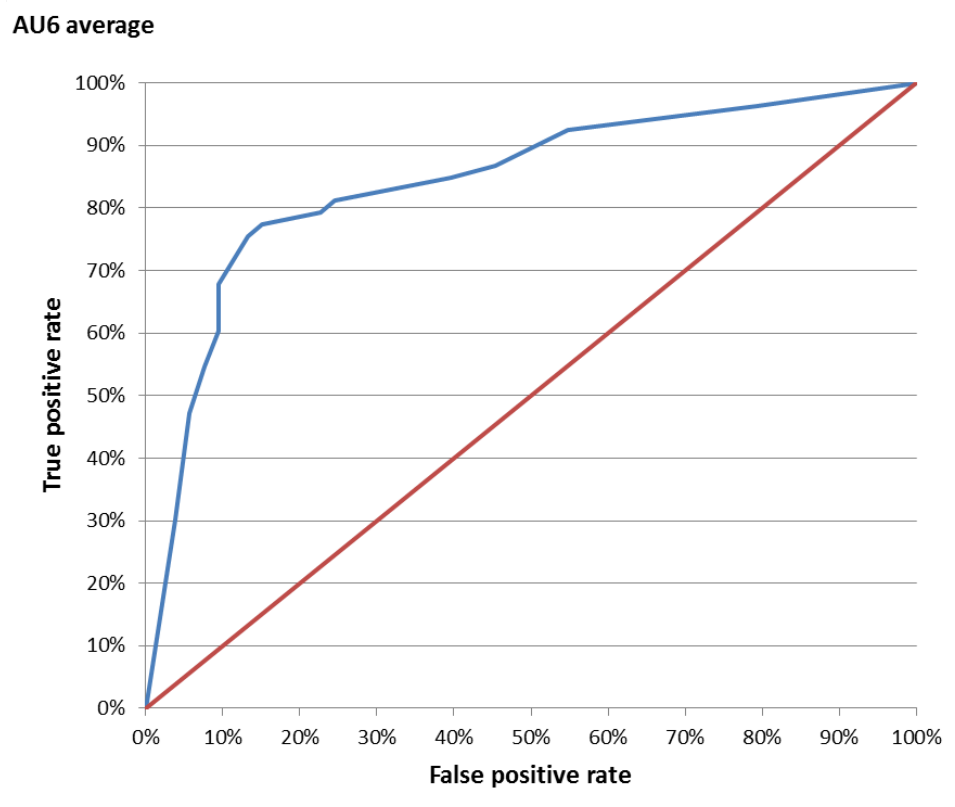
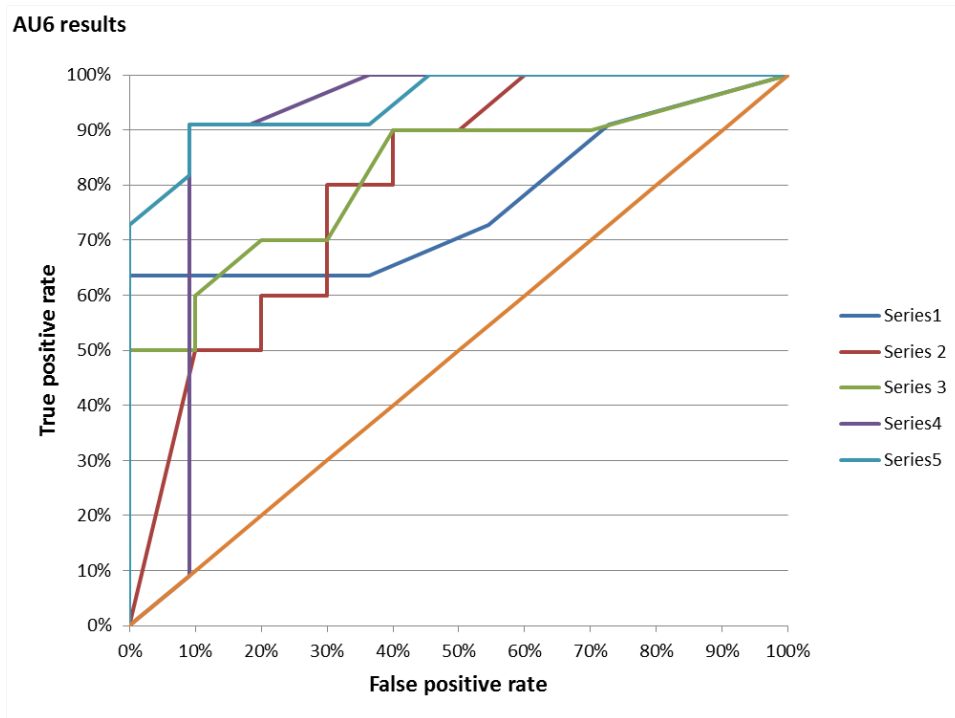


Figure 7.8: Results of AU6 testing.

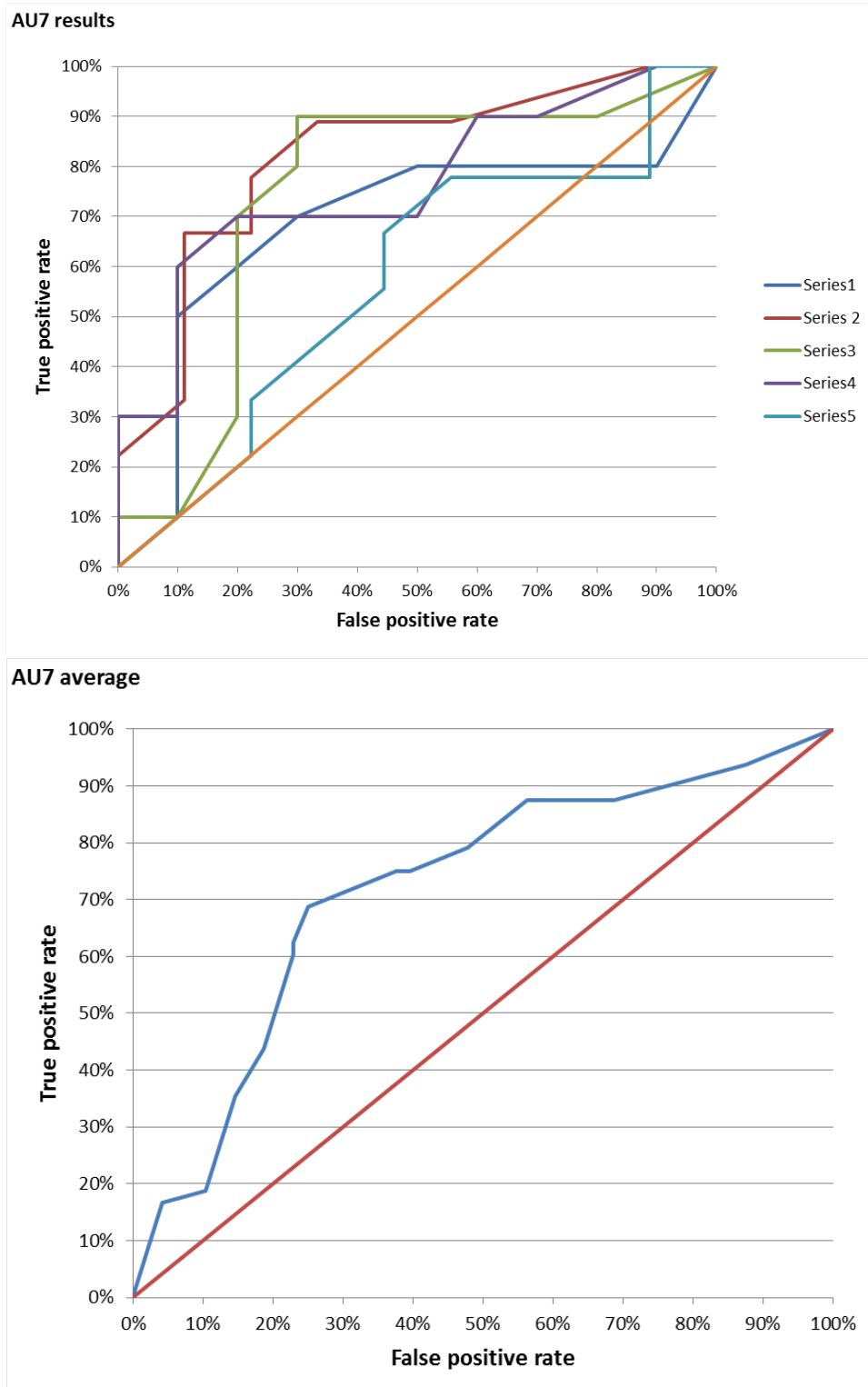


Figure 7.9: Results of AU7 testing.

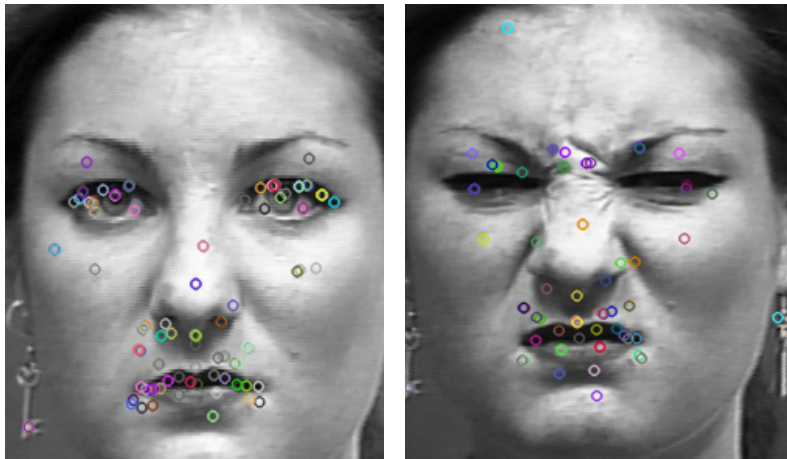


Figure 7.10: Neutral face (left) and face activating 4+7+9+17 (right).

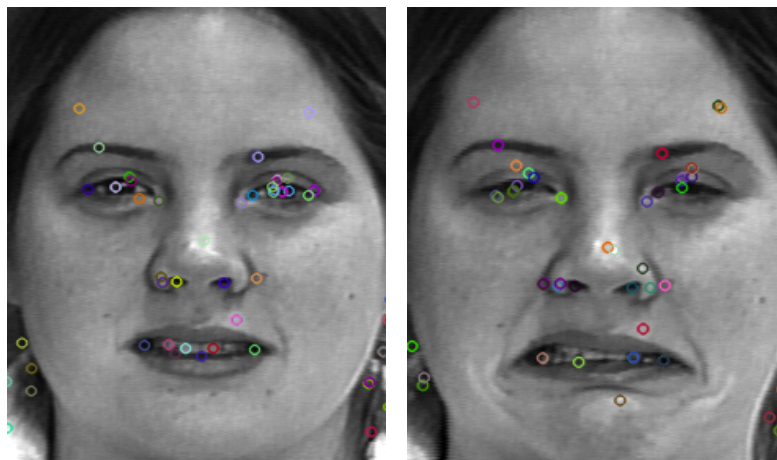


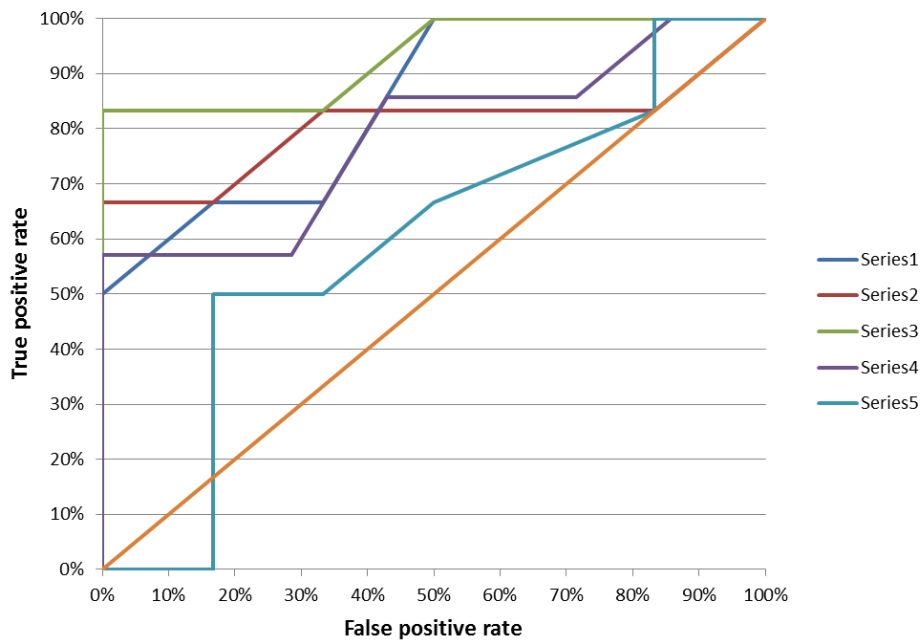
Figure 7.11: Neutral face (left) and face activating 7+15+17+20+25+43 (right).

to recognize AU7. These combinations are the reason that some samples of AU7 are classified as false negatives. Figure 7.10 and Figure 7.11 are examples of this effect. In Figure 7.10, this sample was incorrectly classified as negative. The nose wrinkler effect of AU9 and the brow lowerer effect of AU4 did not allow AU7 to be recognized. Figure 7.11 is a correctly classified positive sample, where AU4, AU6, and AU9 do not occur with AU7. The appearance of AU7 in these two samples is significantly different. As a result, AU7 in combination with AU4, AU6 and AU9 is sometimes not classified correctly.

7.4.7 Results for AU9

AU9 is the nose wrinkler. Although it takes place on the middle face, it is considered by FACS to be a lower face AU. AU9 sees a high true positive rate with 74.19%. However, it also sees a high false positive rate of 29.03%. The results for AU9 are shown in Figure 7.12. Table B7 shows detailed results at certain threshold settings. From Table 7.1 we see that the AU9 sample set also has high instances of AU4 (88% of samples), AU7 (86% of samples) and AU17 (76% of samples). Additionally, there are several more samples containing those AUs than AU9. This means that there are samples containing those AUs which do not contain AU9. These samples form part of the AU9 sample set as negative samples (AU not present). When training the RNN, the classifier receives many samples of AU9 occurring together with AU4, AU7 and AU17. It also receives the negative samples containing AU4, AU7 or AU17 without AU9. Hence, when classifying, the RNN is misclassifying the negative samples which contain AU4, AU7 and AU9 as a positive instead of a negative.

AU9 results



AU9 average

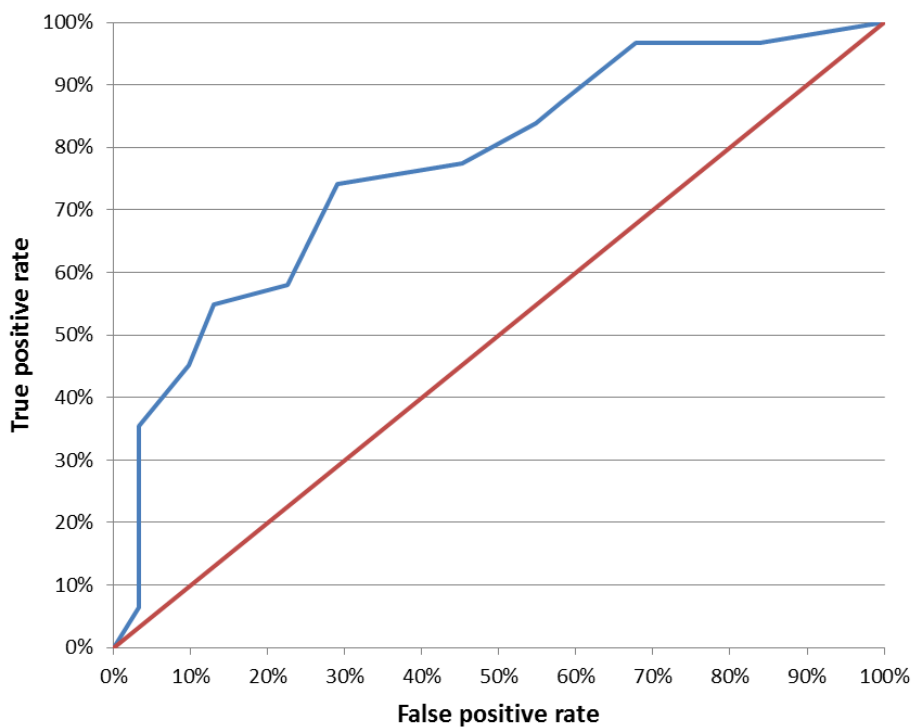
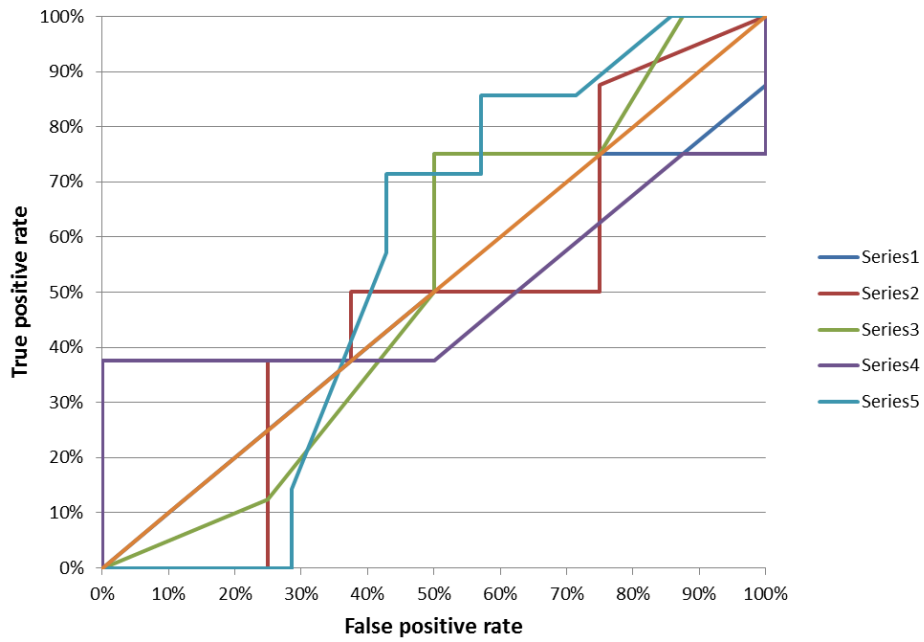


Figure 7.12: Results of AU9 testing.

