

# Recognition of Facial Action Units from Video Streams with Recurrent Neural Networks : A New Paradigm for Facial Expression Recognition

Hima Bindu Vadapalli

A thesis submitted in fulfillment of the requirements for the degree  
of

*Doctor of Philosophiae*

in the Department of Computer Science,

University of the Western Cape.

July 2011 .

Supervisor: Prof. C.W. Omlin

Co-Supervisor: Prof. H.O. Nyongesa

# Abstract

*Recognition of Facial Action Units from Video Streams with Recurrent Neural Networks: A New Paradigm for Facial Expression Recognition*

Hima Bindu Vadapalli

PhD Thesis, Department of Computer Science, University of the Western Cape

This research investigated the application of recurrent neural networks (RNNs) for recognition of facial expressions based on facial action coding system (FACS). Support vector machines (SVMs) were used to evaluate the results obtained by RNNs. In this approach, instead of recognizing whole facial expressions, the focus is on the recognition of action units (AUs) that are defined in FACS. Recurrent neural networks are capable of gaining knowledge from temporal data while SVMs, which are time invariant, are known to be very good classifiers of such features. Thus, the research consists of four important components; these are, the comparison of use of image sequences against single static images, investigation of benchmarking feature selection and network optimization approaches, the study of inter-AU correlations by implementing multiple output RNNs, and the study of difference images and advantages of their application.

In the comparative studies, image sequences were classified using a combination of Gabor filters and RNNs, while single static images were classified using Gabor and SVMs. Set of 11 FACS AUs were classified by both approaches, where a single RNN/SVM classifier is

used for classifying each AU. Results indicated that classifying FACS AUs using image sequences yielded better results than using static images. The average recognition and false alarm rates using image sequences was 82.75% and 7.61%, respectively, while the classification using single static images yielded recognition and a false alarm rate of 79.47% and 9.22%, respectively. The better performance by the use of image sequences can be attributed to RNNs ability, as stated above, to extract knowledge from time-series data.

Subsequent research then investigated benchmarking dimensionality reduction, feature selection and network optimization techniques, in order to improve the performance provided by the use of image sequences. Results showed that an optimized network, using weight decay, gave best recognition and false alarm rate of 85.38% and 6.24%, respectively. The next study was of the inter-AU correlations existing in the Cohn-Kanade database and their effect on classification models. To accomplish this, a model was developed for the classification of a set of AUs by a single multiple output RNN. Results indicated that the high inter-AU correlations do in fact aid classification models to gain more knowledge and, thus, perform better. However, this was limited to AUs that start and reach apex at almost the same time. This suggests the need for availability of a larger database of AUs, which could provide both individual and AU combinations for further investigation.

The final part of this research investigated the use of difference images to track the motion of the image pixels. Difference images provide both noise and feature reduction, an aspect that was studied. Results showed that that the use of difference image sequences provided the best results achieved, with recognition and false alarm rates of 87.95% and 3.45%, respectively, which is shown to be significant when compared to the use of normal image sequences classified using RNNs. In conclusion, the research demonstrates that the use of RNNs

for classification of image sequences is a new and improved paradigm for facial expression recognition.

## **Keywords**

Facial Expression Recognition

Facial Action Coding System

Action Unit Recognition

Feature Extraction

Gabor Filters

Principle Component Analysis

Support Vector Machines

Spatio-Temporal Modeling

Recurrent Neural Networks

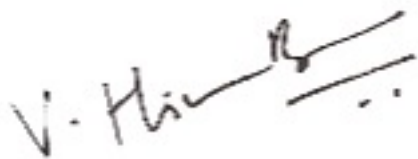
Difference Images

## Declaration

I declare that *Recognition of Facial Action Units from Video Streams with Recurrent Neural Networks: A New Paradigm for Facial Expression Recognition* is my own work, that it has not been submitted before for any degree or examination in any other university, and that all the sources I have used or quoted have been indicated and acknowledged as complete references.

Hima Bindu Vadapalli

July, 2011

A handwritten signature in black ink, appearing to read 'V. Hima Bindu', with a stylized flourish at the end.

Signed: .....

## **Acknowledgements**

I would like to express my sincere gratitude to Prof. C.W. Omlin, for his valuable guidance throughout the entire research work. His assistance in putting this thesis together is much appreciated. Thanks are due to Prof. H Nyongesa for co-supervising my thesis work. I would also like to thank Prof. Ian Sanders, University of Witwatersrand for reviewing the thesis draft.

I would also like to thank my family and parents for their support to carry out my Ph.D work.

# Contents

<b>List of Figures</b>	<b>iii</b>
<b>List of Tables</b>	<b>iv</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation . . . . .	2
1.2 Research Questions . . . . .	4
1.3 Technical Objectives . . . . .	4
1.4 Accomplishments . . . . .	5
1.5 Methodology . . . . .	5
1.6 Thesis Overview . . . . .	6
<b>2 Why Facial Expression Recognition?</b>	<b>8</b>
2.1 Introduction . . . . .	8
2.2 Sign Language . . . . .	8
2.3 Psychological Studies . . . . .	10
2.4 Entertainment . . . . .	10
2.5 Security . . . . .	11
2.6 Bi-modal Emotion Recognition . . . . .	12
2.7 Surveillance/ Monitoring . . . . .	12
2.8 Automated Tutoring . . . . .	12
2.9 Summary . . . . .	13
<b>3 Facial Action Coding System</b>	<b>14</b>
3.1 Introduction . . . . .	14
3.2 Design of FACS . . . . .	17



3.2.1	Action Units and Their Muscular Action . . . . .	17
3.2.2	AU Combinations . . . . .	20
3.2.3	AU Intensity and Scoring . . . . .	21
3.2.4	Reliability of FACS . . . . .	22
3.3	Application of FACS in Computer Vision . . . . .	23
3.4	Suitability of FACS for Measuring Facial Expressions . . . . .	24
3.5	Alternative systems for Measuring Facial Expressions . . . . .	24
3.6	Summary . . . . .	26
<b>4</b>	<b>Literature Review</b>	<b>27</b>
4.1	Introduction . . . . .	27
4.2	Facial Expression Analysis . . . . .	27
4.2.1	Input Data . . . . .	28
4.2.2	Face Retrieval . . . . .	28
4.2.3	Preprocessing . . . . .	29
4.2.4	Feature Extraction . . . . .	29
4.2.5	Expression Classification . . . . .	29
4.3	Different Approaches . . . . .	30
4.4	Geometry Based Methods . . . . .	31
4.4.1	Relative Distances . . . . .	31
4.4.2	Estimation of Face Model Parameters . . . . .	35
4.4.3	3D Model Fitting . . . . .	36
4.5	Appearance Based Methods . . . . .	38
4.5.1	Unsupervised Methods for Dimensionality Reduction . . . . .	38
4.5.2	Supervised Methods for Dimensionality Reduction . . . . .	40
4.5.3	Gabor Filters . . . . .	40
4.5.4	Gabor Responses Over Entire Face Region . . . . .	42
4.5.5	Gabor Responses at Fiducial Points . . . . .	43
4.5.6	Gabor Responses at Learned Locations . . . . .	44
4.5.7	Configuration of Gabor Jets . . . . .	45
4.5.8	Importance of Scales . . . . .	45
4.5.9	Feature Selection Techniques for Gabor Coefficients . . . . .	47
4.6	Comparison of Appearance Based and Geometry Based Methods . . . . .	48