AU15 results



AU15 average

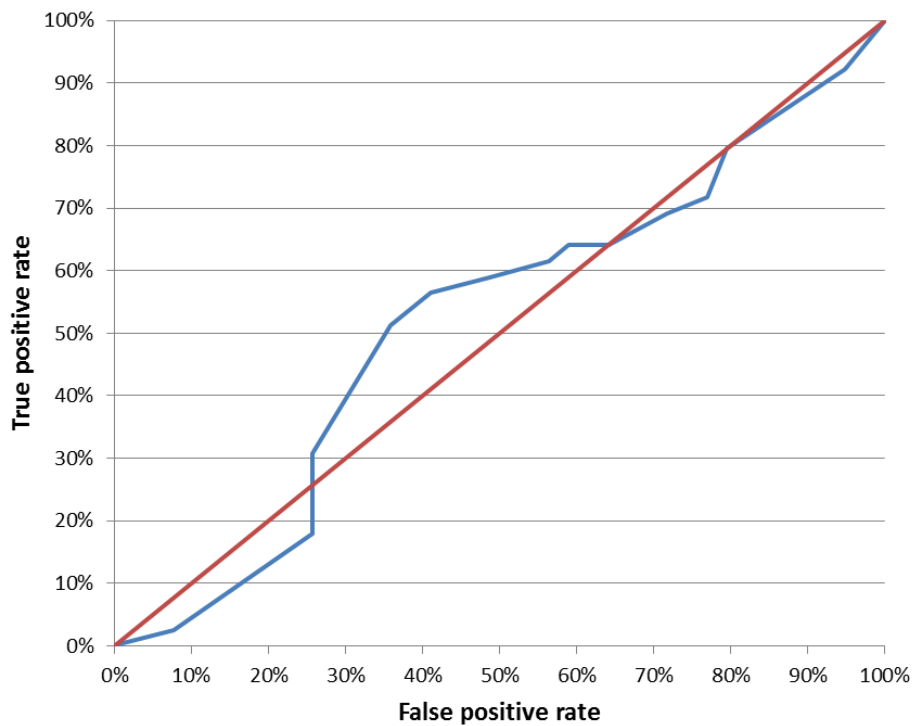


Figure 7.13: Results of AU15 testing.

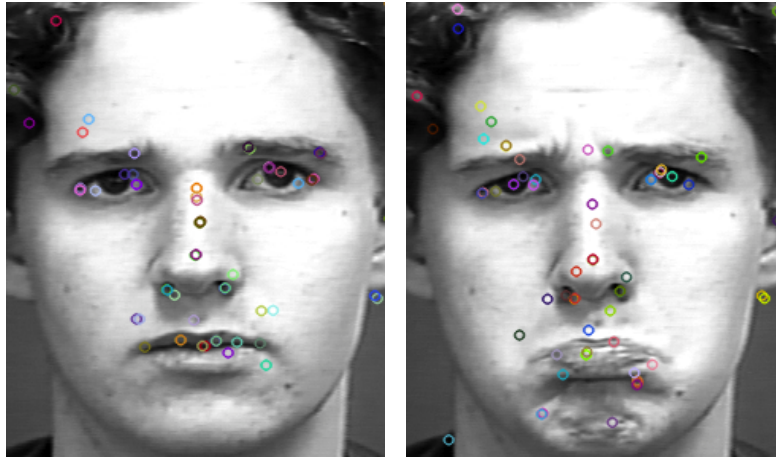


Figure 7.14: Neutral face (left) and face activating 1+4+7+15+17 (right).

7.4.8 Results for AU15

AU15 is the lip corner depressor which pulls the corners of the lips down. AU15 gives some of the worst results of all AUs tested. It has a low true positive rate of 61.54%, an extremely high false positive rate of 48.72%, for a recognition rate of 56.41%. Table B8 shows detailed results at certain threshold settings. From Table 7.1, we see that AU15 almost always occurs in conjunction with AU17, which is the chin raiser. Of the five tests performed as seen in Figure 7.13, three see high true positive rates of over 70% (series 1, series 3, and series 5). However, two samples have very low true positive rates of 50% and 37.5% (series 2 and series 4). Upon inspection of these samples, there are a few samples with the combination of 15+23. The combination of these two AUs tends to attenuate some of the signs associated with each alone. Although this accounts for 14% of the false negatives, there is no particular pattern in the samples that can cause the rest of the false negatives, and there are also no other appearance change combinations which provide a reason for the low true positive rate.

Additionally, AU15 does not share a particularly strong correlation with another AU except for AU17, which is almost always in conjunction with it. Half of the false positive samples have the combination of 1+2+25+27, and almost all contain AU25. AU25 is when the lips part. Since AU15 is also a mouth movement, the classifier could be misclassifying AU25 as AU15 due to the changes in the appearance of the mouth.

Figure 7.14 is a correctly classified sample containing AU15. Figure 7.15 is an incorrectly classified sample containing AU15. The differences lie mainly in the number of keypoints extracted below the bottom lip around the chin area. The correctly classified sample has more keypoints extracted around the chin area due to more furrows appearing and accumulating on the chin area.

7.4.9 Results for AU17

AU17 is the chin raiser. AU17 sees a very low true positive rate of 58.42%, and a high false positive rate of 30.44%, for a recognition rate of 63.99%. Results for AU17 are shown in Figure 7.16. Table B9 shows detailed results at certain threshold settings. There are approximately twice as many samples of AU17 as there are of AU15. All AU15 samples also contain AU17. Thus around 50% of AU17 samples have the combination 15+17, while the other 50% of AU17 samples contain other combinations.

AU10 is the upper lip raiser. The appearance changes caused by AU10 are apparent in the combinations of 10+15, 10+17, and 10+15+17 but absent in 15+17 or 6+15+17. The appearance changes caused by AU15 are apparent in 10+15, 10+15+17, 15+17, and 6+15+17 but are absent in 10+17. The appearance changes caused by AU17 are apparent in 10+17, 10+15+17, 15+17, and 6+15+17 but are absent in 10+15. In all of these combinations 10+15, 10+17, 15+17 and 10+15+17 the corners of the mouth angle down. This angle is due to the effect of appearance change from AU10, appearance changes from AU15,

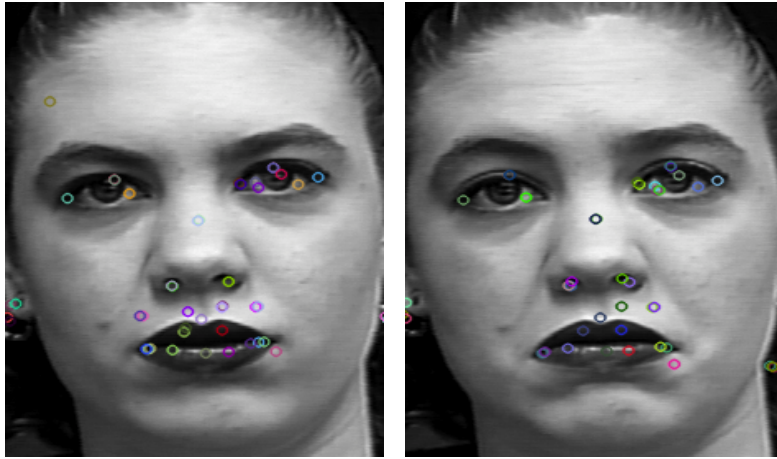


Figure 7.15: Neutral face (left) and face activating 1+2+15+17 (right).

or appearance changes from AU17. Another appearance change is 14+17. AU14 is the dimpler. AU17 becomes less evident in the combination of 14+17, and the shape of the lips is different from what is produced by either AU alone. The combination of 12+17 causes the shape of the lips to be changed. Due to the changing nature of combinations of AU17, the classifier struggles to correctly classify samples containing AU17. The keypoints extracted are different across samples depending on the combination and the appearance change caused by the combination.

7.4.10 Results for AU20

AU20 is the lip stretcher, which pulls the lips back laterally, and elongates the mouth. AU20 has a low true positive rate of 56%, a high false positive rate of 44%, for a low recognition rate of 56%. Table B10 shows detailed results at certain threshold settings. From Figure 7.17, we see that the average results for AU20 lie on the 45-degree line, indicating an inaccurate test.

AU27 is the mouth stretch. The action of elongating the mouth is less obvious in 20+27 because the shape of the mouth is stretched in the vertical direction, although the lips are still elongated horizontally. AU26 is the jaw drop. When combining AU26 and AU27, the jaw is pulled wide open, usually rapidly and beyond the excursion limit of AU26, and space between teeth should be apparent in combination with AU20. AU10 is the upper lip raiser and AU25 is the lip parter. The combination of 10+20+25 causes an angular bend in the shape of the upper lip. The angle is not as sharp as in AU10 alone due to lateral pull by AU20. Additionally, AU10 deepens the nasolabial furrow and raises the upper part of this furrow, and AU20 stretches the lower portion laterally, producing a somewhat laterally stretched version of the characteristic AU10 shape. AU23 is the lip tightener. The combination of 20+23+25 causes the lips appear stretched horizontally (by AU20), but also narrowed and tightened (by AU23), and may cause small wrinkles in the skin above and below the lips and muscle bulges below the lower lip, but AU20 tends to cancel out this appearance change due to AU23. We see many incidents of these combinations in the false negatives. Due to the appearance changes, the classifier does not correctly classify AU20.

Additionally, almost all AU20 samples occur simultaneously with AU25 as seen in Table 7.1. There are significantly more AU25 samples than AU20, which means that the negative samples in the AU20 sample set often contain AU25. The classifier could be misclassifying AU20 as AU25. AU20 also has a high correlation with AU4 and AU1, which also have significantly more samples than AU20. Due to this, the classifier is not receiving enough information from AU20 alone to correctly classify it, and is associating keypoints from other parts of the face as keypoints for AU20. AU25 is around the mouth area, but AU1 and AU4 are from the eye area. Since there is a smaller sample size of AU20, the changes caused by the combinations above make it challenging for the classifier to isolate the changes caused by AU20.

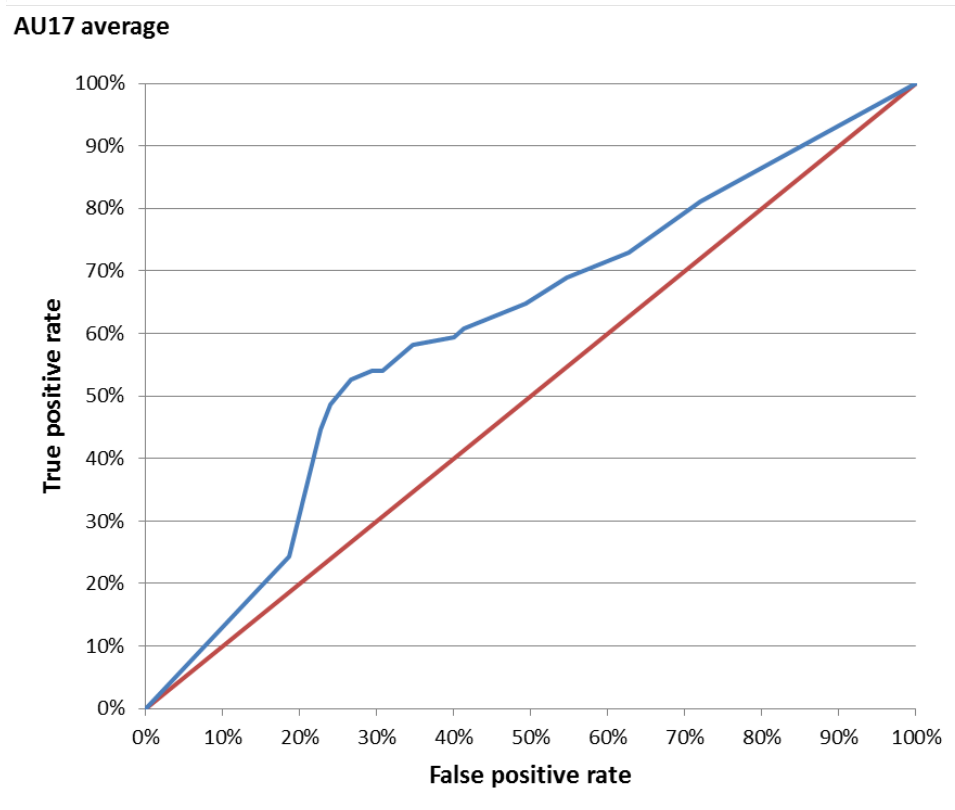
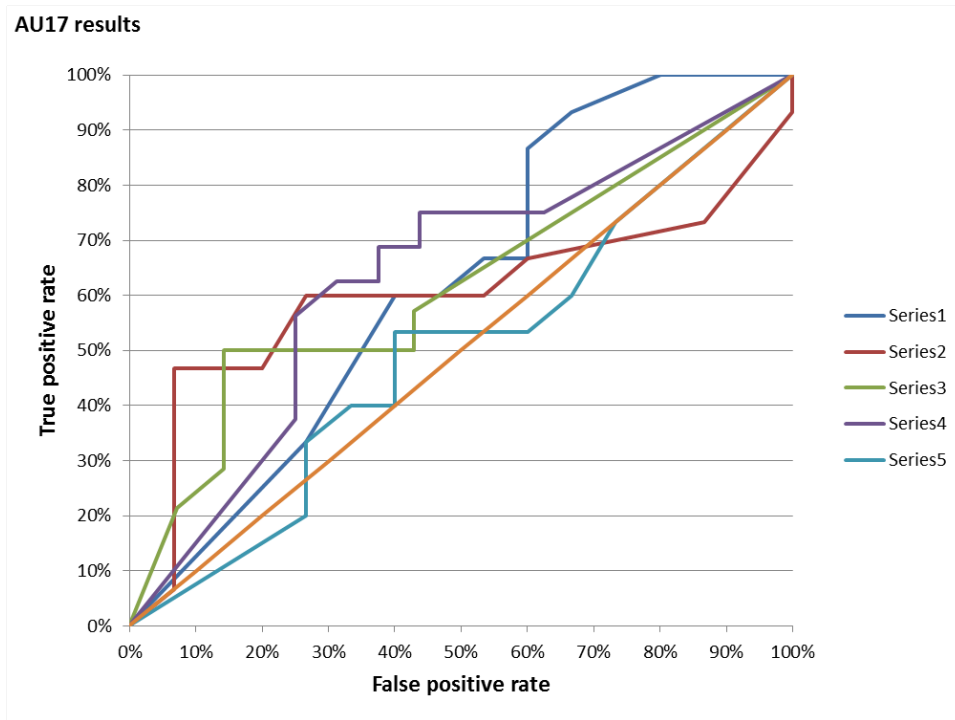
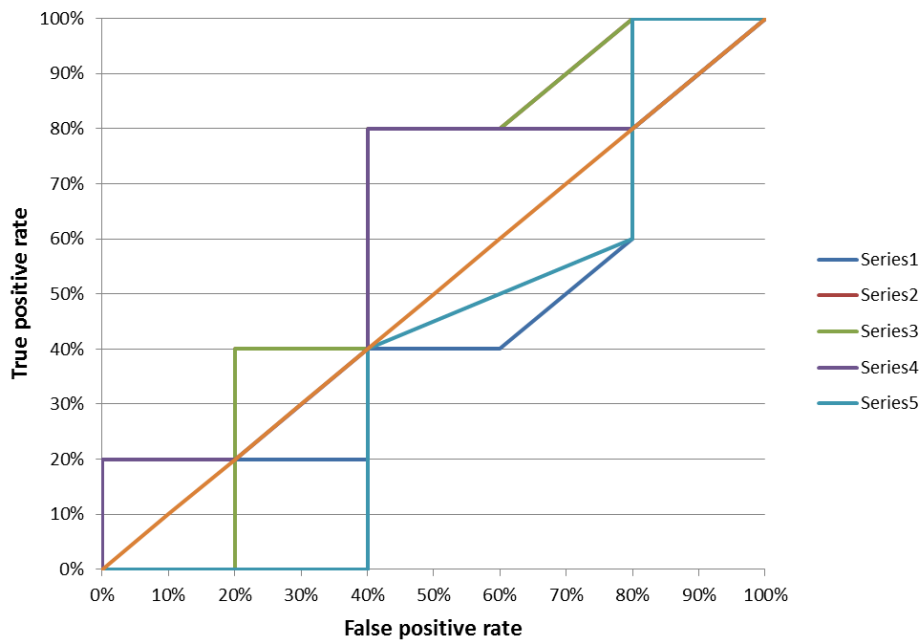


Figure 7.16: Results of AU17 testing.