4.7	Hybrid Systems . . . . .	49
4.8	Image Sequences with Increasing AU Complexity . . . . .	50
4.9	Classification . . . . .	51
4.9.1	Spatial Approach . . . . .	51
4.9.2	Spatio-temporal Approach . . . . .	51
4.10	FACS AU Recognition Using RNNs . . . . .	56
4.11	Conclusion . . . . .	56
4.12	Summary . . . . .	57
<b>5</b>	<b>Feature Extraction Techniques</b>	<b>58</b>
5.1	Introduction . . . . .	58
5.2	Gabor Filters . . . . .	58
5.3	Principle Component Analysis . . . . .	61
5.4	Linear Discriminant Analysis . . . . .	61
5.5	AdaBoost . . . . .	62
5.6	Haar Filters . . . . .	63
5.7	Summary . . . . .	64
<b>6</b>	<b>Modeling</b>	<b>65</b>
6.1	Introduction . . . . .	65
6.2	Artificial Neural Networks . . . . .	66
6.2.1	Processing Unit . . . . .	66
6.2.2	Multilayer Perceptron . . . . .	68
6.2.3	Learning Algorithm . . . . .	71
6.2.4	Theoretical Properties . . . . .	72
6.3	Support Vector Machines . . . . .	73
6.3.1	Basic Principle . . . . .	74
6.3.2	Kernel Trick . . . . .	78
6.3.3	Handling Linear Inseparability . . . . .	79
6.4	Recurrent Neural Networks . . . . .	81
6.4.1	First-Order Recurrent Neural Networks . . . . .	81
6.4.2	Second Order Recurrent Neural networks . . . . .	82
6.4.3	Back Propagation Through Time Learning Algorithm . . .	83
6.4.4	Real Time Recurrent Learning Learning Algorithm . . . .	87

6.4.5	Applications . . . . .	89
6.4.6	Computation Capability of RNNs . . . . .	91
6.4.7	Vanishing Gradient Problem . . . . .	91
6.4.8	Network Optimization using Weight Decay . . . . .	92
6.4.9	Network Optimization using Weight Elimination . . . . .	93
6.5	Hidden Markov Models . . . . .	94
6.5.1	Definition . . . . .	94
6.5.2	Assumptions in HMMs . . . . .	96
6.5.3	Basic Problems in HMMs . . . . .	97
6.5.4	Maximum Likelihood (ML) Criterion . . . . .	97
6.6	LSTM Recurrent Neural Networks . . . . .	98
6.6.1	Architecture of LSTM . . . . .	98
6.7	Learning Algorithm . . . . .	99
6.7.1	Applications . . . . .	102
6.8	Summary . . . . .	103
<b>7</b>	<b>Recognition Using SVMs</b>	<b>105</b>
7.1	Cohn-Kanade Database . . . . .	105
7.1.1	Image Sequences vs Single Static Images . . . . .	106
7.1.2	11 AU and Their Description . . . . .	107
7.2	FACS AU Recognition using SVMs . . . . .	108
7.2.1	Data Collection . . . . .	110
7.2.2	Preprocessing . . . . .	110
7.2.3	Feature Extraction . . . . .	111
7.2.4	Classification . . . . .	111
7.2.5	Results . . . . .	112
7.2.6	Discussion . . . . .	112
7.3	Summary . . . . .	113
<b>8</b>	<b>Recognition Using RNNs</b>	<b>114</b>
8.1	Introduction . . . . .	114
8.1.1	Image Sequences Depicting AUs . . . . .	114
8.2	Baseline FACS AU Recognition Using RNN . . . . .	115
8.2.1	Data Collection . . . . .	116

8.2.2	Preprocessing . . . . .	117
8.2.3	Feature Extraction . . . . .	117
8.2.4	Classification . . . . .	118
8.2.5	Results . . . . .	119
8.2.6	Discussion . . . . .	120
8.3	Effect of Number of Hidden Nodes . . . . .	121
8.3.1	Results . . . . .	123
8.3.2	Discussion . . . . .	123
8.4	Feature Selection . . . . .	124
8.4.1	Frequency Selection . . . . .	124
8.4.2	Local Gabor Filters . . . . .	126
8.4.3	Feature Selection using PCA . . . . .	130
8.5	RNN Network Optimization . . . . .	132
8.5.1	Weight Decay . . . . .	132
8.5.2	Discussion . . . . .	133
8.6	Recognition of other FACS AUs . . . . .	134
8.6.1	AUs and Their Description . . . . .	135
8.6.2	Experimentation and Results . . . . .	135
8.6.3	Discussion . . . . .	136
8.7	Single Static Images vs Image Sequences . . . . .	136
8.7.1	Advantages of using Image Sequences . . . . .	137
8.7.2	Disadvantages of using Image Sequences . . . . .	137
8.8	Conclusions . . . . .	137
8.9	Summary . . . . .	138
<b>9</b>	<b>FACS AU Recognition Using Multiple Output RNNs</b>	<b>139</b>
9.1	Multiple Output RNNs . . . . .	139
9.2	AU Combinations . . . . .	140
9.2.1	Additive and Non-Additive AUs . . . . .	141
9.2.2	Inter AU Correlation in Cohn-Kanade Database . . . . .	142
9.2.3	AU Combinations . . . . .	143
9.3	Data Collection . . . . .	144
9.4	Experiments and Results . . . . .	145

## CONTENTS

---

9.5	Discussions . . . . .	146
9.6	Summary . . . . .	146
<b>10</b>	<b>Recognition with Difference Images</b>	<b>147</b>
10.1	Normal vs. Difference Images . . . . .	148
10.1.1	Feature Reduction . . . . .	148
10.1.2	Noise Reduction . . . . .	148
10.2	Data Preprocessing . . . . .	149
10.3	Feature Extraction . . . . .	149
10.4	Classification . . . . .	150
10.5	Results and Discussion . . . . .	150
10.6	Network Optimization . . . . .	152
10.6.1	Weight Decay . . . . .	152
10.6.2	Weight Elimination . . . . .	152
10.7	Comparison: Normal vs Difference Images . . . . .	153
<b>11</b>	<b>Conclusions and Directions for Future Work</b>	<b>155</b>
	<b>References</b>	<b>173</b>

# List of Figures

1.1	Overview of our recognition model . . . . .	6
2.1	Example based cloning model . . . . .	11
3.1	Different facial areas and their names . . . . .	15
3.2	FACS AUs and their description . . . . .	18
3.3	Muscle Actions (indicated by numbers) and Directions for Upper Face FACS AUs . . . . .	19
3.4	Scale of Evidence and Intensity Scores Relation . . . . .	22
4.1	Overview of an general FER system . . . . .	28
4.2	Overview of an general FER system . . . . .	30
4.3	System Structure . . . . .	32
4.4	Manual tracking of fiducial points in neural frame and automatic tracking in the subsequent frames . . . . .	32
4.5	Geometric face model used to produce the feature vectors . . . . .	33
4.6	SVM based automated FER system using feature displacements .	33
4.7	System architecture used by Kotsia and Pitas for FER . . . . .	34
4.8	Feature based system used in Tian et al. . . . .	35
4.9	Feature tracking to fit a 3D model on to a face . . . . .	36
4.10	3D model fitting used in Braathen et al. . . . .	37
4.11	Candide face model used in Graves et al. 2008 . . . . .	37
4.12	Architecture of radial basis function network . . . . .	39
4.13	The real and imaginary components of the Gabor jet with five spatial frequencies spaced at half-octaves and eight different orien- tations . . . . .	41