AU20 results



AU20 average

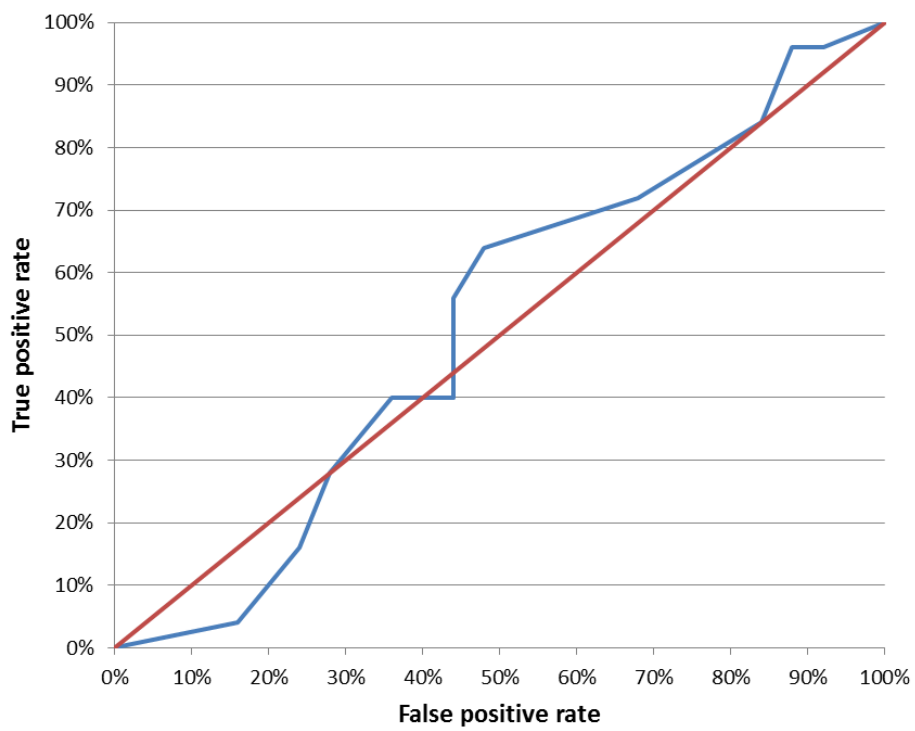


Figure 7.17: Results of AU20 testing.

7.4.11 Results for AU25

AU25 is the lips part. From Table 7.3, AU25 has a low true positive rate of 68.27%, an extremely high false positive rate of 45.19%, for an overall recognition rate of 61.54%. Results of testing are shown in Figure 7.18. Table B11 shows detailed results at certain threshold settings.

This high false positive rate implies that only approximately half of the negative samples are correctly classified. AU25 does not have a high correlation with any other AUs, which could often help explain the high false positive rate. Upon inspection of the false positives, we see that image sequences which have some degree of mouth movement are being considered positive instead of negative. This includes AU15. This implies that mouth movement of any form is considered AU25 by the classifier. We also see that the shape of the mouth and size and thickness of the lips differs significantly between subjects. This means that the parting of the lips looks different across different subjects and the degree of lip parting differs across samples, making it difficult for the classifier to adequately recognize AU25. In the case of AU27, which is the complete stretching of the mouth, the size of the mouth and lip width becomes less relevant than in the case of AU25, making AU27 easier for the classifier to recognize.

In the case of the false negatives, we see that many combinations of AU25 cause appearance changes. AU16 pulls the lower lip down. Sometimes this movement is enough to part the lips, which is the combination of 16+25. However, in these cases the lower teeth are exposed, causing a different appearance to that of AU25 alone. Other appearance change combinations include 10+23+25, 23+25+26, and 22+23+25.

7.4.12 Results for AU27

AU27 is the mouth stretch. It has a true positive rate of 86.27%, a false positive rate of 15.69%, for the highest recognition rate of 85.29%. Table B12 shows detailed results at certain threshold settings. This AU causes large mouth movement. The mandible is pulled down, and the mouth does not appear as if it has fallen open but rather as if it is actively pulled down forcibly or stretched open widely. From Table 7.1, every instance of AU27 occurs in conjunction with AU25. Additionally, AU27 shares a large correlation with AU1, AU2, and AU5. The results of AU27 testing are shown in Figure 7.19. The average ROC curve is close to the top left hand corner, indicating a very accurate test.

7.5 Results of Phase II Testing (UNBC McMaster Pain Database)

As discussed in Chapter 2, the combination of AUs 4, 6 or 7, 9 or 10, and 43 indicate pain. The level of the pain is determined by the PSPI score, which is the number of those AUs present at the same time, as well as the strength at which the AU is present. Thus a PSPI score of zero indicates no pain, one is trace pain, two is weak pain, and a PSPI score of three or greater is considered strong pain. We have trained our RNNs to determine the presence of AU4, AU6, AU7, and AU9. Either as a positive presence (AU is active), or negative presence (AU is inactive), without considering the strength of the AU.

The UNBC McMaster database contains image sequences containing the onset of pain. The database contains both a PSPI score and the FACS AUs present for each frame within an image sequence. We collect samples where no pain was present, i.e. no active AUs. We also collect samples of subjects exhibiting strong pain, i.e. a PSPI score of three or greater. For these samples, we look at the AUs present to give the PSPI score for three or more, as well as the intensities at which they are present. We then take these samples and create sample sets for AU4, AU6, AU7, and AU9. These sample sets contain samples of each AU at differing intensities (i.e. from intensity A to E).

The UNBC-McMaster database contains image sequences where the face moves in-plane or out-of-plane. Subjects in this database are not facing front-on as in the Cohn Kanade database. Thus, each sample contains frames where the subject's head moves, and we collect the AUs present in the last image of the image sequence. For example, if AU4 and AU7 are present in the final frame, we add the sample to the sample set for both AU4 and AU7, irrespective of the intensity at which the AU is present. As discussed in Section 6.9, we need to change the image size to the same as that of Cohn Kanade, as well

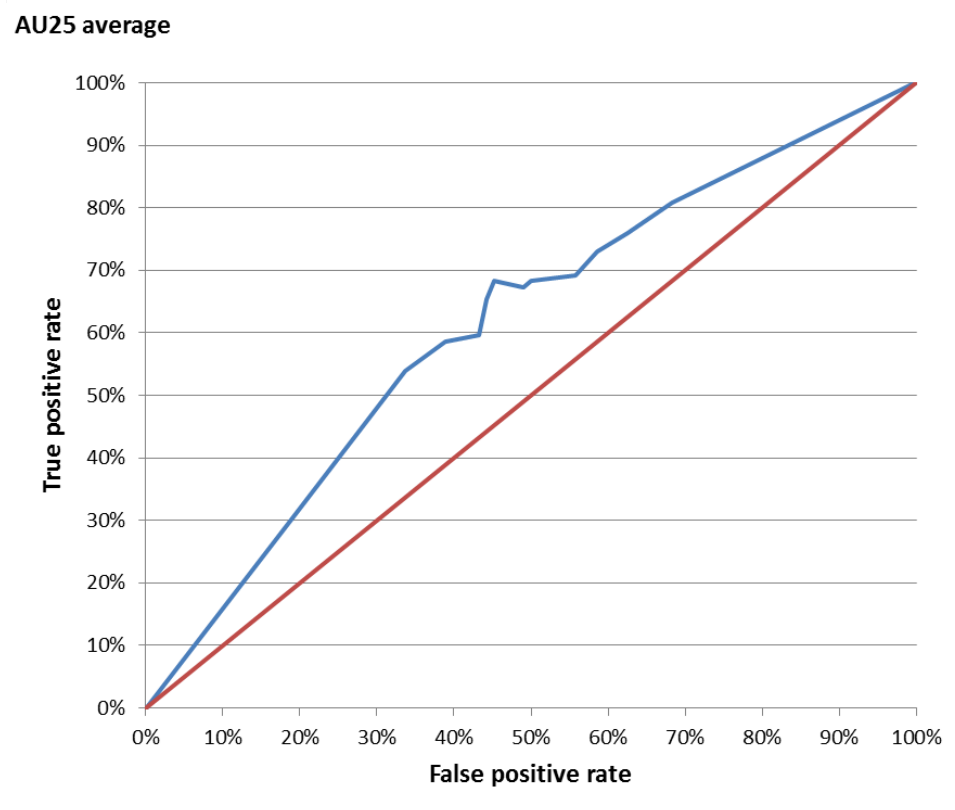
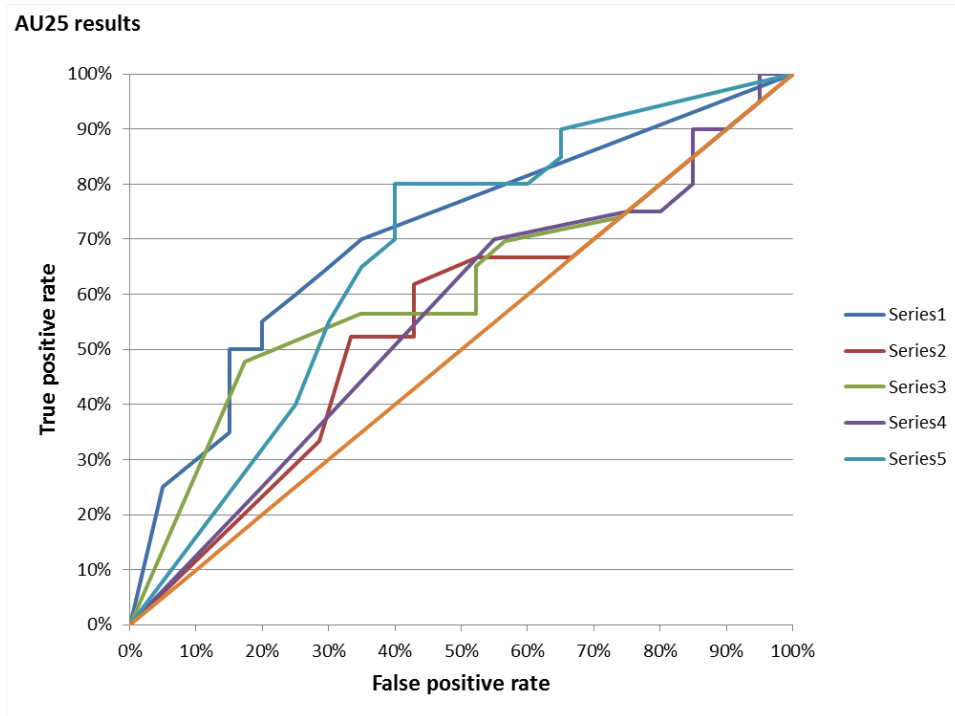


Figure 7.18: Results of AU25 testing.

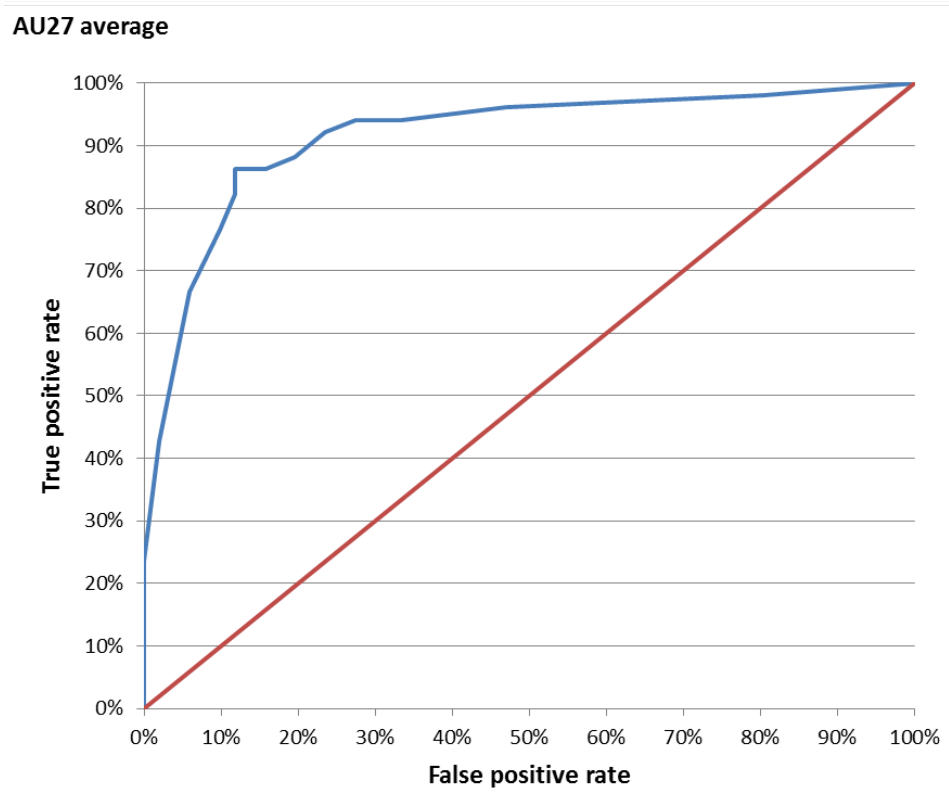
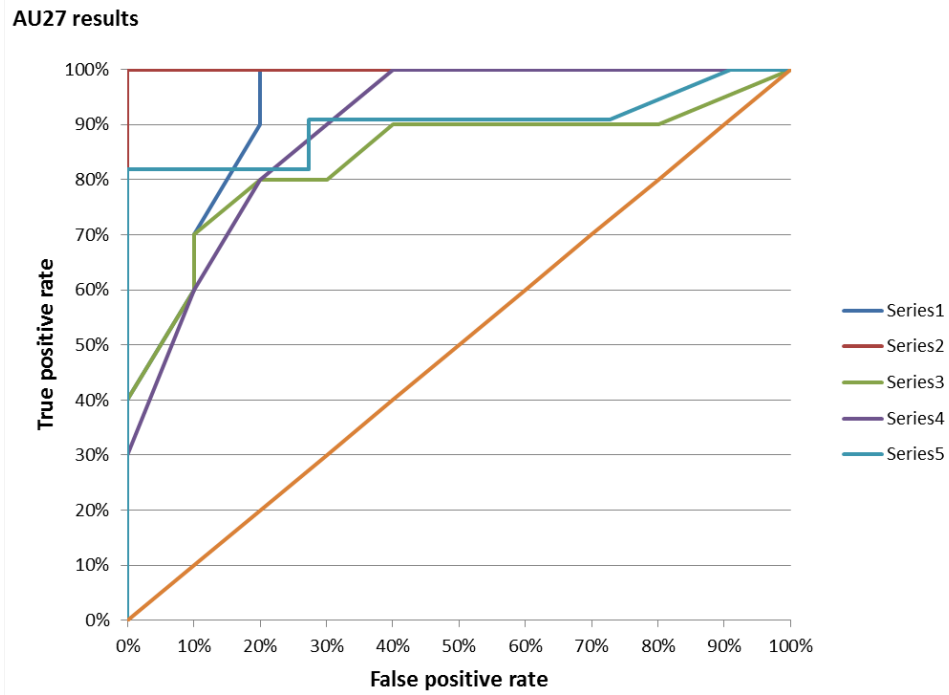


Figure 7.19: Results of AU27 testing.

	AU4	AU6	AU7	AU9
AU4	1.00	0.24	0.41	0.43
AU6	1.00	1.00	0.73	0.93
AU7	0.75	0.32	1.00	0.36
AU9	0.5	0.26	0.23	1.00

Table 7.4: Correlation between AUs in the UNBC-McMaster database.

	Total number of samples	Positive samples	Negative samples
AU4	61	12	49
AU6	174	50	124
AU7	95	22	73
AU9	84	14	70

Table 7.5: UNBC-McMaster sample set.

as convert the image to greyscale for each image in the image sequence. We also have to shorten the image sequences as the sequences in the UNBC-McMaster database are significantly longer than the sequences in the Cohn Kanade database. We use our trained RNNs to classify sample sets from the UNBC-McMaster database. That is, the RNN is trained with samples from the Cohn Kanade database, and tested with samples from the UNBC-McMaster database.

We show the correlation between AUs in the UNBC-McMaster database in Table 7.4. This table shows the percentage of samples where AUs occur together. Correlations greater than 0.5 are shown in bold. This is with the exception of the diagonal. From Table 7.4, we see that 75% of AU4 samples contain AU7, and 41% of AU7 samples contain AU4. Since the RNN for each AU is trained using samples from Cohn Kanade, we need to take into account the correlation between AUs in the Cohn Kanade database as well as the UNBC-McMaster database when analyzing results.

Table 7.5 shows the sample set size for each AU. Unlike with the Cohn Kanade database where there are many samples, the UNBC-McMaster database contains less samples with a positive depiction (i.e. AU is present). Thus, the sample sets are not 50% positive and 50% negative as in our sample sets for the Cohn-Kanade database. The sample sets are on average 22% positive, and on average 78% negative.

Additionally, from the positive samples for each AU as shown in Table 7.5, we show the number of samples of each intensity for each AU in Table 7.6. For example, from the 12 positive samples in the AU4 sample set, zero are 4A, three are 4B, two are 4C, four are 4D, and three are 4E. As discussed in Section 2.3, E is the highest intensity score and indicates the AU at its maximum presence.

The results of testing are shown in Table 7.7. This is with a threshold of 0.5. This means for an output less than 0.5, we consider the output as negative. For an output of 0.5 and greater, we consider the output as positive. The ROC curves for each AU are shown in Figure 7.20. Detailed results at various threshold settings are shown in Appendix B. Table 7.8 shows the percentage of each intensity correctly identified. For example, 71.43% of 7A samples are correctly identified. Table 7.9 shows the positive output means (i.e. the average). For the RNN, a zero output indicates the absence of an AU, and an output of one indicates the presence of the AU. Thus, we look at all of the outputs of the RNN of the positive samples, and find the mean of the outputs to determine approximately what output the RNN gives for different intensities. For example, for all samples of 6A, the mean of the outputs is 0.6444. Table 7.10 shows the output means of only the correctly classified positive samples. From Table 7.8, we see that 71.43% of 6A samples are correctly classified. The mean of the outputs of the correctly classified samples is 0.816. This can help us determine if the intensity of AUs determines the output and if it effects the recognition rate.

Results of Phase II testing.

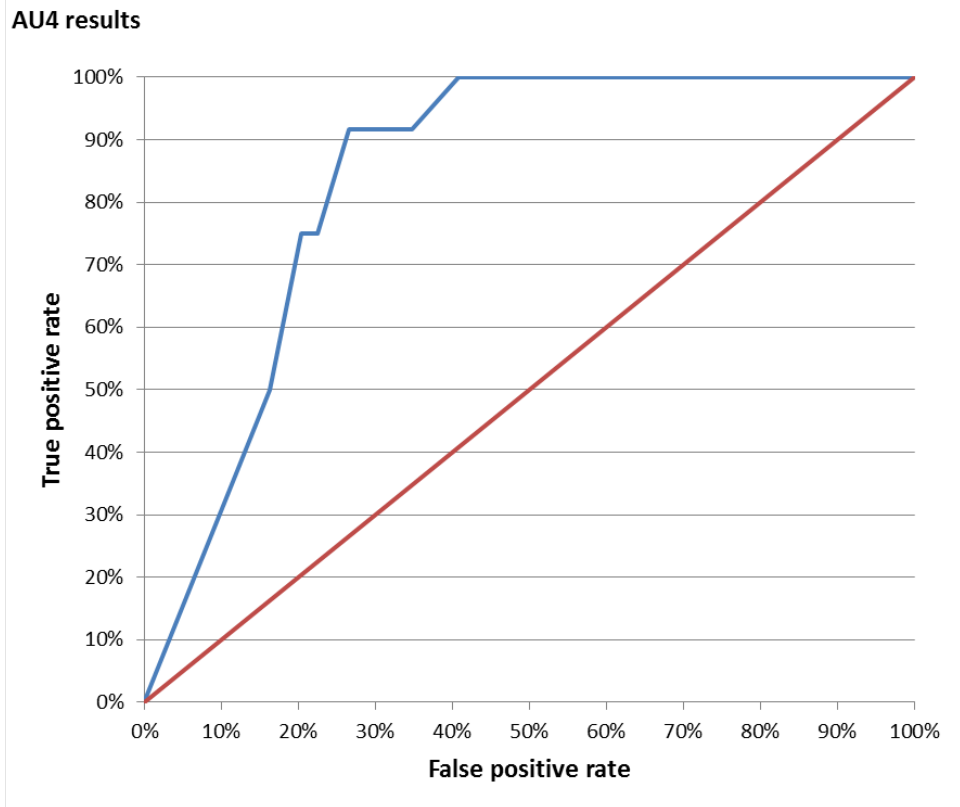


Figure 7.20: Results of AU4 testing.

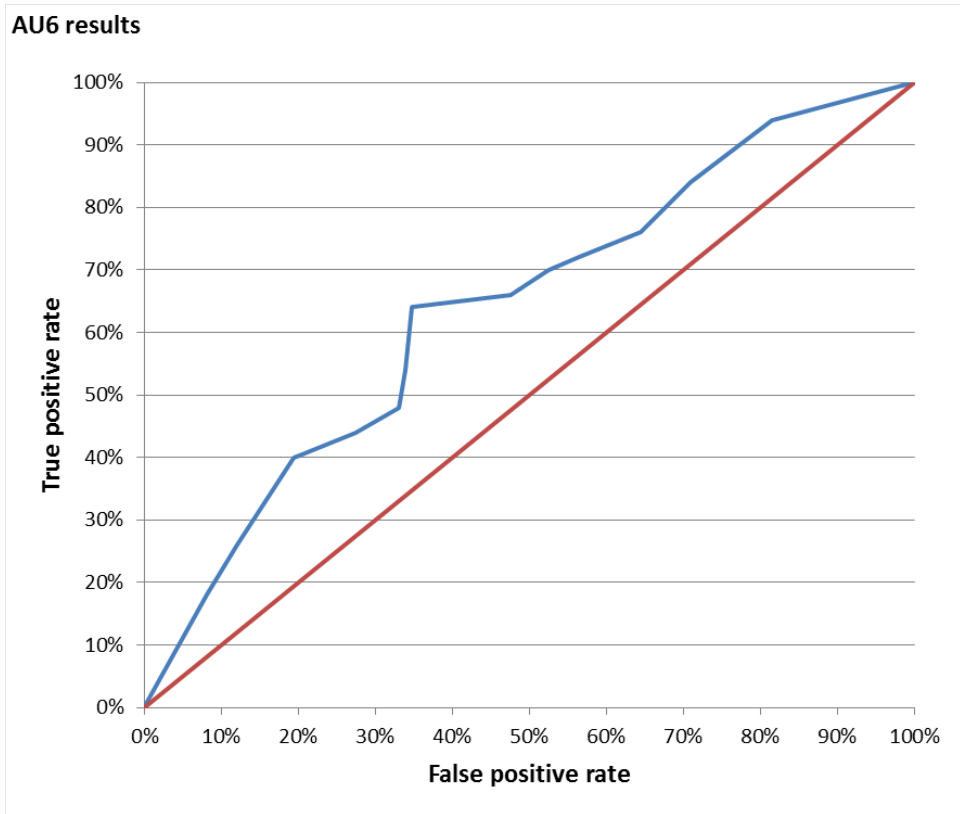


Figure 7.21: Results of AU6 testing.

AU7 results

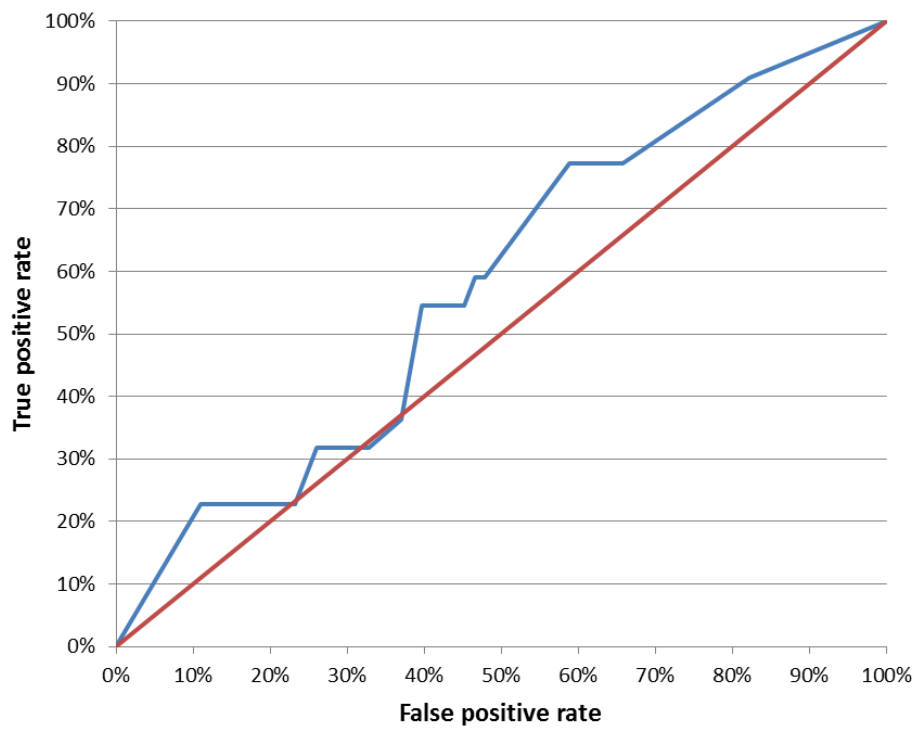


Figure 7.22: Results of AU7 testing.

AU9 results

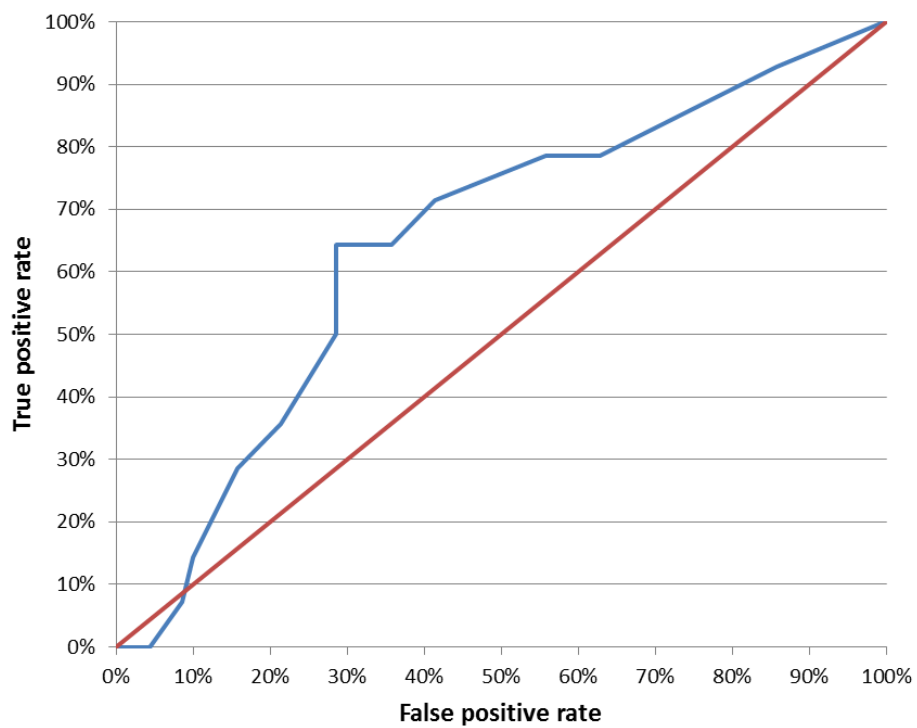


Figure 7.23: Results of AU9 testing.

	A	B	C	D	E
AU4	0	3	2	4	3
AU6	7	8	20	13	2
AU7	1	3	7	8	3
AU9	0	7	2	4	1

Table 7.6: AU intensity sample numbers.

	True positive rate	False positive rate	Recognition rate
AU4	92%	31%	74%
AU6	64%	35%	65%
AU7	55%	40%	59%
AU9	64%	29%	70%

Table 7.7: Results of Phase II Testing.

	A	B	C	D	E
AU4	-	100%	100%	75%	100%
AU6	71%	63%	45%	85%	100%
AU7	100%	67%	43%	38%	100%
AU9	-	43%	100%	75%	0%

Table 7.8: Correctly identified samples for each intensity.

	A	B	C	D	E
AU4	-	0.9963	0.987	0.812	0.889
AU6	0.6444	0.4563	0.4106	0.8165	0.998
AU7	0.54	0.3867	0.4219	0.3375	0.9999
AU9	-	0.4214	0.62	0.5625	0.12

Table 7.9: Positive output means.

	A	B	C	D	E
AU4	-	0.9963	0.987	0.9927	0.889
AU6	0.816	0.6536	0.86825	0.9266	0.998
AU7	0.54	0.56	0.8043	0.8433	0.9999
AU9	-	0.8333	0.625	0.7433	-

Table 7.10: Positive output means - correctly classified samples.

7.5.1 Results for AU4

AU4 is the brow lowerer. From Table 7.7, this sample set sees a true positive rate of 91.67%, a false positive rate of 30.61%, and a recognition rate of 73.77%. Table B13 shows detailed results at certain threshold settings. From Table 7.5, we see that the AU4 sample set has 49 negative samples and twelve positive samples. For a true positive rate of 91.67%, eleven out of the twelve positive samples are

correctly classified. From Table 7.10, we see that for intensities B to E, the outputs for correctly classified positive samples are all close to one. One is a perfectly positive output, so the closer to one, the more certain the classifier that it is a positive output. Intuitively, we would expect higher intensity positive samples to have outputs closer to one. Only one positive sample is incorrectly classified as negative, and that sample is at intensity D.

The false positive rate of 30.61% is high. Most of the misclassified negative samples do not contain any AUs (neutral faces). However, some samples are not neutral faces, and these all contain AU7 at varying intensities. From Table 7.1, we know that 58% of AU4 samples contain AU7 in the Cohn Kanade database. Since the RNN is trained using samples from the Cohn Kanade database, it could misclassify AU7 as AU4 since they often occur together.

7.5.2 Results for AU6

AU6 is the cheek raiser which results in lid compression. From Table 7.7, AU6 sees a true positive rate of 64%, a false positive rate of 34.68%, and a recognition rate of 64.94%. Table B14 shows detailed results at certain threshold settings. From Table 7.5 we see that AU6 has the largest sample set of all AUs. Out of a total of 174 samples, 50 are positive. The ROC curve for AU6 is shown in Figure 7.20.

From Table 7.6, we have 20 samples of 4C. However, from Table 7.8, only 45% of these are correctly classified. The most commonly correctly classified intensities for AU6 are D and E. These are high intensities and show AU6 at maximum or almost maximum intensity. It would be easiest for the classifier to classify AUs when they are at a higher intensity rather than a low intensity, especially since the classifier is trained with samples from the Cohn Kanade database at peak intensity.

From Table 7.10, which shows the output means of the correctly classified positive samples, we see that with the exception of intensity A and B, the output mean increases as the intensity increases. Thus, for intensity E, the classifier has an output closer to one than it has for intensity D.

Upon inspection of the false negatives, we see some combinations that cause appearance changes. The first is 6+43. AU43 is the eye closure. The combination of 6+43 raises the infraorbital triangle, and crow's feet wrinkles appear. Keypoints are extracted around crow's feet, and results in the classifier not correctly classifying AU6. 32% of the false negative samples contain the combination 6+43.

Another misclassified combination is 6+12. AU12 is the lip corner puller. The stronger the action of AU12, the more it hides the effects of AU6 and makes it more challenging to classify, especially when AU6 is relatively less intense than AU12. 79% of false negative samples contain the combination 6+12.

7.5.3 Results for AU7

AU7 is the lid tightener. From Table 7.7, AU7 has a low true positive rate of 54.55% and a high false positive rate of 39.73%, for a recognition rate of 58.95%. From Figure 7.20, the ROC curve for AU7 is close to the 45-degree line, indicating a less accurate test. Table B15 shows detailed results at certain threshold settings.

From Table 7.4, 73% of AU7 samples contain AU6 in the UNBC-McMaster database. Additionally, from Table 7.1, 30% of AU7 samples from the Cohn Kanade database contain AU6. Thus, the training sample set for the RNN contains 30% 6+7 samples, and the testing sample set has 72.73% 6+7 samples. Additionally, 41% of AU7 samples contain AU4, and 23% of AU7 samples contain AU9. When AU6 and AU7 are combined, AU6 can hide AU7. AU9 is the nose wrinkler. AU9 can be strong enough to narrow the eye aperture. AU9 can therefore obscure the presence of AU7, and unless the actions of AU7 and AU9 are sequential in a motion record, it is difficult to see the signs of AU7, especially if AU9 is strong. AU4 is the brow lowerer, which in lowering the brow may also narrow the eye aperture. If AU4 is present, the lower lid must also be raised in order to recognize AU7. The presence of AU7 in the combination 7+43 may be difficult to detect.

From the samples that are incorrectly classified as negative (false negative), all of them have a combination of AU4, AU6, AU9 or AU43. As discussed, these AUs can affect the presence of AU7. From Table 7.8, only 42.68% of 7C and 37.5% of 7D samples are recognized. We see that many samples of

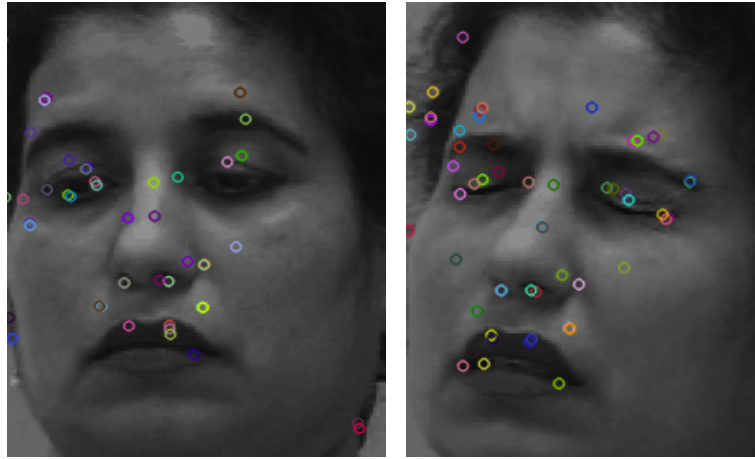


Figure 7.24: Extract showing 4d+6a+7e+10a+26a+43.

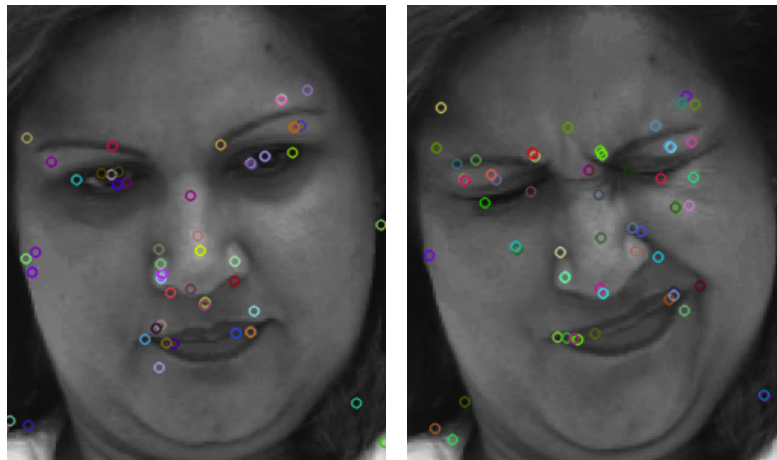


Figure 7.25: Extract showing 4d+6c+7c+9c+12c+43.