## LIST OF FIGURES

---

4.14	Gabor wavelet representation of a facial image . . . . .	43
4.15	The Gabor, Adaboost and AdaSVM used in Littlewort et al. . . . .	44
4.16	Examples of local Gabor filters proposed by Deng et al. . . . .	46
4.17	System architecture used in Zhang et al. . . . .	49
4.18	20 fiducial points in the upper half of the face region and the neural network architecture used in Tian et al. . . . .	50
4.19	Block Diagram of the recognition system used in Lien 1998 . . . . .	52
4.20	Block Diagram of the HMM based FER system used in Petar and Aggelos 2006 . . . . .	53
4.21	HMM structure capable of modeling whole image sequence used in Muller et al. . . . .	53
4.22	Discrete-time recurrent neural network used in Kobayashi and Hara . . . . .	54
4.23	Hierarchy of networks based on decomposition of emotions used in Rosenblum et al. . . . .	55
4.24	RNN structure used in Hai Tao et al. . . . .	55
4.25	Uni-directional and bi-directional networks used in Graves et al. . . . .	56
5.1	The real and imaginary parts of a complex sinusoid with $u_0 =$ $v_0 = 1/80$ cycles/pixel and $P = 0$ deg. as shown in (87) . . . . .	59
6.1	Structure of a single neuron . . . . .	67
6.2	Network Topology of a Feed-forward Network . . . . .	69
6.3	Maximum margin hyperplane H that separates the hypothetical training data points . . . . .	75
6.4	Architecture of First Order Recurrent Neural Network . . . . .	81
6.5	Architecture of Second Order Recurrent Neural Network . . . . .	83
6.6	Recurrent Neural Network a) Elman Network b) Elman Network “unfolded” in time for 2 time steps . . . . .	84
6.7	LSTM Architecture with Memory Cell . . . . .	99
7.1	Image sequences with the first and last two frames from the video clip . . . . .	107
7.2	Image sequences for AU 1+2, 4 and 6 from the Cohn-Kanade database . . . . .	108

## LIST OF FIGURES

---

7.3	11 FACS AUs and their description . . . . .	109
7.4	Cropped lower face regions . . . . .	111
8.1	Image Sequences Depicting the Upper Face AUs Classified . . . .	115
8.2	Image Sequences Depicting the Lower Face AUs Classified . . . .	116
8.3	Overview of the FACS AU classification Model . . . . .	117
8.4	Elman Recurrent Neural Network for the Recognition of FACS AUs	118
8.5	Elman Recurrent Neural Network Unfolded in Time . . . . .	119
8.6	System error vs. Epochs. . . . .	122
8.7	Effect of number of hidden units on the performance of the model for the 11 FACS AUs . . . . .	123
8.8	Significance of selection of spatial frequencies for both upper and lower face AUs . . . . .	125
8.9	Real and imaginary parts of a 5*8 Gabor jet . . . . .	127
8.10	Local Gabor filters a) LG1 (5x8) b) LG2 (5x8) and c) LG3 (5x8) .	128
8.11	Other AUs in the FACS system and their description (Note: Blink cannot be depicted using a single frame.) . . . . .	135
9.1	Structure of a Multiple Output RNN . . . . .	140
10.1	Original and the difference image sequences . . . . .	150
10.2	Filtered 5*8 coefficients of the last difference image . . . . .	150
11.1	Baseline recognition (RR %) and false alarm rate (FAR %) for all the upper face AUs obtained during fold 1 using RNN classifier .	162
11.2	Baseline recognition (RR %) and false alarm rate (FAR %) for all the upper face AUs obtained during fold 2 using RNN classifier .	162
11.3	Baseline recognition (RR %) and false alarm rate (FAR %) for all the upper face AUs obtained during fold 3 using RNN classifier .	163
11.4	Baseline recognition (RR %) and false alarm rate (FAR %) for all the lower face AUs obtained during fold 1 using RNN classifier . .	163
11.5	Baseline recognition (RR %) and false alarm rate (FAR %) for all the lower face AUs obtained during fold 2 using RNN classifier . .	164
11.6	Baseline recognition (RR %) and false alarm rate (FAR %) for all the lower face AUs obtained during fold 3 using RNN classifier . .	164



## LIST OF FIGURES

---

11.7	Recognition (RR %) and false alarm rate (FAR %) for all the upper face AUs obtained during fold 1 using frequency scale selection and RNN classifier . . . . .	165
11.8	Recognition (RR %) and false alarm rate (FAR %) for all the upper face AUs obtained during fold 2 using frequency scale selection and RNN classifier . . . . .	165
11.9	Recognition (RR %) and false alarm rate (FAR %) for all the upper face AUs obtained during fold 3 using frequency scale selection and RNN classifier . . . . .	166
11.10	Recognition (RR %) and false alarm rate (FAR %) for all the upper face AUs obtained during fold 1 using optimized(baseline+weight decay with a decay constant of 0.0001) RNN classifier . . . . .	166
11.11	Recognition (RR %) and false alarm rate (FAR %) for all the upper face AUs obtained during fold 2 using optimized(baseline+weight decay with a decay constant of 0.0001) RNN classifier . . . . .	167
11.12	Recognition (RR %) and false alarm rate (FAR %) for all the upper face AUs obtained during fold 3 using optimized(baseline+weight decay with a decay constant of 0.0001) RNN classifier . . . . .	167
11.13	Recognition (RR %) and false alarm rate (FAR %) for all the lower face AUs obtained during fold 1 using optimized(baseline+weight decay with a decay constant of 0.0001) RNN classifier . . . . .	168
11.14	Recognition (RR %) and false alarm rate (FAR %) for all the lower face AUs obtained during fold 2 using optimized(baseline+weight decay with a decay constant of 0.0001) RNN classifier . . . . .	168
11.15	Recognition (RR %) and false alarm rate (FAR %) for all the lower face AUs obtained during fold 3 using optimized(baseline+weight decay with a decay constant of 0.0001) RNN classifier . . . . .	169
11.16	Recognition (RR %) and false alarm rate (FAR %) for all the upper face AUs obtained during fold 1 using difference images and RNN classifier . . . . .	169
11.17	Recognition (RR %) and false alarm rate (FAR %) for all the upper face AUs obtained during fold 2 using difference images and RNN classifier . . . . .	170

## LIST OF FIGURES

---

11.18	Recognition (RR %) and false alarm rate (FAR %) for all the upper face AUs obtained during fold 3 using difference images and RNN classifier . . . . .	171
11.19	Recognition (RR %) and false alarm rate (FAR %) for all the lower face AUs obtained during fold 1 using difference images and RNN classifier . . . . .	171
11.20	Recognition (RR %) and false alarm rate (FAR %) for all the lower face AUs obtained during fold 2 using difference images and RNN classifier . . . . .	172
11.21	Recognition (RR %) and false alarm rate (FAR %) for all the lower face AUs obtained during fold 3 using difference images and RNN classifier . . . . .	172

# List of Tables

3.1	Description of facial areas . . . . .	16
7.1	Number of samples for train and test sets used for upper face AUs	106
7.2	Number of samples for train and test sets used for lower face AUs	106
7.3	Recognition results for the six AUs detected in the upper half of the face using SVM as a classifier . . . . .	112
7.4	Recognition results for the five AUs detected in the lower half of the face using SVM as a classifier . . . . .	113
8.1	Recognition results for the six AUs detected in the upper face using RNN as a classifier . . . . .	120
8.2	Recognition results for the five AUs detected in the lower face using RNN as a classifier . . . . .	120
8.3	Recognition rates using higher, middle and lower set of frequencies for the upper and lower face AUs . . . . .	124
8.4	Recognition and False Alarm Rate for the upper face AUs with the first three higher frequencies . . . . .	126
8.5	Recognition and False Alarm Rate for the lower face AUs with the first three higher frequencies . . . . .	126
8.6	Memory requirements of global and local Gabor filters and the number of hidden units used by RNN based FACS AU recognition model . . . . .	129
8.7	Recognition Rate and False Alarm Rate for the three variants of local Gabor filters for the six upper face AUs . . . . .	129