7C and 7D contain high intensities of AU6 or AU9, and contain AU43. The most influential AU on the classification of AU7 is AU6, and we see that high intensity AU6 samples affect the recognition of AU7. These samples are not correctly classified by the RNN. Therefore, the presence of AU6, AU9, or AU43 in these samples are inhibiting the recognition of AU7 in the samples. However, we see in Table 7.10 that of the samples that are being correctly classified as positive, the output increases as the intensity increases. In other words, the more intense that AU7 is, the closer to one the output is. For an intensity of A or B, which indicates only a slight presence of AU7, the classifier is outputting close to 0.5, indicating an almost uncertain response from the classifier. For an intensity of E (peak intensity), the classifier is outputting almost one, indicating a certain positive output. Figure 7.21 shows a correctly classified positive sample of AU7. This sample shows a low intensity of AU6. Figure 7.22 shows an incorrectly classified positive sample of AU7. This has a high intensity level of AU6, as well as AU9, AU4 and AU43. This shows the effect of the intensity of AU6 on the recognition of AU7. A high intensity AU6 sample can inhibit the recognition of AU7. The extracts from Figure 7.21 and Figure 7.22 show the first frame of the image sequence on the left, and the last frame of the image sequence on the right.

The false positive samples contain the combination of 4+6 or 6+12. From Table 7.1, we know that 85% of AU7 samples contain AU4 in the Cohn Kanade database. Since the classifier is trained with samples from the Cohn Kanade database, the high correlation with AU4 could cause false positive classifications for AU7. We also know that AU6 can be misclassified as AU7.



Figure 7.26: 4+7+9+17 from Cohn Kanade database.

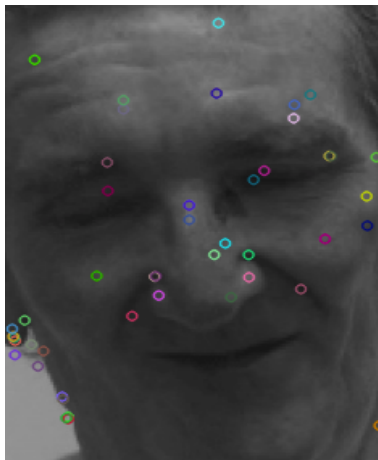


Figure 7.27: 6c+9b+12b from UNBC-McMaster database.

7.5.4 Results for AU9

AU9 is the nose wrinkler. From Table 7.7, AU9 has a true positive rate of 64.29%, a false positive rate of 28.57%, and a recognition rate of 70.24%. This is the lowest false positive rate, and the second highest recognition rate of the AUs tested in the UNBC-McMaster database. Table B16 shows detailed results at certain threshold settings.

From Table 7.8, we see that the lower intensity samples of AU9 are not correctly classified. Only 42.86% of 9B samples are correctly classified. Due to the strength of the keypoints, lower intensity AU9 samples do not have many keypoints extracted from the nose area. Figure 7.23 shows an image from the Cohn Kanade database in which AU9 is shown at peak intensity. Many keypoints cluster around the nose area and between the eyes. Figure 7.24 shows an image from the UNBC-McMaster database where AU9 is shown at intensity B. Due to the low intensity, keypoints are not picked up around the nose area. This sample is incorrectly classified as false by the RNN. Hence, AU9 is more easily classified when at higher intensities.

7.6 Comparison With Other Methods

We discussed other methods for facial expression recognition as well as AU recognition in Chapter 3. Often it is difficult to compare methods as they are tested on different databases. Additionally, the proportion of positive and negative samples used in samples sets often differs, which makes comparisons

	Gabor+SVM	Haar+Adaboost	Gabor+RNN	SURF+RNN
AU1	77.99%	82.83%	89.96%	70.01%
AU2	88.29%	93.26%	92.66%	80.51%
AU4	86.65%	85.23%	81.5%	65.29%
AU5	94.08%	94.39%	83.59%	82.93%
AU6	87.8%	93.39%	77.56%	82.08%
AU7	93.86%	88.31%	75.8%	71.88%
AU9	-	-	-	72.58%
AU15	94.96%	95.66%	74.41%	56.41%
AU17	90.67%	89.51%	72.33%	63.99%
AU20	96.51%	97.27%	84.27%	56%
AU25	96.49%	97.85%	91.1%	60.58%
AU27	98.16%	98.11%	87.5%	85.29%

Table 7.11: Comparison with other methods

challenging. We look at methods that are tested on FACS AUs to determine how our method compares to other state of the art methods. So far, SURF has not been tested on the Cohn Kanade database or any other FACS AU database.

A well established method is the use of Gabor filters to extract features and SVMs to classify. This has been tested on FACS AUs by Bartlett *et al.* [2002] and Donato *et al.* [1999]. Donato *et al.* [1999] tested twelve AUs. Bartlett *et al.* [2002] used Gabor filters, SVMs and HMMs to detect AUs 1, 2 and 4. This method does not use image sequences, and rather uses single static images for training and testing purposes. The first column in Table 7.11 shows the results achieved by Bartlett *et al.* [2002].

Whitehill and Omlin [2006a] used the combination of Haar filters for feature extraction and Adaboost for classification to detect 11 AUs. Since this method does not suffer from the inefficiency in memory usage and high redundancy of the Gabor representation, it is a faster method for AU detection. This method uses single static images. The second column in Table 7.11 shows the results of experimentation by Whitehill and Omlin [2006a].

Vadapalli [2014] made use of Gabor filters and RNNs to detect eleven AUs. Because of the temporal processing ability of the RNNs, Vadapalli [2014] used image sequences in which the face is shown to develop the AU from a neutral face to peak intensity. The thirs column in Table 7.11 shows the results achieved by Vadapalli [2014].

The recognition rates of these systems as well as the recognition rates of our system are shown in Table 7.11. The results achieved by our system are shown in the fourth column. We see that the SURF and RNN method shows a much lower recognition rate, especially for the lower face AUs. Additionally, SVMs on single static images show a much higher recognition rate in general than RNNs on image sequences.

7.7 Conclusion

This chapter shows the results for Phase I and Phase II testing. Results for the upper face AUs are promising, however the lower face AUs have a low recognition rate with the exception of AU27. When looking at other methods, we see that RNNs see a lower recognition rate in general than other methods which use single static images. This is because image sequences, especially image sequences of varying length, are more challenging to work with than single, static images. RNNs also see a lower recognition rate on lower face AUs than upper face AUs. The reasons for this can be investigated, and solutions

to combat the lower recognition rate on lower face AUs can be discussed further. Phase II of testing showed the trained RNNs being tested on the UNBC McMaster database which contains out-of-plane image sequences. Although the results are not as high as when tested on frontal sequences from the Cohn Kanade database, results are promising nonetheless with AU4 showing a particularly high recognition rate. The other AUs see recognition rates which are a promising start to further investigate out-of-plane head movement and the ability of RNNs to take advantage of temporal information to classify AUs.

Chapter 8

Conclusion

With the advances in robotics, the need for a robust FER system is apparent. As humans and robots interact more and more, it is important for robots to be able to understand a human's feelings and emotions in order to interpret a human's position and to therefore give the correct response. Facial expression recognition systems provide a visual interface between humans and computers. In particular, we look at the recognition of Facial Action Coding System's (FACS) Action Units (AUs). AUs provide a more fine-grained means of measurement and looks at muscle movement and the effects of contraction of specific muscles rather than facial expressions in general. The analysis of AUs has applications in many scientific subjects such as psychology, psychiatry, lie detection, pain assessment, and neurology. By understanding the reasons or causes of activation of specific AUs, it allows us to better understand the emotions, state, or feelings of humans. For example, by understanding which AUs are often active in the presence of pain, we can determine the level of pain a human is experiencing by analyzing their face without the need to the person explicitly stating their level of pain. Although this is easy for a human to recognize in another human, the same is not an easy task for a computer. Thus, it is important to develop a robust AU recognition system. It also has applications in determining whether a person is lying or not, or the level of sobriety of a person. The face is a fundamental tool in showing the many physical states or states of mind that the person is experiencing. By understanding the face, we can understand more and more of the human experience.

The purpose of this research was to develop a system to recognize the development of AUs in an image sequence. The image sequence shows the development of an AU or group of AUs from a neutral face. To achieve this, we use recurrent neural networks (RNNs) which allow us to do temporal processing. Many methods of feature extraction suffer from the high dimensionality problem. Thus, we use Speeded Up Robust Feature (SURF) as a feature extraction technique. This is since SURF features are a vector of only length 64. The Cohn Kanade database is used to train the RNN. We test our system on the UNBC-McMaster database which shows the development of pain in subjects. We wish to see if our system can recognize high levels of pain on subjects, in particular when the face moves in-plane and out-of-plane. Since the RNN can handle temporal processing, we hypothesized that it would be able to handle some level of head movements for some part of the sequence.

Our first research question asked if SURF descriptors extract enough information from the face to recognize AUs. From our results, we see that recognition rates are promising. Although other methods have seen a higher recognition rate in general, in particular methods using Gabor filters, our results for the first phase of testing have shown to be optimistic for a new method. Since the combination of SURF keypoints and RNNs has not been tested before there is room for improvement and refinement. We see that upper face AUs have a higher recognition rate than lower face AUs. This implies that not enough information is extracted from the lower face to provide adequate recognition.

Our second research question asked if it is possible to train an RNN to recognize AUs using the SURF feature vectors. When using SURF feature vectors as training inputs to our RNN, we see that the RNN does sufficiently train to recognize AUs. Although some recognition rates are low, we still see that the RNNs are correctly classifying AUs more than half of the time. Additionally, for a learning rate of

0.1 and 0.01, the RNNs are trained (i.e. average error of less than 1%) in less than the maximum allowed iterations. The Cohn Kanade database contains image sequences where the AUs are usually shown at maximum intensity. However, other databases contain instances where AUs are shown only at intensity A or B. This sometimes poses a problem for the classifier as it has been trained on samples of maximum intensity.

Our third research question asked the optimal number of keypoints which must be extracted from face images in order to optimally recognize FACS AUs. Thus, we tested how many keypoints need to be extracted from the face to recognize AUs. Recognition rate is maximized by maximizing true positive rate and minimizing false positive rate. We tested the optimal number of keypoints for three AUs: AU9, AU5, and AU17. For all three of those AUs, the true positive rate was maximized when 60 keypoints are extracted. In the case of false positives, 60 keypoints provided the lowest false positive rates for AU9 and AU17. It was not the lowest rate of false positives for AU5, however the number of keypoints that provided the lowest rate of false positives for AU5 also resulted in much worse true positive rates for all three AUs tested. Thus, we chose 60 keypoints as the optimal number as it provided the best results in general. We then used 60 keypoints for testing across all AU sample sets.

Our last research question asked if SURF descriptors and RNNs can be used to recognize FACS AUs when the face moves out-of-plane. After training our RNNs using samples from the Cohn Kanade database, we tested on samples from the UNBC-McMaster database. Results of testing showed promising recognition rates. Samples from this database show the subjects moving their heads either in-plane or out-of-plane. We see moderate to good recognition rates for all four AUs tested. However, we did see problems in recognizing AUs where the AU intensity levels were low.

The initial results of experimentation were promising, yet could possibly be improved. In the next chapter, we discuss future work which will attempt to achieve better recognition rates on both the Cohn Kanade database and UNBC-McMaster database testing.

Chapter 9

Future Work

We see that the upper face AUs see a higher recognition rate than lower face AUs. This could imply that not enough information is extracted from the lower face. We have used a global segmentation approach when extracting features from the face. To potentially combat the problem of not enough information extracted from the lower face, future work will include local segmentation of the face. This means that we will look at each area of the face individually. This will ensure that enough information is extracted from each area of the face in order to recognize AUs from that area. Additionally, SURF does not require that the face is normalized before feature extraction. Future work could attempt to normalize the face in order to remove the effect of different face structures. An example of this is to fix the number of pixels between the eyes in each face image, so that keypoint locations are more standardized across images.

Since our system was initially trained using samples at full intensity or near full intensity, future work could change the output structure of the RNN to include all intensities (from intensity A to E). For example, for an AU not present, output is zero. For an AU present at intensity A, output is one. For an AU present at intensity B, output is two, and so on. This will allow the classifier to distinguish between different intensity AUs. We saw this problem apparent when doing Phase II testing on the UNBC-McMaster database. Since our RNN is trained on maximum intensity samples, the classifier had problems recognizing AUs at intensity A or B.

Initially, we extracted 50 keypoints from the face to determine learning rate and number of hidden neurons. This number was randomly chosen following the work of Du *et al.* [2009], who found that the optimal number of keypoints extracted from the face is between 30 to 100 in order to recognize AUs. We found that a learning rate of 0.1 and 15 hidden neurons provide the overall highest true positive rate and lowest false positive rate. We did not initially test learning rate and hidden neurons on all numbers of keypoints. Thus, we know from initial experiments using 50 keypoints that a learning rate of 0.1 and 15 hidden neurons is optimal for 50 keypoints. We do not, however, know if that is the optimal RNN structure for all keypoint numbers. We also only used a subset of AUs for experimental purposes. Thus, for future work, we could look at each AU individually to determine the optimal number of keypoints for each AU, as well as re-test the RNN structure more extensively to find a more optimal solution (if any). This will help to determine whether some AUs require different numbers of keypoints extracted at a global face level or at a local level, as well as if different AUs are classified better with differing RNN structures.

The UNBC-McMaster database results were low, although a promising first start in out-of-plane AU recognition. This could be attributed to the low intensity of AUs shown in the database. Future work includes a method to combat the problem of differing intensities by changing the RNN output structure as discussed above. This should provide even more accurate results when tested on the UNBC-McMaster database. Additionally, future work will include an additional phase of testing on out-of-plane head movement in image sequences. The Cohn Kanade database is said to be publishing a future updated version (version three) showing some degree of out-of-plane head movement, which we will be testing on. Since the RNN handles temporal processing, it can handle instances of head movement where certain AUs are obscured or obstructed from view for some parts of the sequence.

References

- [Abdulrahman *et al.* 2014] Muzammil Abdulrahman, Tajuddeen R Gwadabe, Fahad J Abdu, and Alaa Eleyan. Gabor wavelet transform based facial expression recognition using PCA and LBP. In *Signal Processing and Communications Applications Conference (SIU), 2014 22nd*, pages 2265–2268. IEEE, 2014.
- [Ahmad *et al.* 2004] Abdul Manan Ahmad, Saliza Ismail, and DF Samaon. Recurrent neural network with backpropagation through time for speech recognition. In *IEEE International Symposium on Communications and Information Technology, 2004. ISCIT 2004.*, volume 1, pages 98–102. IEEE, 2004.
- [Anderson and McOwan 2006] Keith Anderson and Peter W McOwan. A real-time automated system for the recognition of human facial expressions. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 36(1):96–105, 2006.
- [Angelova *et al.* 2005] Anelia Angelova, Y Abu-Mostafam, and Pietro Perona. Pruning training sets for learning of object categories. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005. CVPR 2005.*, volume 1, pages 494–501. IEEE, 2005.
- [Arai 1993] Masahiko Arai. Bounds on the number of hidden units in binary-valued three-layer neural networks. *Neural Networks*, 6(6):855–860, 1993.
- [Bänziger and Scherer 2010] Tanja Bänziger and Klaus R Scherer. Introducing the geneva multimodal emotion portrayal (gemep) corpus. *Blueprint for affective computing: A sourcebook*, pages 271–294, 2010.
- [Bartlett *et al.* 1999] Marian Stewart Bartlett, Joseph C Hager, Paul Ekman, and Terrence J Sejnowski. Measuring facial expressions by computer image analysis. *Psychophysiology*, 36(2):253–263, 1999.
- [Bartlett *et al.* 2001] Marian S Bartlett, Bjorn Braathen, Gwen Littlewort-Ford, John Hershey, Ian Fasel, Tim Marks, Evan Smith, Terrence J Sejnowski, and Javier R Movellan. Automatic analysis of spontaneous facial behavior: A final project report. *University of California at San Diego*, 2001.
- [Bartlett *et al.* 2002] Marian Stewart Bartlett, B Braathen, TJ Sejnowski, JR Movellan, and Gwen Littlewort. A prototype for automatic recognition of spontaneous facial actions. In *Advances in neural information processing systems*, pages 1271–1278, 2002.
- [Bartlett *et al.* 2003] Marian Stewart Bartlett, Gwen Littlewort, Ian Fasel, and Javier R Movellan. Real time face detection and facial expression recognition: Development and applications to human computer interaction. In *Conference on Computer Vision and Pattern Recognition Workshop, 2003. CVPRW'03*, volume 5, pages 53–53. IEEE, 2003.
- [Bay *et al.* 2006] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. SURF: Speeded up robust features. In *Computer Vision–ECCV 2006*, pages 404–417. Springer, 2006.

- [Bay *et al.* 2008] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (SURF). *Computer vision and image understanding*, 110(3):346–359, 2008.
- [Bengio and LeCun 2007] Yoshua Bengio and Yann LeCun. Scaling learning algorithms towards AI. *Large-Scale Kernel Machines*, 34:1–41, 2007.
- [Bertsekas and Tsitsiklis 1995] Dimitri P Bertsekas and John N Tsitsiklis. Neuro-dynamic programming: an overview. In *Decision and Control, 1995., Proceedings of the 34th IEEE Conference on*, volume 1, pages 560–564. IEEE, 1995.
- [Bhuiyan and Liu 2007] Al-Amin Bhuiyan and Chang Hong Liu. On face recognition using gabor filters. In *Proceedings of world academy of science, engineering and technology*, volume 22, pages 51–56, 2007.
- [Black and Yacoob 1997] Michael J Black and Yaser Yacoob. Recognizing facial expressions in image sequences using local parameterized models of image motion. *International Journal of Computer Vision*, 25(1):23–48, 1997.
- [Blanz and Vetter 1999] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3D faces. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 187–194. ACM Press/Addison-Wesley Publishing Co., 1999.
- [Blanz and Vetter 2003] Volker Blanz and Thomas Vetter. Face recognition based on fitting a 3D morphable model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(9):1063–1074, 2003.
- [Blanz *et al.* 2005] Volker Blanz, Patrick Grother, P Jonathon Phillips, and Thomas Vetter. Face recognition based on frontal views generated from non-frontal images. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005. CVPR 2005*, volume 2, pages 454–461. IEEE, 2005.
- [Bosch *et al.* 2006] Anna Bosch, Andrew Zisserman, and Xavier Muñoz. Scene classification via pLSA. In *Computer Vision–ECCV 2006*, pages 517–530. Springer, 2006.
- [Boser *et al.* 1992] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM, 1992.
- [Bradski 2000] G. Bradski. The OpenCV Library. *Dr. Dobb’s Journal of Software Tools*, 2000.
- [Brandt *et al.* 2004] Thomas Brandt, Ralf Stemmer, and Andry Rakotonirainy. Affordable visual driver monitoring system for fatigue and monotony. In *IEEE International Conference on Systems, Man and Cybernetics, 2004*, volume 7, pages 6451–6456. IEEE, 2004.
- [Burghouts and Geusebroek 2009] Gertjan J Burghouts and Jan-Mark Geusebroek. Performance evaluation of local colour invariants. *Computer Vision and Image Understanding*, 113(1):48–62, 2009.
- [Butko *et al.* 2011] Nicholas J Butko, Georgios Theodorou, Matthai Philipose, and Javier R Movellan. Automated facial affect analysis for one-on-one tutoring applications. In *2011 IEEE International Conference on Automatic Face & Gesture Recognition and Workshops (FG 2011)*, pages 382–387. IEEE, 2011.
- [Chen *et al.* 2000] Li-Fen Chen, Hong-Yuan Mark Liao, Ming-Tat Ko, Ja-Chen Lin, and Gwo-Jong Yu. A new LDA-based face recognition system which can solve the small sample size problem. *Pattern recognition*, 33(10):1713–1726, 2000.

- [Cohen *et al.* 2003a] Ira Cohen, Nicu Sebe, Ashutosh Garg, Lawrence S Chen, and Thomas S Huang. Facial expression recognition from video sequences: temporal and static modeling. *Computer Vision and Image Understanding*, 91(1):160–187, 2003.
- [Cohen *et al.* 2003b] Ira Cohen, Nicu Sebe, FG Gozman, Marcelo Cesar Cirelo, and Thomas S Huang. Learning bayesian network classifiers for facial expression recognition both labeled and unlabeled data. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings*, volume 1, pages I–595. IEEE, 2003.
- [Cohn *et al.* 1999] Jeffrey F Cohn, Adena J Zlochower, James Lien, and Takeo Kanade. Automated face analysis by feature point tracking has high concurrent validity with manual FACS coding. *Psychophysiology*, 36(1):35–43, 1999.
- [Cohn *et al.* 2001] Jeffrey F Cohn, Takeo Kanade, Tsuyoshi Moriyama, Zara Ambadar, Jing Xiao, Jiang Gao, and Hiroki Imamura. *Final Report, CIA Contract# 2000-A128400-000: A Comparative Study of Alternative FACS Coding Algorithms*, volume 9. November, 2001.
- [Cottrell and Metcalfe 1990] Garrison W Cottrell and Janet Metcalfe. Empath: Face, emotion, and gender recognition using holons. In *Proceedings of the 1990 conference on Advances in neural information processing systems 3*, pages 564–571. Morgan Kaufmann Publishers Inc., 1990.
- [Darwin 1872] Charles Darwin. The expression of the emotions in man and animals. *London, UK: Murray (3rd edn, ed. P. Ekman, London, UK: HarperCollins, 1998)*, 1872.
- [Darwish *et al.* 1997] AM Darwish, MS Bedair, and SI Shaheen. Adaptive resampling algorithm for image zooming. *IEE Proceedings-Vision, Image and Signal Processing*, 144(4):207–212, 1997.
- [Donato *et al.* 1999] Gianluca Donato, Marian Stewart Bartlett, Joseph C. Hager, Paul Ekman, and Terrence J. Sejnowski. Classifying facial actions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(10):974–989, 1999.
- [Draper *et al.* 2003] Bruce A Draper, Kyungim Baek, Marian Stewart Bartlett, and J Ross Beveridge. Recognizing faces with PCA and ICA. *Computer vision and image understanding*, 91(1):115–137, 2003.
- [Du *et al.* 2009] Geng Du, Fei Su, and Anni Cai. Face recognition using SURF features. In *Proc. of SPIE Vol.*, volume 7496, pages 749628–1, 2009.
- [Ekman and Friesen 1971] Paul Ekman and Wallace V Friesen. Constants across cultures in the face and emotion. *Journal of personality and social psychology*, 17(2):124, 1971.
- [Ekman and Friesen 1978] Paul Ekman and Wallace V Friesen. Facial action coding system: A technique for the measurement of facial movement. palo alto. *CA: Consulting Psychologists Press. Ellsworth, PC, & Smith, CA (1988). From appraisal to emotion: Differences among unpleasant feelings. Motivation and Emotion*, 12:271–302, 1978.
- [Ekman and Rosenberg 1997] Paul Ekman and Erika L Rosenberg. *What the face reveals: Basic and applied studies of spontaneous expression using the Facial Action Coding System (FACS)*. Oxford University Press, 1997.
- [Ekman *et al.* 2002] Paul Ekman, Wallace V Friesen, and Joseph C Hager. Facial action coding system. *A Human Face*, 2002.
- [Elman 1990] Jeffrey L Elman. Finding structure in time. *Cognitive science*, 14(2):179–211, 1990.
- [Essa and Pentland 1997] Irfan A. Essa and Alex Paul Pentland. Coding, analysis, interpretation, and recognition of facial expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):757–763, 1997.

- [FACS] FACS. Areas of application. <http://www.facsencodinggroup.com/about/areas-of-application>.
- [Fischler and Bolles 1981] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.
- [Freund and Schapire 1995] Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *Computational learning theory*, pages 23–37. Springer, 1995.
- [Fujita 1998] Osamu Fujita. Statistical estimation of the number of hidden units for feedforward neural networks. *Neural Networks*, 11(5):851–859, 1998.
- [Gao *et al.* 2008] Wen Gao, Bo Cao, Shiguang Shan, Xilin Chen, Delong Zhou, Xiaohua Zhang, and Debin Zhao. The cas-peal large-scale chinese face database and baseline evaluations. *IEEE transactions on systems, man and cybernetics part A systems and humans*, 38(1):149, 2008.
- [Geman *et al.* 1992] Stuart Geman, Elie Bienenstock, and René Doursat. Neural networks and the bias/variance dilemma. *Neural computation*, 4(1):1–58, 1992.
- [Georghiades 1997] A Georghiades. Yale face database. *Center for computational Vision and Control at Yale University*, <http://cvc.yale.edu/projects/yalefaces/yalefa>, 1997.
- [Hagiwara 1994] Masafumi Hagiwara. A simple and effective method for removal of hidden units and weights. *Neurocomputing*, 6(2):207–218, 1994.
- [Harris and Stephens 1988] Chris Harris and Mike Stephens. A combined corner and edge detector. In *Alvey vision conference*, volume 15, page 50. Manchester, UK, 1988.
- [Heiselet *et al.* 2001] B Heiselet, Thomas Serre, Massimiliano Pontil, and Tomaso Poggio. Component-based face detection. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2001. CVPR 2001*, volume 1, pages I–657. IEEE, 2001.
- [Heller and Haynal 2002] Michel Heller and Véronique Haynal. The doctor’s face: A mirror of his patient’s suicidal projects. *Body Psychotherapy in Progressive and Chronic Disorders*, page 46, 2002.
- [Hjortsjö 1969] Carl-Herman Hjortsjö. *Man’s face and mimic language*. Studen litteratur, 1969.
- [Hopfield 1982] John J Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8):2554–2558, 1982.
- [Hornik *et al.* 1989] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.
- [Hu *et al.* 2003] Changbo Hu, Rogerio Feris, and Matthew Turk. Real-time view-based face alignment using active wavelet networks. In *IEEE International Workshop on Analysis and Modeling of Faces and Gestures, 2003. AMFG 2003*, pages 215–221. IEEE, 2003.
- [Huang and Tai 2012] Hung-Fu Huang and Shen-Chuan Tai. Facial expression recognition using new feature extraction algorithm. *Electronic Letters on Computer Vision and Image Analysis*, 11(1):41–54, 2012.
- [Ishiguro *et al.* 2001] Hiroshi Ishiguro, Tetsuo Ono, Michita Imai, Takeshi Maeda, Takayuki Kanda, and Ryohei Nakatsu. Robovie: an interactive humanoid robot. *Industrial robot: An international journal*, 28(6):498–504, 2001.

- [Islam and Murase 2001] Md Monirul Islam and Kazuyuki Murase. A new algorithm to design compact two-hidden-layer artificial neural networks. *Neural Networks*, 14(9):1265–1278, 2001.
- [Jain and Li 2005] Anil K. Jain and Stan Z. Li. *Handbook of Face Recognition*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.
- [Ji and Yang 2002] Qiang Ji and Xiaojie Yang. Real-time eye, gaze, and face pose tracking for monitoring driver vigilance. *Real-Time Imaging*, 8(5):357–377, 2002.
- [Jiang *et al.* 2008] Nan Jiang, Zhaozhi Zhang, Xiaomin Ma, and Jian Wang. The lower bound on the number of hidden neurons in multi-valued multi-threshold neural networks. In *Intelligent Information Technology Application, 2008. IITA'08. Second International Symposium on*, volume 1, pages 103–107. IEEE, 2008.
- [Jordan 1986] Michael I Jordan. Attractor dynamics and parallelism in a connectionist sequential machine. 1986.
- [Kamachi *et al.* 1998] Miyuki Kamachi, Michael Lyons, and Jiro Gyoba. The japanese female facial expression (JAFFE) database. URL <http://www.kasrl.org/jaffe.html>, 21, 1998.
- [Kanade *et al.* 2000] Takeo Kanade, Jeffrey F Cohn, and Yingli Tian. Comprehensive database for facial expression analysis. In *Fourth IEEE International Conference on Automatic Face and Gesture Recognition, 2000. Proceedings*, pages 46–53. IEEE, 2000.
- [Kandil *et al.* 1993] N Kandil, K Khorasani, RV Patel, and VK Sood. Optimum learning rate for back-propagation neural networks. In *Electrical and Computer Engineering, 1993. Canadian Conference on*, pages 465–468. IEEE, 1993.
- [Karhunen *et al.* 1997] Juha Karhunen, Erkki Oja, Liuyue Wang, Ricardo Vigario, and Jyrki Joutsensalo. A class of neural networks for independent component analysis. *IEEE Transactions on Neural Networks*, 8(3):486–504, 1997.
- [Karsoliya 2012] Saurabh Karsoliya. Approximating number of hidden layer neurons in multiple hidden layer bpnn architecture. *International Journal of Engineering Trends and Technology*, 3(6):713–717, 2012.
- [Ke and Liu 2008] Jinchuan Ke and Xinzhe Liu. Empirical analysis of optimal hidden neurons in neural network modeling for stock prediction. In *Computational Intelligence and Industrial Application, 2008. PACIIA'08. Pacific-Asia Workshop on*, volume 2, pages 828–832. IEEE, 2008.
- [Keeni *et al.* 1999] Kanad Keeni, Kenji Nakayama, and Hiroshi Shimodaira. Estimation of initial weights and hidden units for fast learning of multilayer neural networks for pattern classification. In *Neural Networks, 1999. IJCNN'99. International Joint Conference on*, volume 3, pages 1652–1656. IEEE, 1999.
- [Keltner and Bonanno 1997] Dacher Keltner and George A Bonanno. A study of laughter and dissociation: distinct correlates of laughter and smiling during bereavement. *Journal of personality and social psychology*, 73(4):687, 1997.
- [Kim and Dahyot 2008] Donghoon Kim and Rozenn Dahyot. Face components detection using SURF descriptors and SVMs. In *International Machine Vision and Image Processing Conference, 2008. IMVIP'08*, pages 51–56. IEEE, 2008.
- [Kirby and Sirovich 1990] Michael Kirby and Lawrence Sirovich. Application of the Karhunen-Loeve procedure for the characterization of human faces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(1):103–108, 1990.

- [Kobayashi and Hara 1993] Hiroshi Kobayashi and Fumio Hara. Dynamic recognition of basic facial expressions by discrete-time recurrent neural network. In *Proceedings of 1993 International Joint Conference on Neural Networks, 1993. IJCNN'93-Nagoya*, volume 1, pages 155–158. IEEE, 1993.
- [Kotsia and Pitas 2007] Irene Kotsia and Ioannis Pitas. Facial expression recognition in image sequences using geometric deformation features and support vector machines. *Image Processing, IEEE Transactions on*, 16(1):172–187, 2007.
- [Krogh *et al.* 1995] Anders Krogh, Jesper Vedelsby, et al. Neural network ensembles, cross validation, and active learning. *Advances in neural information processing systems*, pages 231–238, 1995.
- [Lades *et al.* 1993] Martin Lades, Jan C Vorbruggen, Joachim Buhmann, Jörg Lange, Christoph von der Malsburg, Rolf P Wurtz, and Wolfgang Konen. Distortion invariant object recognition in the dynamic link architecture. *IEEE Transactions on Computers*, 42(3):300–311, 1993.
- [Lam and Yan 1998] Kin-Man Lam and Hong Yan. An analytic-to-holistic approach for face recognition based on a single frontal view. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(7):673–686, 1998.
- [Laptev and Lindeberg 2006] Ivan Laptev and Tony Lindeberg. Local descriptors for spatio-temporal recognition. In *Spatial Coherence for Visual Motion Analysis*, pages 91–103. Springer, 2006.
- [Li *et al.* 1995] Jin-Yan Li, Tommy WS Chow, and Ying-Lin Yu. The estimation theory and optimization algorithm for the number of hidden units in the higher-order feedforward neural network. In *Neural Networks, 1995. Proceedings., IEEE International Conference on*, volume 3, pages 1229–1233. IEEE, 1995.
- [Lien *et al.* 1998] James J Lien, Takeo Kanade, Jeffrey F Cohn, and Ching-Chung Li. Automated facial expression recognition based on FACS action units. In *Third IEEE International Conference on Automatic Face and Gesture Recognition, 1998. Proceedings*, pages 390–395. IEEE, 1998.
- [Lien *et al.* 2000] James Jenn-Jier Lien, Takeo Kanade, Jeffrey F Cohn, and Ching-Chung Li. Detection, tracking, and classification of action units in facial expression. *Robotics and Autonomous Systems*, 31(3):131–146, 2000.
- [Lin *et al.* 2012] Shinfeng D Lin, B Liu, and J Lin. Combining speeded-up robust features with principal component analysis in face recognition system. *International Journal of Innovative Computing, Information and Control*, 8(12):8545–56, 2012.
- [Lindeberg 1998] Tony Lindeberg. Feature detection with automatic scale selection. *International journal of computer vision*, 30(2):79–116, 1998.
- [Littlewort *et al.* 2002] Gwen Littlewort, Ian Fasel, M Stewart Bartlett, and Javier R Movellan. Fully automatic coding of basic expressions from video. *INC MPLab TR*, pages 53–56, 2002.
- [Liu 2004] Chengjun Liu. Gabor-based kernel PCA with fractional power polynomial models for face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(5):572–581, 2004.
- [Lowe 1999] David G Lowe. Object recognition from local scale-invariant features. In *The proceedings of the seventh IEEE international conference on Computer vision, 1999*, volume 2, pages 1150–1157. Ieee, 1999.
- [Lucey *et al.* 2011] Patrick Lucey, Jeffrey F Cohn, Kenneth M Prkachin, Patricia E Solomon, and Iain Matthews. Painful data: The UNBC-McMaster shoulder pain expression archive database. In *2011 IEEE International Conference on Automatic Face & Gesture Recognition and Workshops (FG 2011)*, pages 57–64. IEEE, 2011.