## LIST OF TABLES

---

8.8	Recognition and False Alarm Rate for the six upper face AUs with LG3 (5*8) variant of local Gabor filters . . . . .	130
8.9	Recognition Rate and False Alarm Rate for the three variants of the Local Gabor filters for the five Lower Face AUs using RNN . .	130
8.10	Recognition and False Alarm Rate for the five Lower Face AU with LG3 (5*8) of local Gabor filters using RNN . . . . .	131
8.11	Performance with and without using PCA . . . . .	132
8.12	Average recognition and false alarm rates for the six upper face AUs using three different decay constants . . . . .	133
8.13	Recognition and false alarm rate for the six upper face AUs with a decay constant of 0.0001 . . . . .	133
8.14	Average recognition and false alarm rates for the five lower face AUs for three different decay constants . . . . .	134
8.15	Recognition and false alarm rate for the five lower face AUs with a decay constant of 0.0001 . . . . .	134
8.16	Classification performance for other FACS AUs . . . . .	136
9.1	Problematic AU Combinations present in Cohn-Kanade database	141
9.2	AU correlation in our data subset for the upper face AUs . . . . .	143
9.3	AU correlation in our data subset for the lower face AUs . . . . .	143
9.4	AU combinations considered for classification . . . . .	144
9.5	Recognition and False alarm rate for six upper face AUs . . . . .	145
9.6	Recognition and False alarm rate for five lower face AUs . . . . .	145
10.1	Recognition and False alarm rate for six upper face AUs using difference images and RNN . . . . .	151
10.2	Recognition and False alarm rate for five lower face AUs using difference images and RNN . . . . .	151
10.3	FACS AU recognition with weight elimination . . . . .	153

## Glossary

<b>HCI</b>	Human Computer Interaction
<b>FER</b>	Facial Expression Recognition
<b>FACS</b>	Facial Action Coding System
<b>AU</b>	Action Unit
<b>SVM</b>	Support Vector Machine
<b>RNN</b>	Recurrent Neural Network
<b>PCA</b>	Principle Component Analysis
<b>LDA</b>	Linear Discriminant Analysis
<b>IDA</b>	Indepedent Component Analysis
<b>SASL</b>	South African Sign Language
<b>HMM</b>	Hidden Markov Model
<b>LSTM</b>	Long Short Term Memory
<b>BPTT</b>	Back Propagation Through Time
<b>RTRL</b>	Real Time Recurrent Learning

## Articles

### Articles published so far:

- H. Vadapalli, H. Nyongesa, C.W. Omlin. Facial Action Unit Recognition using Recurrent Neural Networks. In Proceedings of the 2009 International Conference on Image Processing, Computer Vision, Pattern Recognition, IPCV 2009, July 13-16, 2009, Las Vegas, Nevada, USA, Volume 2, 2009.
- H. Vadapalli, H. Nyongesa, C.W. Omlin. Recurrent Neural Networks for Facial Action Unit Recognition from Image Sequences. In the Proceedings of the Twenty-first Annual Symposium of the Pattern Recognition Association of South Africa (PRASA2010), South Africa, pp. 269-273, November 2010.
- H. Vadapalli, H. Nyongesa, C.W. Omlin. Classifying Facial Action Units: Use of Time Variant Data and Recurrent Neural Networks. In The 11th International Conference on Pattern Recognition and Information Processing (PRIP 2011), Minsk, Belarus, May, 2011.

### Articles under preparation:

- H. Vadapalli, H. Nyongesa, C.W. Omlin. Are Recurrent Neural Networks Better than SVMs? An Approach to Facial Action Unit Recognition.
- H. Vadapalli, H. Nyongesa, C.W. Omlin. Effect of Inter-Action Unit Correlations and Multi Output Recurrent Neural Networks.

# Chapter 1

## Introduction

Computer vision has been a major area of research for the past three decades. Among many studies carried out, they were studies on analysis of human faces. Human facial expressions are a means for non-verbal communication between individuals, and thus, their study is of interest in understanding human natural languages. Analysis of the basic expressions helps develop appropriate computer vision tools. Analysis of facial expressions also has major applications in human-computer interaction (HCI) and in physiological studies. Existing technologies in HCI are application specific and require the understanding of machine specific operations by the humans, which involves a lot of resources in terms of time and manpower. Enabling machines to adapt to human behavior would save resources. To make interaction between humans and computers natural, machines should derive the required information from human cues such as speech, gestures and facial expressions. Recent HCI technologies are based on verbal communications; however, the use of nonverbal communication is gaining momentum and is usually referred to as paralinguistic, because they substitute the verbal communication. Based on this paradigm, facial expressions can be used to provide supplementary information for models based on speech. The combination of both speech and facial expressions provides better understanding of the information being conveyed. Facial expressions not only provide information regarding the physiological state of the person but also provide a platform for better understanding their behavior when combined with other human communication modes. Thus, simply expressions such as, “happy” can be used to understand the state of a human subject.

In physiological studies, facial expression information not only provides the state of the subject, but also his personality, temperament and verity.

Over the years, methods have been developed to recognize the prototypic facial expressions, such as, angry, sad, happy, etc. These prototypical expressions often commonly correspond to the emotions expressed by humans. However, in real world applications prototypical expressions occur relatively infrequently. Emotion is rather more often communicated by subtle changes in facial features, for example obliquely lowering the lips in sadness. The subtle changes near the brows, eyelids, nose and mouth regions provide a full range of facial displays that can possibly be generated by any human subject.

Ekman and Friesen (43) developed the Facial Action Coding System (FACS) which anatomically relates the contraction of specific facial muscles to a set of units called action units (AUs). These AUs describe the facial movements at their base level. Facial action coding system provides a ground truth that can be used to describe the facial expressions. In itself, the FACS uses no emotion specific labels and is purely descriptive about the muscles and the facial movements generated by their contractions. The original FACS (43) contained 44 AUs concentrating on the muscle movements around eyes, nose and mouth. These AUs singly, or as a set describe the different facial displays. The updated version (45) has an additional 12 AUs describing the movements of eyeballs and head. Manual coding of all facial displays by humans is possible by viewing videotapes depicting these facial displays and applying FACS rules to them. However, this form of coding and recognizing is labor-intensive and to some degree dependent on the coder. Standardization of the system using machine specific tools is highly recommended for their effective modeling.

## 1.1 Motivation

Modeling of FACS AUs and their recognition requires tools that can perform well under real time constraints. This suggested the use of machine learning tools such as neural networks and support vector machines (SVMs). Significant work has been done in both facial expression and FACS AU recognition using these machine learning tools, with each one of the tools having their own advantages



and disadvantages. On one hand, SVMs have been widely used for expression recognition because they are easy to train with huge amounts of data. On the other hand, neural networks require extensive training, but have good generalization capability. Most of the work in the field of FACS AU recognition using the above approaches have concentrated on the use of single static images or image sequences with no time variant components present in the data. This is mainly based on the fact that these classifiers lack the ability to handle time variant data. The work by Bartlett et al. (11) used single static images to classify the seven basic expressions using Gabor filters and SVMs. The study reported promising results. The work by Tian et al. (112) was based on the use of neural networks using geometric features and Gabor filters for recognizing a set of nine FACS AUs. None of these studies used time variant data.

Studies in fields such as speech recognition and financial forecasting have demonstrated the advantage of using time variant data towards understanding and extracting important knowledge from the structure of the speech fluctuations. Classifiers that have been widely used in both speech and financial forecasting for handling of time variant data are recurrent neural networks (RNNs) (69), (75). This motivated the design of a real time FACS AU recognition model in this thesis. To our knowledge, no previous research has investigated this approach. In contrast to the use of single static images, the proposed approach uses sequences of images that depict presence or absence of AUs. The image sequences provide information about the formation of different FACS AUs rather than the AUs themselves.

Automatic recognition of FACS AUs, in general, is divided into three major steps: (a) Image pre-processing including face detection and normalization, (b) Feature extraction, and (3) FACS AU classification. In this research, the focus is only on feature extraction and FACS AU classification. Image normalization, in particular, is avoided to maintain the basic head movements in image sequences, which provides time variant components in data. Basic face detection and normalization are still carried out with the use of eye coordinates.

## **1.2 Research Questions**

The central research question of this thesis is whether temporal modeling of FACS AUs from video streams can improve the recognition performance as compared to recognition from static images. FACS AU recognition from video streams thus gives rise to the following investigative research questions:

- Which salient features facilitate robust FACS AU recognition;
- Whether RNNs are capable of modeling time variant facial AUs from video sequences;
- Whether RNNs are better modeling tool compared to SVMs.

Research method and technique issues studied include:

- The cross-correlation between AUs and their impact on recognition performance;
- The use of difference images as a viable alternative to more elaborate feature extraction and dimensionality reduction techniques.

## **1.3 Technical Objectives**

The following technical objectives need to be addressed in order to answer the research questions:

- Representation of feature extraction methods;
- Training of SVMs for AU recognition from static images;
- Training of RNNs for AU recognition from video streams;
- Comparison of recognition performances;
- Training of multiple output RNN for simultaneous recognition of multiple AUs;
- Investigation of difference images as features and comparison with other feature dimensionality reduction techniques.

## 1.4 Accomplishments

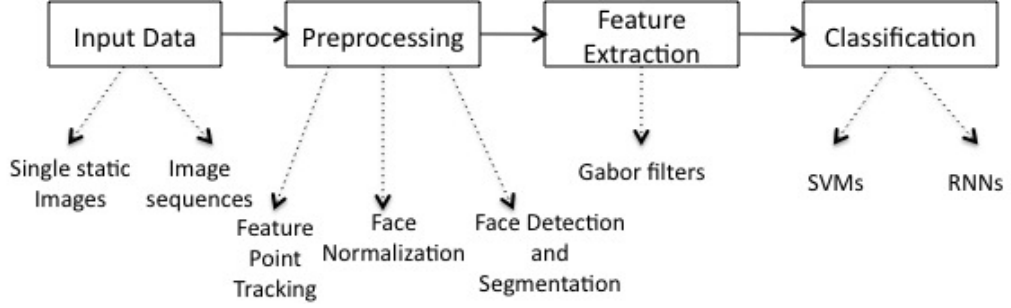
This work accomplished the following

- Designed and implemented an automatic FACS AU recognition model using Gabor filters and RNNs capable of handling video streams;
- Established that RNNs can perform better than SVMs for FACS AU recognition;
- Established that cross-correlation between AUs can be used to provide better performance under certain conditions;
- Established that difference image sequences give statistically more significant results compared to normal image sequences.

## 1.5 Methodology

Recognition of facial AUs can be performed in three steps: (1) Data collection and pre-processing, (2) Feature extraction and (3) Classification. An overview of our recognition model is given in Fig. 1.1. In data collection, we look at both the single static images and image sequences. Single static images are used in conjunction with SVMs and image sequences are used in conjunction with RNNs. The data is taken from the Cohn-Kanade database. In the pre-processing phase, we perform the following steps: (1) Manual detection of facial features such as eye centres, (2) Face normalization using these eye centres, and (3) Face detection and segmentation. For feature extraction, we apply Gabor filters to extract the salient features from our data. We choose five frequencies and eight orientations as in Tian et al. (112). The output responses of the 40 Gabor filters are further down sampled by a factor of 16 and normalized to unit length as in (13) to reduce the dimensionality. The classification phase involves the use of SVMs and RNNs. Support vector machines use single static images depicting FACS AUs for classification, where as RNNs use image sequences for the same. This will provide a basis for our hypothesis, that the use of time variant data will give a better classification. The RNN based model is also investigated in

conjunction with feature reduction and network optimization techniques. The model is further studied for its ability to handle multi-AU classification and the use of difference images for improved classification performance.



**Figure 1.1:** Overview of our recognition model

## 1.6 Thesis Overview

The overview of the remaining thesis is as follows. Chapter 2 provides a context to our work in the field of FER. We review the different applications where FER can be used. In Chapter 3, the FACS is discussed and how this standard is useful for our research. Alternative systems are also reviewed with special focus on the suitability of FACS for this particular work. A detailed literature review about the different facial expression and FACS AU recognition techniques is given in Chapter 4. The chapter first reviews the literature based on the use of feature extraction techniques. Then a brief review of models using spatial and spatio-temporal classifiers is given.

Chapter 5 reviews the standard feature extraction and feature selection techniques such as Gabor filters, principle component analysis (PCA), linear discriminant analysis (LDA) and Haar filters. The fundamentals of neural networks, SVMs, RNNs, hidden Markov models (HMMs) and long short-term memory (LSTMs) are discussed in Chapter 6. This chapter also gives a brief overview and applications of RNNs. Recurrent neural networks being the major focus in this research, we review the different learning algorithms such as back propagation through time (BPTT) and real time recurrent learning (RTRL).

In Chapter 7, we discuss the classification process using SVMs in conjunction with single static images. Chapter 8 focuses on the use of RNNs and the base line recognition rates we achieved with the use of Gabor filters as feature extractors. The results are indicative about the successful use of RNNs in the field of FACS AU recognition. We also discuss the importance of the number of hidden units and their effect on the overall performance of the recognition model. Later in the chapter, we discuss different dimensionality reduction techniques, such as frequency selection, local Gabor filters and PCA. Finally we study the use of optimization techniques for RNNs including weight decay. The last section of the chapter generalizes our RNN based FACS AU recognition model for classifying other FACS AUs.

Chapter 9 is dedicated to the study of single AU/single RNN vs multi-AU/single RNN. This study is aimed at understanding the advantages and disadvantages of using a single robust RNN for the recognition of set of FACS AUs and also the effect of inter-AU correlation present in the Cohn-Kanade database. Chapter 10 deals with the use of difference images in the field of FACS AU recognition and their importance. This chapter specifically deals with the advantages offered by difference images and their usefulness towards improving the models performance. In Chapter 11, we outline our conclusions and suggest other future work that can be carried out in this regard.

## Chapter 2

# Why Facial Expression Recognition?

### 2.1 Introduction

The significance of a classification model depends on its potential for applications. Facial expression recognition (FER) is a broad area of research that not only looks at classification of prototypic expressions such as happy, sad, angry etc, but also the use of frameworks such as FACS (45). Such classification of expressions or actions has been explored in different applications beyond the classical tools for HCI. A tutorial on FER is given in (27). The emphasis of this chapter is to review some of the applications and how the recognition of facial expressions play a vital role in these applications. Facial expressions on their own and when combined with other non-verbal behavior give out important cues about the subjects in question. The use of such information is discussed in applications such as sign languages, entertainment, psychological and monitoring systems.

### 2.2 Sign Language

One of the applications using expression classification is the sign language recognition. Sign languages are a primary means of communication for deaf individuals around the world. The deaf communities around the world could benefit from

the sign language translation systems. Facial expressions in a signed language are used to convey the state of mind of a speaker. This is in addition to the hand gestures that usually form a large part of communication in signed languages. Due to the absence of vocal communication, facial expressions in signed languages provide lexical, adverbial and syntactic functionality. This additional information that extends beyond the expressions themselves is absent in the spoken language. In signed languages, the emotional states such as sad and happy are conveyed by producing the standard prototypic expressions on the facial region. Even though the basic structure of the sign language remains the same, different versions are available around the world which basically differ by their countries own language. There exists different sign languages including American Sign Language (ASL), South African Sign Language (SASL) and Taiwanese Sign Language (TSL). The recognition of such expressions can be simply performed as a normal expression recognition process, where for every expression to be recognized a set of samples are collected. These samples are used to train the classification system. Work in (86) & (115) used such a system to recognize some of the basic expressions in ASL. Similar work for the recognition of facial expressions in TSL has been done in (78). One of the drawbacks with such a mechanism, however, is the repetition of the whole process when a new expression is introduced. A new classifier with a whole new set of training examples depicting the new expression requires extra resources and time.