- [Lyons *et al.* 1998] Michael Lyons, Shigeru Akamatsu, Miyuki Kamachi, and Jiro Gyoba. Coding facial expressions with gabor wavelets. In *Third IEEE International Conference on Automatic Face and Gesture Recognition, 1998. Proceedings*, pages 200–205. IEEE, 1998.
- [Ma *et al.* 2002] Lia Ma, Yunhong Wang, and Tieniu Tan. Iris recognition based on multichannel gabor filtering. In *Proc. Fifth Asian Conf. Computer Vision*, volume 1, pages 279–283, 2002.
- [Martinez 1998] Aleix M Martinez. The AR face database. *CVC Technical Report*, 24, 1998.
- [Mikolajczyk and Schmid 2004] Krystian Mikolajczyk and Cordelia Schmid. Scale & affine invariant interest point detectors. *International journal of computer vision*, 60(1):63–86, 2004.
- [Mozer 1995] Michael C Mozer. A focused backpropagation algorithm for temporal pattern recognition. *Backpropagation: Theory, architectures, and applications*, page 137, 1995.
- [Ni and Schneiderman 2005] Jiang Ni and Henry Schneiderman. Face view synthesis across large angles. In *Analysis and Modelling of Faces and Gestures*, pages 364–376. Springer, 2005.
- [Niese *et al.* 2012] R Niese, A Al-Hamadi, A Farag, H Neumann, and B Michaelis. Facial expression recognition based on geometric and optical flow features in colour image sequences. *IET computer vision*, 6(2):79–89, 2012.
- [Ojala *et al.* 1994] Timo Ojala, Matti Pietikainen, and David Harwood. Performance evaluation of texture measures with classification based on kullback discrimination of distributions. In *Proceedings of the 12th IAPR International Conference on Pattern Recognition, 1994. Vol. 1-Conference A: Computer Vision & Image Processing*, volume 1, pages 582–585. IEEE, 1994.
- [Onoda 1995] Takashi Onoda. Neural network information criterion for the optimal number of hidden units. In *Neural Networks, 1995. Proceedings., IEEE International Conference on*, volume 1, pages 275–280. IEEE, 1995.
- [Owusu *et al.* 2014a] Ebenezer Owusu, Yongzhao Zhan, and Qi Rong Mao. A neural-adaboost based facial expression recognition system. *Expert Systems with Applications*, 41(7):3383–3390, 2014.
- [Owusu *et al.* 2014b] Ebenezer Owusu, Yonzhao Zhan, and Qi Rong Mao. An svm-adaboost facial expression recognition system. *Applied intelligence*, 40(3):536–545, 2014.
- [Padgett and Cottrell 1997] Curtis Padgett and Garrison W Cottrell. Representing face images for emotion classification. *Advances in neural information processing systems*, pages 894–900, 1997.
- [Pantic and Rothkrantz 2000] Maja Pantic and Leon JM Rothkrantz. Expert system for automatic analysis of facial expressions. *Image and Vision Computing*, 18(11):881–905, 2000.
- [Papageorgiou *et al.* 1998] Constantine P Papageorgiou, Michael Oren, and Tomaso Poggio. A general framework for object detection. In *Sixth International Conference on Computer Vision, 1998*, pages 555–562. IEEE, 1998.
- [Park *et al.* 1993] JH Park, YS Kim, IK Eom, and KY Lee. Economic load dispatch for piecewise quadratic cost function using hopfield neural network. *Power Systems, IEEE Transactions on*, 8(3):1030–1038, 1993.
- [Parke 1972] Frederick I Parke. Computer generated animation of faces. In *Proceedings of the ACM annual conference-Volume 1*, pages 451–457. ACM, 1972.
- [Parke 1974] Frederick Ira Parke. A parametric model for human faces. Technical report, DTIC Document, 1974.

- [Penev and Atick 1996] Penio S Penev and Joseph J Atick. Local feature analysis: A general statistical theory for object representation. *Network: computation in neural systems*, 7(3):477–500, 1996.
- [Pentland *et al.* 1994] Alexander Pentland, Baback Moghaddam, and Thad Starner. View-based and modular eigenspaces for face recognition. In *Proceedings CVPR'94., 1994 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1994*, pages 84–91. IEEE, 1994.
- [Phillips *et al.* 1996] P Jonathon Phillips, Patrick J Rauss, Sandor Z Der, et al. *FERET (face recognition technology) recognition algorithm development and test results*. Army Research Laboratory Adelphi, MD, 1996.
- [Prkachin and Solomon 2008] Kenneth M Prkachin and Patricia E Solomon. The structure, reliability and validity of pain expression: Evidence from patients with shoulder pain. *Pain*, 139(2):267–274, 2008.
- [Prkachin 1992] Kenneth M Prkachin. The consistency of facial expressions of pain: a comparison across modalities. *Pain*, 51(3):297–306, 1992.
- [Rabiner 1989] Lawrence Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.
- [Riedmiller and Braun 1993] Martin Riedmiller and Heinrich Braun. A direct adaptive method for faster backpropagation learning: The rprop algorithm. In *Neural Networks, 1993., IEEE International Conference on*, pages 586–591. IEEE, 1993.
- [Robinson and Fallside 1987] AJ Robinson and Frank Fallside. *The utility driven dynamic error propagation network*. University of Cambridge Department of Engineering, 1987.
- [Rosenberg *et al.* 2001] Erika L Rosenberg, Paul Ekman, Wei Jiang, Michael Babyak, R Edward Coleman, Michael Hanson, Christopher O'Connor, Robert Waugh, and James A Blumenthal. Linkages between facial expressions of anger and transient myocardial ischemia in men with coronary artery disease. *Emotion*, 1(2):107, 2001.
- [Rosenblum *et al.* 1996] Mark Rosenblum, Yaser Yacoob, and Larry S Davis. Human expression recognition from motion using a radial basis function network architecture. *IEEE Transactions on Neural Networks*, 7(5):1121–1138, 1996.
- [Rumelhart *et al.* 1985] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning internal representations by error propagation. Technical report, DTIC Document, 1985.
- [Rumelhart *et al.* 1988] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *Cognitive modeling*, 5, 1988.
- [Samaria and Harter 1994] Ferdinando S Samaria and Andy C Harter. Parameterisation of a stochastic model for human face identification. In *Proceedings of the Second IEEE Workshop on Applications of Computer Vision, 1994.*, pages 138–142. IEEE, 1994.
- [Sartori and Antsaklis 1991] Michael A Sartori and Panos J Antsaklis. A simple method to derive bounds on the size and to train multilayer neural networks. *Neural Networks, IEEE Transactions on*, 2(4):467–471, 1991.
- [Schneiderman and Kanade 2000] Henry Schneiderman and Takeo Kanade. A statistical method for 3d object detection applied to faces and cars. In *Proceedings. IEEE Conference on Computer Vision and Pattern Recognition, 2000*, volume 1, pages 746–751. IEEE, 2000.
- [Sirovich and Kirby 1987] Lawrence Sirovich and Michael Kirby. Low-dimensional procedure for the characterization of human faces. *JOSA A*, 4(3):519–524, 1987.

- [Smith 2002] Lindsay I Smith. A tutorial on principal components analysis. *Cornell University, USA*, 51:52, 2002.
- [Suwa *et al.* 1978] Motoi Suwa, Noboru Sugie, and Keisuke Fujimora. A preliminary note on pattern recognition of human emotional expression. In *International joint conference on pattern recognition*, pages 408–410, 1978.
- [Tai and Chung 2007] SC Tai and KC Chung. Automatic facial expression recognition system using neural networks. In *TENCON 2007-2007 IEEE Region 10 Conference*, pages 1–4. IEEE, 2007.
- [Tao and Huang 1999] Hai Tao and Thomas S Huang. Explanation-based facial motion tracking using a piecewise bezier volume deformation model. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1999*, volume 1. IEEE, 1999.
- [Tao and Tan 2005] Jianhua Tao and Tieniu Tan. Affective computing: A review. In *Affective computing and intelligent interaction*, pages 981–995. Springer, 2005.
- [Tian *et al.* 2000] Ying-li Tian, Takeo Kanade, and Jeffrey F Cohn. Eye-state action unit detection by gabor wavelets. In *Advances in Multimodal Interfaces ICMI 2000*, pages 143–150. Springer, 2000.
- [Tian *et al.* 2001] Y-I Tian, Takeo Kanade, and Jeffrey F Cohn. Recognizing action units for facial expression analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(2):97–115, 2001.
- [Tian *et al.* 2002] Ying-li Tian, Takeo Kanade, and Jeffrey F Cohn. Evaluation of gabor-wavelet-based facial action unit recognition in image sequences of increasing complexity. In *Proceedings. Fifth IEEE International Conference on Automatic Face and Gesture Recognition, 2002*, pages 229–234. IEEE, 2002.
- [Tian *et al.* 2003] Ying-li Tian, Lisa Brown, Arun Hampapur, Sharat Pankanti, Andrew Senior, and Ruud Bolle. Real world real-time automatic recognition of facial expressions. In *Proceedings of IEEE workshop on*. Citeseer, 2003.
- [Tiflin and Omlin 2003] Conrad Tiflin and Christian W Omlin. LSTM recurrent neural networks for signature verification. In *Proc. Southern African Telecommunication Networks & Applications Conference (SATNAC 2003)*, 2003.
- [Tino *et al.* 2001] Peter Tino, Christian Schittenkopf, and Georg Dorffner. Financial volatility trading using recurrent neural networks. *IEEE Transactions on Neural Networks*, 12(4):865–874, 2001.
- [Trenn 2008] Stephan Trenn. Multilayer perceptrons: approximation order and necessary number of hidden units. *Neural Networks, IEEE Transactions on*, 19(5):836–844, 2008.
- [Turk and Pentland 1991] Matthew Turk and Alex Pentland. Eigenfaces for recognition. *Journal of cognitive neuroscience*, 3(1):71–86, 1991.
- [Vadapalli 2014] Hima Vadapalli. Facial action unit recognition from video streams with recurrent neural networks. *International Journal of Computer Applications*, 96(19):31–39, 2014.
- [Van De Sande *et al.* 2010] Koen EA Van De Sande, Theo Gevers, and Cees GM Snoek. Evaluating color descriptors for object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1582–1596, 2010.
- [Van De Weijer and Schmid 2006] Joost Van De Weijer and Cordelia Schmid. Coloring local feature extraction. In *Computer Vision—ECCV 2006*, pages 334–348. Springer, 2006.
- [Viola and Jones 2001] Paul Viola and Michael Jones. Robust real-time object detection. *International Journal of Computer Vision*, 57(2):137–154, 2001.

- [Wan *et al.* 2012] Chuan Wan, Yantao Tian, and Shuایشی Liu. Facial expression recognition in video sequences. In *Intelligent Control and Automation (WCICA), 2012 10th World Congress on*, pages 4766–4770. IEEE, 2012.
- [Wang and Huang 2011] Xiaoyu Wang and Yong Huang. Convergence study in extended kalman filter-based training of recurrent neural networks. *Neural Networks, IEEE Transactions on*, 22(4):588–600, 2011.
- [Wang *et al.* 2005] Changhu Wang, Shuicheng Yan, Hongjiang Zhang, and Weiyang Ma. Realistic 3D face modeling by fusing multiple 2D images. In *Proceedings of the 11th International Multimedia Modelling Conference, 2005. MMM 2005*, pages 139–146. IEEE, 2005.
- [Wen and Huang 2003] Zhen Wen and Thomas S Huang. Capturing subtle facial motions in 3D face tracking. In *Ninth IEEE International Conference on Computer Vision, 2003. Proceedings*, pages 1343–1350. IEEE, 2003.
- [Werbos 1988] Paul J Werbos. Generalization of backpropagation with application to a recurrent gas market model. *Neural Networks*, 1(4):339–356, 1988.
- [Whitehill and Omlin 2006a] Jacob Whitehill and Christian W Omlin. Haar features for FACS AU recognition. In *7th International Conference on Automatic Face and Gesture Recognition, 2006. FGR 2006*, pages 5–pp. IEEE, 2006.
- [Whitehill and Omlin 2006b] Jacob Whitehill and Christian W Omlin. Local versus global segmentation for facial expression recognition. In *7th International Conference on Automatic Face and Gesture Recognition, 2006. FGR 2006*, pages 357–362. IEEE, 2006.
- [Whitehill *et al.* 2008] Jacob Whitehill, Marian Bartlett, and Javier Movellan. Automatic facial expression recognition for intelligent tutoring systems. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08*, pages 1–6. IEEE, 2008.
- [Wu *et al.* 2002] Haiyuan Wu, Yukio Yoshida, and Tadayoshi Shioyama. Optimal gabor filters for high speed face identification. In *16th International Conference on Pattern Recognition, 2002. Proceedings*, volume 1, pages 107–110. IEEE, 2002.
- [Xiao *et al.* 2003] Jing Xiao, Tsuyoshi Moriyama, Takeo Kanade, and Jeffrey F Cohn. Robust full-motion recovery of head by dynamic templates and re-registration techniques. *International Journal of Imaging Systems and Technology*, 13(1):85–94, 2003.
- [Yang *et al.* 2004] Jian Yang, David Zhang, Alejandro F Frangi, and Jing-yu Yang. Two-dimensional pca: a new approach to appearance-based face representation and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(1):131–137, 2004.
- [Zavaschi *et al.* 2013] Thiago HH Zavaschi, Alceu S Britto Jr, Luiz ES Oliveira, and Alessandro L Korerich. Fusion of feature sets and classifiers for facial expression recognition. *Expert Systems with Applications*, 40(2):646–655, 2013.
- [Zeng and Yeung 2006] Xiaoqin Zeng and Daniel S Yeung. Hidden neuron pruning of multilayer perceptrons using a quantified sensitivity measure. *Neurocomputing*, 69(7):825–837, 2006.
- [Zhang *et al.* 1998] Zhengyou Zhang, Michael Lyons, Michael Schuster, and Shigeru Akamatsu. Comparison between geometry-based and gabor-wavelets-based facial expression recognition using multi-layer perceptron. In *Proceedings. Third IEEE International Conference on Automatic Face and Gesture Recognition, 1998*, pages 454–459. IEEE, 1998.
- [Zhang *et al.* 2003] Zhaozhi Zhang, Xiaomin Ma, and Yixian Yang. Bounds on the number of hidden neurons in three-layer binary neural networks. *Neural networks*, 16(7):995–1002, 2003.

[Zhao *et al.* 2002] Liang Zhao, Gopal Pingali, and Ingrid Carlbom. Real-time head orientation estimation using neural networks. In *Proceedings. 2002 International Conference on Image Processing. 2002*, volume 1, pages I–297. IEEE, 2002.

Appendix A

FACS AUs and their descriptions



Upper Face Action Units					
AU 1	AU 2	AU 4	AU 5	AU 6	AU 7
					
Inner Brow Raiser	Outer Brow Raiser	Brow Lowerer	Upper Lid Raiser	Cheek Raiser	Lid Tightener
*AU 41	*AU 42	*AU 43	AU 44	AU 45	AU 46
					
Lid Droop	Slit	Eyes Closed	Squint	Blink	Wink
Lower Face Action Units					
AU 9	AU 10	AU 11	AU 12	AU 13	AU 14
					
Nose Wrinkler	Upper Lip Raiser	Nasolabial Deepener	Lip Corner Puller	Cheek Puffer	Dimpler
AU 15	AU 16	AU 17	AU 18	AU 20	AU 22
					
Lip Corner Depressor	Lower Lip Depressor	Chin Raiser	Lip Puckerer	Lip Stretcher	Lip Funneler
AU 23	AU 24	*AU 25	*AU 26	*AU 27	AU 28
					
Lip Tightener	Lip Pressor	Lips Part	Jaw Drop	Mouth Stretch	Lip Suck

Figure A.1: Some of the AUs represented in the Cohn-Kanade database.

AU	Name
9	Nose wrinkler
10	Upper lip raiser
11	Nasolabial furrow deepender
12	Lip corner puller
13	Sharp lip puller
14	Dimpler
15	Lip corner depressor
16	Lower lip depress
17	Chin raiser
18	Lip pucker
20	Lip stretcher
22	Lip funneler
23	Lip tightener
24	Lip presser
28	Lips suck
72	Lower face not visible

Table A.1: Lower face AUs.

AU	Name
1	Inner brow raiser
2	Outer brow raiser
4	Brow lowerer
5	Upper lid raiser
6	Cheek raiser
7	Lids tightener
43	Eye closure
45	Blink
46	Wink
70	Brows not visible
71	Eyes not visible

Table A.2: Upper face AUs.

AU	Name
51	Turned left
52	Turned right
53	Up
54	Down
55	Tilted left
56	Tilted right
57	Forward
58	Back

Table A.3: Head Positions.

AU	Name
61	Eyes left
62	Eyes right
63	Eyes up
64	Eyes down
65	Walleye
66	Crosseye

Table A.4: Eye Positions.

AU	Name
25	Lips parted
26	Jaw dropped
27	Mouth stretched

Table A.5: Lip Parting and Jaw Opening.

AU	Name
8	Lips toward each other
19	Tongue showing
21	Neck tightener
29	Jaw thruster
30	Jaw sideways
31	Jaw clencher
32	Bite
33	Blow
34	Puff
35	Cheek sucked
36	Tongue bulged
37	Lip wiped
38	Nostril dilated
39	Nostril compressor

Table A.6: Miscellaneous.

Appendix B

Detailed Results

We show detailed results of experimentation on both the Cohn Kanade and UNBC-McMaster databases. The threshold (TH) is shown in the first column, and is used to determine the true positive (TP) rate and false positive (FP) rate at that particular threshold. For example, from Table B.1, we see that at a threshold of 0.01, test 1 sees a false positive rate of 63% and a true positive rate of 81%.