One solution would be the introduction of an intermediary framework which translates the expressions into some basic actions. Using such a framework will enable the recognition of expressions as a set of actions. In (118), (119), recognition of SASL expressions was performed using FACS as a framework. Here the addition of a new SASL expression to the classification model was just an addition of extra AUs to expression mapping in the translation module . This leaves the recognition code unchanged. Thus the use of FACS framework reduced the effort needed to introduce new expressions into the recognition system.

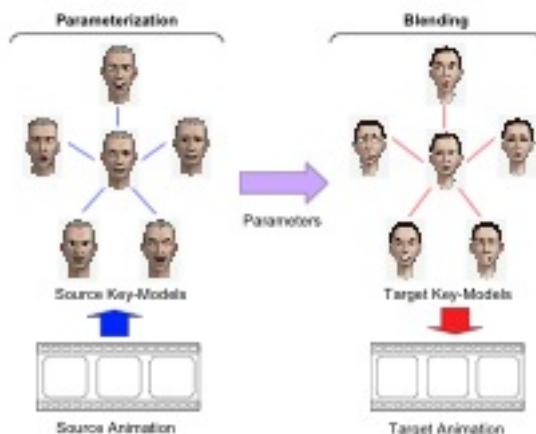
## 2.3 Psychological Studies

One of the important applications of expression recognition is towards the psychological studies. Humans have the capability for interpreting the facial expressions and then access the emotional state of a person. A person's response to a particular situation or question gives important cues which are then used by the psychologists to access that person's state of mind. Many systems have been developed to identify expressions that are thought to be associated with the emotions. Some of these are the Maximally Discriminative Facial Movement Coding System (MAX) (62), A System for Identifying Affect Expressions by Holistic Judgment (63), the FACS (43) and EMFACS (44). Many of these systems give the same labels to different facial actions. Under these conditions, they assume that facial expressions and emotions expressed by a subject have an exact correspondence. This kind of assumption is problematic (103) where a certain nonverbal expression may be used as an emotion display in certain communities but may not be the same in others. As a result descriptive systems such as FACS and MAX were highly recommended for emotion research in psychological studies. The recognition of spontaneous facial expressions (12) also improved the state-of-art psychological studies. Previous research has shown the high correlation between spontaneous expressions in psychology (101).

## 2.4 Entertainment

The entertainment industry also focuses on the recognition of facial expressions for the creation of better animations (93). Animation of avatars has gained interest and is one of the booming field in entertainment industry. Animation is usually performed using facial expression cloning as in (98). Here an example source model is cloned to get a target model, preserving the characteristic features of a target model as shown in Fig. 2.1. This form of cloning preserves the facial expressions of the source model and also the characteristic features of the target model defined by the animators. This has many real world applications such as video games. Another alternative is to model an avatar similar to the source model, and then replicate the facial expressions. This finds its applications in





**Figure 2.1:** Example based cloning model used in [Pyun et al., 2003]

animation movies where real faces are modeled including all the major features. Cloning of facial expressions from the source to the avatar then produces natural looking expressions on the avatar's face and observers are able to view the same expressions that a source will generate under the given conditions. Work has also been carried out in identifying subjects and the emotions generated by a local avatar with a global audience (123).

## 2.5 Security

Recognition of facial expressions has been introduced into the modeling of security systems. The most common area is the access control at airports and high profile buildings. Previously, the emphasis was only on face recognition for security purposes, but recently the importance of analyzing facial expressions has seen increased interest. One does not want to model a system which is able to classify only neutral expressions and falsely rejects an authorized person from entering a high security area just because he looks confused. The goal here is to train security systems for different facial expressions that a person may display, varying head poses, lighting conditions and other external accessories such as glasses that may change a person's basic appearance.

## 2.6 Bi-modal Emotion Recognition

Humans use all the modalities such as audiovisual, physiological and contextual for the recognition of emotions (67). This enables humans to have high recognition rates for emotion recognition. However, most of the research in emotion recognition has focused only on the use of single modality such as facial expressions or speech (32). It would however, benefit an emotion recognition system to use multimodal emotion data and in order to achieve the recognition rates of humans. Bimodal emotion recognition systems were investigated focussing on combining audiovisual modalities such as speech and facial expressions (67),(34). For the successful implementation of bi-modal emotion recognition, the accurate recognition of facial expressions plays a vital role.

## 2.7 Surveillance/ Monitoring

There is growing interest in the visual surveillance of humans and high security areas such as airports, sports grounds and borders. The variation in the facial expression is one the major problems faced by the surveillance systems that are basically designed around accurate face identification. The accurate recognition of facial expressions depicted by a subject will give away cues regarding the state of mind and can alert the authorities above any suspicious behavior. The study of depression and suicide faces (60) can provide crucial information about the psychological state. Using this knowledge of facial expressions, a simple video surveillance will allow the warden of a jail to monitor the inmates who are suicidal and are confined to a small room. Similar monitoring systems based on nonverbal expressions can be used for psychiatric patients (40) and deception cues in courtrooms (52).

## 2.8 Automated Tutoring

Automated individual student tutoring is gaining significance. The results in (19) showed that students tutored on a one-to-one basis outperformed their peers taught in classroom settings. Computer-based tutoring has the potential to alter

the nature of education by providing a customized learning experience to individual students (10). The tutoring proves more efficient if automated computer systems adapt to the emotional and cognitive state of a student (10). Using estimation of the emotional state of a student being tutored, FER has shown to be effective (120), (24). In (120) different FACS AUs were studied to estimate a student's emotional state and thus determining how fast or slow he would prefer to watch a lecture during the tutoring session. In (24), the video samples were taken and analyzed to estimate the affective state which were categorized into interested, thinking, tired/bored, confused, confident/proud, frustrated and distracted. Studying the facial expressions using a sequence of frames was extremely helpful towards an effective estimation of the student's emotional state.

## 2.9 Summary

This chapter described different contexts in which FER can be used, thus highlighting the need for accurate FER.