	Test 1		Test 2		Test 3		Test 4		Test 5		Average	
TH	FP	TP	FP	TP	FP	TP	FP	TP	FP	TP	FP	TP
0	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
1	63%	81%	50%	100%	75%	100%	50%	78%	82%	87%	65%	89%
5	31%	75%	25%	81%	75%	94%	44%	67%	71%	73%	49%	78%
10	31%	63%	19%	81%	63%	88%	33%	67%	65%	73%	42%	75%
20	31%	56%	19%	81%	63%	88%	28%	61%	47%	67%	37%	72%
30	31%	56%	13%	81%	56%	88%	28%	50%	41%	67%	34%	68%
40	19%	56%	13%	81%	44%	88%	22%	50%	35%	60%	27%	67%
50	19%	50%	6%	81%	44%	88%	22%	50%	35%	60%	25%	65%
60	19%	50%	6%	81%	38%	81%	22%	50%	35%	60%	24%	64%
70	19%	44%	0%	81%	31%	75%	22%	50%	29%	60%	20%	62%
80	19%	44%	0%	75%	31%	75%	11%	39%	29%	60%	18%	58%
90	19%	38%	0%	69%	31%	69%	6%	33%	24%	53%	16%	52%
95	19%	38%	0%	44%	19%	63%	6%	28%	18%	33%	12%	41%
99	13%	19%	0%	44%	6%	25%	0%	22%	12%	33%	6%	28%
100	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%

Table B.1: Results for AU1 - Cohn Kanade database

	Test 1		Test 2		Test 3		Test 4		Test 5		Average	
TH	FP	TP	FP	TP	FP	TP	FP	TP	FP	TP	FP	TP
0	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
1	67%	75%	62%	92%	83%	92%	58%	100%	90%	100%	71%	92%
5	33%	75%	15%	85%	42%	92%	42%	100%	80%	100%	41%	90%
10	25%	67%	0%	85%	33%	92%	42%	100%	80%	100%	34%	88%
20	8%	58%	0%	85%	25%	83%	42%	92%	70%	100%	27%	83%
30	8%	58%	0%	85%	25%	83%	42%	92%	40%	100%	22%	83%
40	8%	58%	0%	77%	25%	83%	42%	92%	40%	100%	22%	81%
50	8%	58%	0%	77%	25%	83%	33%	92%	30%	90%	19%	80%
60	8%	58%	0%	77%	25%	75%	25%	92%	30%	70%	17%	75%
70	8%	33%	0%	77%	25%	75%	25%	83%	30%	70%	17%	68%
80	8%	33%	0%	69%	25%	75%	25%	83%	20%	70%	15%	66%
90	8%	33%	0%	62%	8%	67%	17%	75%	10%	60%	8%	59%
95	8%	33%	0%	54%	8%	58%	0%	58%	10%	60%	5%	53%
99	8%	17%	0%	38%	0%	8%	0%	17%	0%	50%	2%	25%
100	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%

Table B.2: Results for AU2 - Cohn Kanade database

	Test 1		Test 2		Test 3		Test 4		Test 5		Average	
TH	FP	TP	FP	TP	FP	TP	FP	TP	FP	TP	FP	TP
0	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
1	79%	93%	64%	93%	93%	100%	76%	76%	73%	100%	77%	92%
5	50%	93%	57%	93%	79%	79%	71%	76%	60%	80%	64%	84%
10	43%	79%	36%	93%	71%	79%	71%	76%	60%	73%	57%	79%
20	43%	79%	36%	79%	57%	79%	59%	71%	60%	67%	51%	75%
30	36%	71%	14%	79%	43%	79%	53%	71%	47%	53%	39%	71%
40	36%	71%	14%	79%	36%	64%	47%	65%	33%	53%	34%	67%
50	29%	64%	14%	79%	36%	57%	47%	59%	33%	53%	32%	63%
60	29%	64%	14%	79%	36%	50%	35%	47%	33%	53%	30%	59%
70	29%	64%	14%	71%	36%	50%	12%	47%	27%	47%	23%	56%
80	29%	57%	14%	57%	21%	50%	6%	47%	27%	27%	19%	48%
90	29%	50%	7%	43%	21%	43%	6%	41%	27%	20%	18%	40%
95	29%	36%	7%	43%	14%	43%	6%	35%	20%	20%	15%	36%
99	21%	29%	0%	21%	7%	36%	0%	18%	13%	20%	8%	25%
100	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%

Table B.3: Results for AU4 - Cohn Kanade database

	Test 1		Test 2		Test 3		Test 4		Test 5		Average	
TH	FP	TP	FP	TP	FP	TP	FP	TP	FP	TP	FP	TP
0	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
1	88%	100%	88%	100%	38%	100%	100%	100%	67%	100%	76%	100%
5	75%	100%	25%	88%	38%	100%	88%	100%	56%	100%	56%	98%
10	50%	100%	25%	88%	25%	100%	63%	100%	44%	100%	41%	98%
20	25%	100%	25%	88%	13%	100%	63%	88%	44%	100%	34%	95%
30	13%	100%	25%	88%	13%	88%	63%	88%	44%	100%	32%	93%
40	13%	88%	13%	88%	0%	88%	50%	88%	44%	100%	24%	90%
50	13%	88%	13%	75%	0%	88%	38%	88%	44%	100%	22%	88%
60	13%	88%	13%	75%	0%	75%	38%	88%	44%	100%	22%	85%
70	13%	63%	13%	75%	0%	75%	38%	88%	44%	100%	22%	80%
80	13%	50%	13%	63%	0%	75%	25%	88%	22%	78%	15%	71%
90	13%	38%	0%	50%	0%	38%	13%	63%	11%	67%	7%	51%
95	0%	13%	0%	38%	0%	25%	13%	63%	11%	67%	5%	41%
99	0%	13%	0%	13%	0%	13%	0%	0%	0%	22%	0%	12%
100	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%

Table B.4: Results for AU5 - Cohn Kanade database

	Test 1		Test 2		Test 3		Test 4		Test 5		Average	
TH	FP	TP	FP	TP	FP	TP	FP	TP	FP	TP	FP	TP
0	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
1	73%	91%	70%	100%	70%	90%	91%	100%	91%	100%	79%	96%
5	55%	73%	60%	100%	40%	90%	64%	100%	55%	100%	55%	92%
10	36%	64%	60%	100%	30%	70%	55%	100%	45%	100%	45%	87%
20	36%	64%	50%	90%	30%	70%	36%	100%	45%	100%	40%	85%
30	9%	64%	40%	90%	20%	70%	18%	91%	36%	91%	25%	81%
40	9%	64%	40%	80%	20%	70%	9%	91%	36%	91%	23%	79%
50	0%	64%	40%	80%	10%	60%	9%	91%	18%	91%	15%	77%
60	0%	64%	30%	80%	10%	50%	9%	91%	18%	91%	13%	75%
70	0%	45%	30%	80%	0%	50%	9%	73%	9%	91%	9%	68%
80	0%	36%	30%	60%	0%	40%	9%	73%	9%	91%	9%	60%
90	0%	36%	20%	60%	0%	30%	9%	64%	9%	82%	8%	55%
95	0%	27%	20%	50%	0%	30%	9%	55%	0%	73%	6%	47%
99	0%	18%	10%	50%	0%	20%	9%	9%	0%	55%	4%	30%
100	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%

Table B.5: Results for AU6 - Cohn Kanade database

	Test 1		Test 2		Test 3		Test 4		Test 5		Average	
TH	FP	TP	FP	TP	FP	TP	FP	TP	FP	TP	FP	TP
0	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
1	90%	80%	89%	100%	80%	90%	90%	100%	89%	100%	88%	94%
5	80%	80%	56%	89%	50%	90%	70%	90%	89%	89%	69%	88%
10	50%	80%	56%	89%	30%	90%	60%	90%	89%	89%	56%	88%
20	30%	70%	44%	89%	30%	90%	50%	70%	89%	78%	48%	79%
30	20%	60%	33%	89%	30%	80%	30%	70%	89%	78%	40%	75%
40	20%	60%	33%	89%	30%	80%	30%	70%	78%	78%	38%	75%
50	10%	50%	22%	78%	20%	70%	20%	70%	56%	78%	25%	69%
60	10%	40%	22%	67%	20%	70%	10%	60%	56%	78%	23%	63%
70	10%	40%	22%	67%	20%	60%	10%	60%	56%	78%	23%	60%
80	10%	30%	11%	67%	20%	30%	10%	30%	44%	67%	19%	44%
90	10%	20%	11%	67%	10%	10%	0%	30%	44%	56%	15%	35%
95	10%	10%	11%	33%	10%	10%	0%	10%	22%	33%	10%	19%
99	0%	0%	0%	22%	0%	10%	0%	0%	22%	22%	4%	17%
100	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%

Table B.6: Results for AU7 - Cohn Kanade database

	Test 1		Test 2		Test 3		Test 4		Test 5		Average	
TH	FP	TP	FP	TP	FP	TP	FP	TP	FP	TP	FP	TP
0	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
1	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
5	83%	100%	83%	83%	67%	100%	86%	100%	100%	100%	84%	97%
10	50%	100%	50%	83%	67%	100%	86%	100%	83%	100%	68%	97%
20	33%	67%	33%	83%	67%	100%	71%	86%	83%	100%	58%	87%
30	33%	67%	33%	83%	50%	100%	71%	86%	83%	83%	55%	84%
40	33%	67%	17%	67%	33%	83%	57%	86%	83%	83%	45%	77%
50	17%	67%	0%	67%	33%	83%	43%	86%	50%	67%	29%	74%
60	17%	67%	0%	33%	33%	83%	29%	57%	33%	50%	23%	58%
70	17%	67%	0%	17%	33%	83%	0%	57%	17%	50%	13%	55%
80	0%	50%	0%	0%	33%	83%	0%	57%	17%	33%	10%	45%
90	0%	50%	0%	0%	0%	83%	0%	29%	17%	17%	3%	35%
95	0%	17%	0%	0%	0%	50%	0%	14%	17%	0%	3%	16%
99	0%	0%	0%	0%	0%	33%	0%	0%	17%	0%	3%	6%
100	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%

Table B.7: Results for AU9 - Cohn Kanade database

	Test 1		Test 2		Test 3		Test 4		Test 5		Average	
TH	FP	TP	FP	TP	FP	TP	FP	TP	FP	TP	FP	TP
0	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
1	100%	88%	100%	100%	88%	100%	100%	75%	86%	100%	95%	92%
5	88%	75%	75%	88%	75%	75%	88%	75%	71%	86%	79%	79%
10	88%	75%	75%	63%	75%	75%	75%	63%	71%	86%	77%	72%
20	88%	75%	75%	50%	63%	75%	75%	63%	57%	86%	72%	69%
30	75%	75%	63%	50%	63%	75%	63%	50%	57%	71%	64%	64%
40	63%	75%	63%	50%	50%	75%	63%	50%	57%	71%	59%	64%
50	63%	75%	63%	50%	50%	75%	50%	38%	57%	71%	49%	62%
60	63%	75%	50%	50%	50%	63%	38%	38%	43%	71%	49%	59%
70	63%	75%	38%	50%	50%	50%	13%	38%	43%	71%	41%	56%
80	50%	75%	38%	38%	50%	50%	0%	38%	43%	57%	36%	51%
90	50%	63%	25%	38%	25%	13%	0%	25%	29%	14%	26%	31%
95	50%	50%	25%	13%	25%	13%	0%	13%	29%	0%	26%	18%
99	13%	13%	25%	0%	0%	0%	0%	0%	0%	0%	8%	3%
100	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%

Table B.8: Results for AU15 - Cohn Kanade database

	Test 1		Test 2		Test 3		Test 4		Test 5		Average	
TH	FP	TP	FP	TP	FP	TP	FP	TP	FP	TP	FP	TP
0	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
1	80%	100%	100%	93%	43%	57%	63%	75%	73%	73%	72%	81%
5	67%	93%	87%	73%	43%	57%	50%	75%	67%	60%	63%	73%
10	67%	93%	60%	67%	43%	50%	44%	75%	60%	53%	55%	69%
20	60%	87%	53%	60%	36%	50%	44%	69%	53%	53%	49%	65%
30	60%	67%	47%	60%	14%	50%	44%	69%	40%	53%	41%	61%
40	60%	67%	47%	60%	14%	50%	38%	69%	40%	47%	40%	59%
50	53%	67%	27%	60%	14%	50%	38%	63%	40%	47%	30%	58%
60	47%	60%	20%	47%	14%	50%	31%	63%	40%	47%	31%	54%
70	47%	60%	13%	47%	14%	50%	31%	63%	40%	47%	29%	54%
80	40%	60%	7%	47%	14%	50%	31%	63%	40%	40%	27%	53%
90	40%	60%	7%	47%	14%	36%	25%	56%	33%	40%	24%	49%
95	40%	60%	7%	47%	14%	29%	25%	50%	27%	33%	23%	45%
99	27%	33%	7%	7%	7%	21%	25%	38%	27%	20%	19%	24%
100	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%

Table B.9: Results for AU17 - Cohn Kanade database

	Test 1		Test 2		Test 3		Test 4		Test 5		Average	
TH	FP	TP	FP	TP	FP	TP	FP	TP	FP	TP	FP	TP
0	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
1	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
5	80%	80%	100%	100%	100%	100%	100%	100%	80%	100%	92%	96%
10	80%	80%	100%	100%	80%	100%	100%	100%	80%	100%	88%	96%
20	80%	60%	100%	100%	60%	80%	100%	100%	80%	80%	84%	84%
30	60%	40%	80%	100%	40%	80%	80%	80%	80%	60%	68%	72%
40	40%	40%	60%	80%	40%	80%	60%	80%	40%	40%	48%	64%
50	40%	20%	40%	80%	40%	60%	60%	80%	40%	40%	44%	56%
60	40%	20%	40%	40%	40%	60%	60%	80%	40%	20%	44%	44%
70	40%	20%	40%	40%	40%	40%	60%	80%	40%	20%	44%	40%
80	40%	20%	40%	40%	20%	40%	40%	80%	40%	20%	36%	40%
90	20%	20%	20%	40%	20%	40%	40%	40%	40%	0%	28%	28%
95	20%	0%	20%	40%	20%	20%	20%	20%	40%	0%	24%	16%
99	0%	0%	20%	0%	20%	0%	0%	20%	14%	0%	16%	4%
100	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%

Table B.10: Results for AU20 - Cohn Kanade database

	Test 1		Test 2		Test 3		Test 4		Test 5		Average	
TH	FP	TP	FP	TP	FP	TP	FP	TP	FP	TP	FP	TP
0	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
1	35%	70%	95%	100%	71%	71%	74%	74%	65%	90%	68%	81%
5	30%	65%	95%	95%	67%	67%	57%	70%	65%	85%	63%	76%
10	25%	60%	95%	95%	62%	67%	52%	65%	60%	80%	59%	73%
20	20%	55%	90%	90%	62%	67%	52%	57%	55%	80%	56%	69%
30	20%	50%	90%	90%	57%	67%	43%	57%	40%	80%	50%	68%
40	20%	50%	90%	90%	57%	67%	39%	57%	40%	75%	49%	67%
50	15%	50%	85%	90%	52%	67%	39%	57%	40%	75%	45%	68%
60	15%	50%	85%	80%	43%	62%	39%	57%	40%	75%	44%	65%
70	15%	50%	85%	80%	43%	62%	39%	57%	40%	75%	44%	65%
80	15%	45%	80%	75%	43%	52%	39%	57%	40%	70%	43%	60%
90	15%	45%	75%	75%	38%	52%	39%	57%	35%	65%	39%	59%
95	15%	35%	55%	70%	33%	52%	35%	57%	30%	55%	34%	54%
99	5%	25%	20%	25%	29%	33%	17%	48%	25%	40%	19%	31%
100	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%

Table B.11: Results for AU25 - Cohn Kanade database

	Test 1		Test 2		Test 3		Test 4		Test 5		Average	
TH	FP	TP	FP	TP	FP	TP	FP	TP	FP	TP	FP	TP
0	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
1	80%	100%	70%	100%	80%	90%	80%	100%	91%	100%	80%	98%
5	50%	100%	10%	100%	40%	90%	60%	100%	73%	91%	47%	96%
10	40%	100%	10%	100%	30%	80%	50%	100%	36%	91%	33%	94%
20	30%	100%	10%	100%	20%	80%	50%	100%	27%	91%	27%	94%
30	20%	100%	10%	100%	20%	80%	40%	100%	27%	82%	24%	92%
40	20%	90%	10%	100%	20%	80%	30%	90%	18%	82%	20%	88%
50	20%	90%	0%	100%	10%	70%	30%	90%	18%	82%	16%	86%
60	20%	90%	0%	100%	10%	70%	30%	90%	0%	82%	12%	86%
70	20%	90%	0%	90%	10%	60%	30%	90%	0%	82%	12%	82%
80	10%	70%	0%	80%	10%	60%	30%	90%	0%	82%	10%	76%
90	10%	60%	0%	70%	0%	40%	20%	80%	0%	82%	6%	67%
95	0%	40%	0%	60%	0%	40%	10%	60%	0%	55%	2%	43%
99	0%	20%	0%	40%	0%	0%	0%	30%	0%	27%	0%	24%
100	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%

Table B.12: Results for AU27 - Cohn Kanade database

Threshold	False positives	True positives
0	100%	100%
1	67%	100%
5	53%	100%
10	45%	100%
20	41%	100%
30	35%	92%
40	33%	92%
50	31%	92%
60	31%	92%
70	27%	92%
80	27%	92%
90	22%	75%
95	20%	75%
99	16%	50%
100	0%	0%

Table B.13: Results for AU4 - UNBC McMaster database

Threshold	False positives	True positives
0	100%	100%
1	81%	94%
5	71%	84%
10	65%	76%
20	56%	72%
30	52%	70%
40	48%	66%
50	35%	64%
60	34%	54%
70	33%	48%
80	27%	44%
90	19%	40%
95	12%	26%
99	8%	18%
100	0%	0%

Table B.14: Results for AU4 - UNBC McMaster database

Threshold	False positives	True positives
0	100%	100%
1	82%	91%
5	66%	77%
10	59%	77%
20	48%	59%
30	47%	59%
40	45%	55%
50	40%	55%
60	37%	36%
70	33%	32%
80	29%	32%
90	26%	32%
95	23%	23%
99	11%	23%
100	0%	0%

Table B.15: Results for AU7 - UNBC McMaster database

Threshold	False positives	True positives
0	100%	100%
1	86%	93%
5	63%	79%
10	56%	79%
20	41%	71%
30	36%	64%
40	33%	64%
50	29%	64%
60	29%	50%
70	21%	36%
80	16%	29%
90	10%	14%
95	9%	7%
99	4%	0%
100	0%	0%

Table B.16: Results for AU9 - UNBC McMaster database