## Chapter 3

# Facial Action Coding System

### 3.1 Introduction

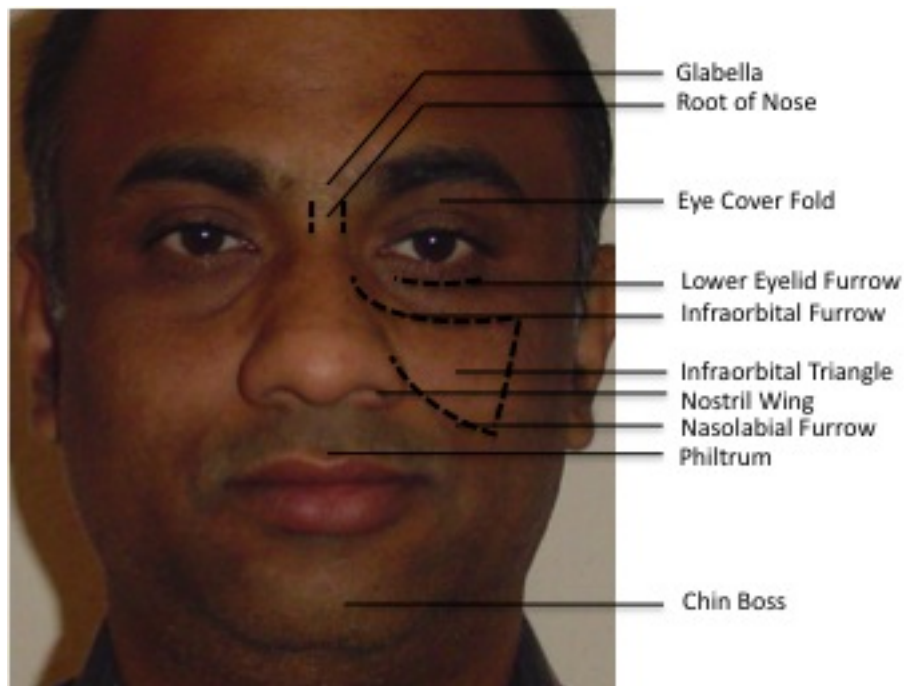
In the FER literature, the successful recognition of facial expressions has paved the way for the development of systems that are capable of recognizing minute changes in the face regions rather than the full expressions. This was motivated by the real world human-to-human interactions that are generally based on a set of subtle changes. However, systems that could explain an expression could hardly explain the many subtle changes in the face region that form the basis for the generation of such an expression. In the case of recognizing what a person/subject is generating on his/her face, one needs more knowledge than the basic expressions. The knowledge of methods that help towards measuring facial actions have been devised (46). In the real world, communication between humans does not always include the generation of an expression to communicate a message; in contrast, it could be a simple change in one of the face regions or facial features. This theory needs a system which can explain the subtle changes in the face due to internal muscle or skin transformations, which in turn can explain the basic components needed for expressions. This led to the development of FACS by Ekman and Friesen (43).

The main goal of FACS was to design and development of a system which could distinguish between visible facial movements. It was to describe expressions in the form of basic components. Paul Ekman and W.V. Friesen developed the original FACS in the 1970s by determining how the contraction of each facial muscle singly

### 3.1 Introduction

---

and in combination with other muscles changes the appearance of a face. The entire face region was divided into separate regions as shown in Fig. 3.1. The description of each term is given in Table 3.1. They associated the appearance changes with the action of muscles that produced them by studying anatomy, reproducing the appearances, and palpating their faces. Thus, they could create a reliable means for skilled human scorers to determine the category or categories in which to fit each facial behavior. Unlike other intrusive methods such as electromyography, where physical presence of individuals is needed to connect wires to their faces to measure the facial behavior, FACS was designed for use on subjects who are unaware of the fact that they are being monitored. This limits the use of FACS for coding facial expressions using only visual measurements. This in turn makes the measurement of those muscle movements difficult which produce no appearance change or whose effect on the face is too subtle to study for a human in order to give a reliable reading.



**Figure 3.1:** Different Facial Areas and their Names as Given in (45)

Facial Area	Description
GLABELLA	Area of the forehead between the eyebrows.
ROOT OF NOSE	The beginning of the nose between the eyes; also called the nasal root.
EYE APERTURE	The degree to which the eye is open; the eye opening
EYE COVER FOLD	The skin between the eyebrows and the palprebral part of the upper eyelid (the part that contacts the eyeball), which folds into the eye socket.
LOWER EYELID FURROW	A place below the lower eyelid where a line or wrinkle may appear. A line or wrinkle may be permanently etched into the face; if so, it will deepen with certain AUs. If not, it should appear when these AUs are contracted.
INFRAORBITAL FURROW	A place where a line or wrinkle may appear parallel to and below the lower eyelid running from near the inner corner of the eye and following the cheek bone laterally.
NOSTRIL WINGS	The fleshy skin of the side of the nose that forms the outside of each nostril.
NASOLABIAL FURROW	A place where a line or wrinkle may appear which begins adjacent to the nostril wings and runs down and outwards beyond the lip corners. In some people it is permanently etched in the face; if so, it will deepen with certain AUs. If not, it will appear on most people's faces with certain AUs.
PHILTRUM	The vertical depression in the center of the upper lip directly under the tip of the nose.
CHIN BOSS	The skin covering the bone of the chin.
SCLERA	The white part of the eyeball.

**Table 3.1:** Description of Facial Areas Given in (45).

The basic unit of measurement in FACS is an AU which denotes subtle change in a face region rather than the muscles actions themselves. The reasons behind taking AUs as the fundamental measuring unit are; few appearance changes by a set of muscles are indistinguishable and are thus combined under one single AU. The other reason was that appearance changes produced by one muscle were sometimes separated into two or more AUs to represent relatively independent actions of different parts of the muscle.

## 3.2 Design of FACS

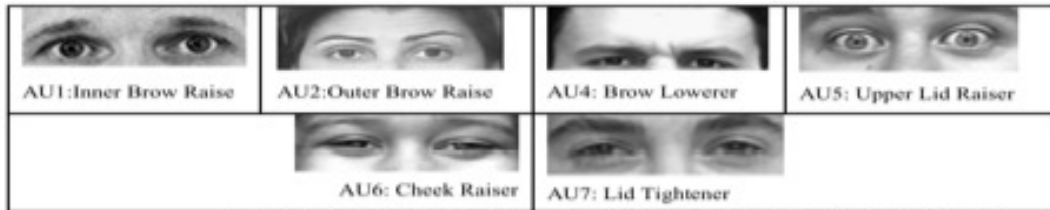
FACS combines AUs into groups. The groups are based upon the location and/or the type of action involved. The two major groups are upper face AUs and lower face AUs. Upper face AUs account for eyebrows, forehead and eyelids; lower face AUs account for nasolabial, mouth and chin regions. Lower face AUs are divided into five groups: up/down, horizontal, oblique, orbital and miscellaneous actions. Each AU in every group is assigned a unique number for coding.

### 3.2.1 Action Units and Their Muscular Action

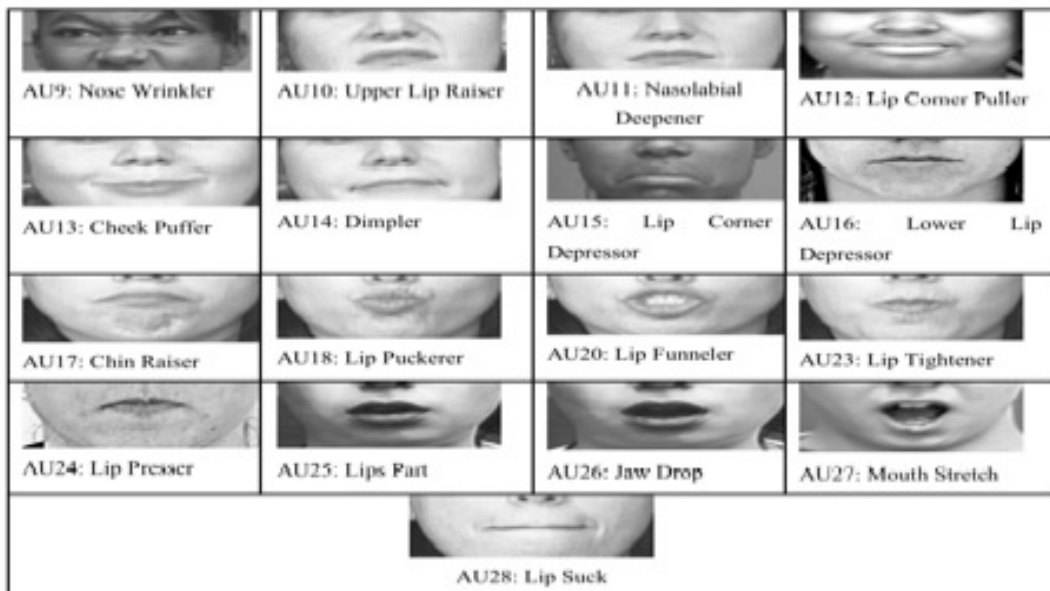
In the original FACS (43), there were 44 AUs ranging from 1 to 46 (AU numbers 3 and 40 were not used). The updated version released in 2002 (45), which included the movements of eyeball and head added an additional 12 AUs numbered from 51 and higher. AUs from 1 to 7 describe upper face actions and AU numbers from 8 to 46 describe the lower face actions. The different AUs defined in FACS and their pictorial depiction are given in Fig. 3.2. The illustrations provided are just to typify the changes. The exact appearance change for each AU varies from one person to another depending upon their bone structure, variations in the facial musculature, fatty deposits, permanent wrinkles, shape of features, etc. (43). However, some common elements appear across people during activation of an AU.

Before discussing AUs in each group, it is important to review the muscles and their resulting actions as described in the FACS system. Such description helps in better understanding of the deformations and the direction of movements that

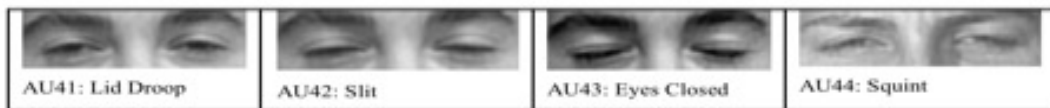
### 3.2 Design of FACS



**Figure 3.2(a): Upper Face FACS AUs and their description**



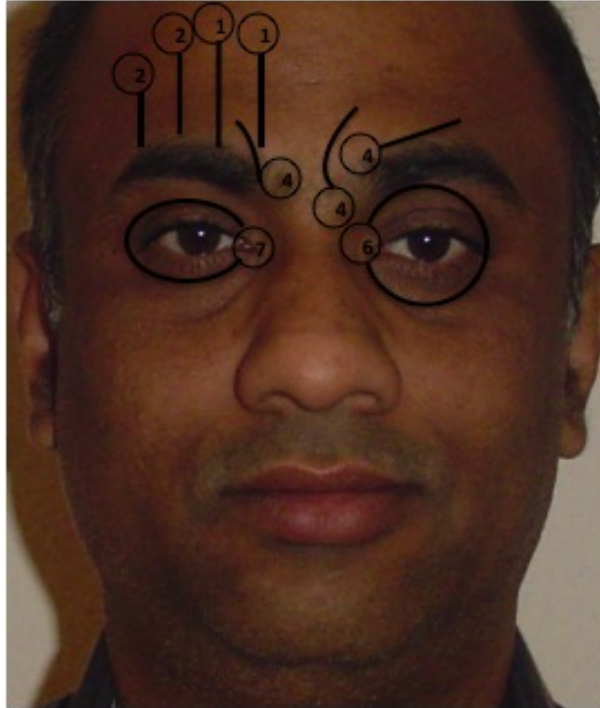
**Figure 3.2(b). Lower Face FACS AUs and their description**



**Figure 3.2(c). FACS AUs related to eyes and their description**

**Figure 3.2:** FACS AUs and their description. Pictures courtesy of Carnegie Mellon University Automated Face Analysis Group, <http://www-2.cs.cmu.edu/afs/cs/project/face/www/facs.htm>.





**Figure 3.3:** Muscle Actions (indicated by numbers) and Directions for Upper Face FACS AUs

appear on the face regions. The different muscles and the action for the upper face AUs are given in Fig. 3.3. The direction of muscle action is towards the AU number. Similar illustrations for the other AUs defined in FACS are present in the FACS manual (45).

In each group, every individual AU is explained in three parts:

- The appearance changes due to the AU with detailed illustrations using photographs and video clips
- Steps for performing the AU
- Rules for scoring the intensity for the AU

The appearance changes produced by a muscle that allows the specific action to be distinguished from the other AUs is first step towards establishing the appearance change due to an AU. A description of the movement due to the

muscle action and the static appearance of the face when such an action is held for a period of time (static images) is given. For example, the appearance changes due to AU 4 which relates to brow lowered is given as:

1. Eyebrow is lowered. This action may include only the inner portion of the eyebrow or both inner and central portion of the eyebrow.
2. Eye aperture may be narrowed by pushing the eye fold.
3. Eyebrows are pulled closer to one another.
4. Deep vertical wrinkles are formed between the eyebrows. In few cases the wrinkles are at an angle of 45 degrees with some horizontal ones at the root of the nose.
5. An oblique wrinkle or muscle bulge may be seen above the middle of the eyebrow to the inner corner of the eyebrow.

While looking for the above appearance changes, one studies the image(s) and/or video(s) provided and carefully identifies the following as described in (45):

- the parts of the face that have moved and the direction of their movement;
- the wrinkles that have appeared or have deepened;
- the alterations in the shape of the facial parts.

The above steps help a scorer to accurately identify the relevant muscle actions and the AU that describes such an action.

The next step deals with performing the AUs. Making AUs on ones own face while looking into a mirror is the most important activity in practicing AU scoring. The FACS manual provides tips on how to perform the AU accurately and also sheds some light on the common mistakes while mimicking these AUs. Once the AUs are mimicked accurately by a scoring the focus then shifts to the third part which is scoring the intensity of the AUs.

### 3.2.2 AU Combinations

In most real world interactions, activation in the face region occur due to a set of muscles. In this case, each AU is attributed to changes generated by a set of

muscles. In other cases, different AUs are attributed to the change in a single muscle, where different parts of the same muscle are activated individually.

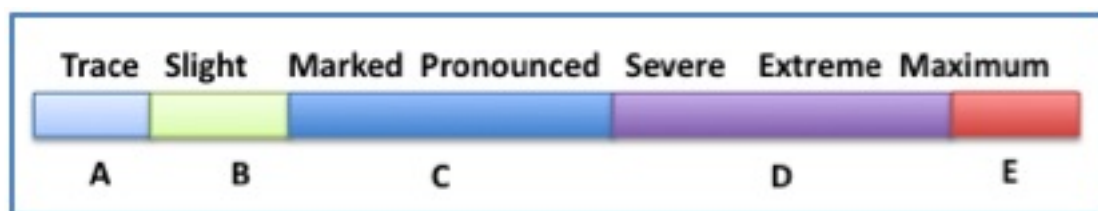
When multiple AUs coexist, a set of combinations of AUs are formed. 7000 such combinations have been observed in FACS (45). Co-occurring AUs produce appearance changes that are relatively independent, changes in which one action masks another or a new and distinctive set of appearances can be observed. When the appearance of each AU in combination is identical to that of its appearance when it occurs alone, such combination of AUs is said to be additive. In additive AU combinations, the evidence of each AU remains clearly recognizable because different AUs have combined without distorting or changing the appearance of other AUs. In some cases, these additive changes are totally independent. Here the AUs are from different parts of the face. Apart from additive combinations, there exists a set of AU combinations that are non-additive or distinctive. In such cases, the appearance change is in some way different from the separate appearances caused by the individual AUs. Here the appearance change is not simply the sum of the appearance changes due to different AUs, but a distinctively new appearance change.

The other relation that exists between AUs in a combination is dominant and subordinate AUs. When actions co-occur one AU can interfere with the detection of other AU by masking the signs of its action. The AU that obscures the actions of other AUs is called the dominant AU. Dominant AU appearances are apparent, while it tends to obscure the signs of other AUs. An AU whose signs are obscured by a dominant AU is known as a subordinate AU.

### 3.2.3 AU Intensity and Scoring

One other aspect of FACS is the intensity scoring of the AUs once their appearance is determined. The intensity of an AU indicates the strength of the actions that result in variations in the intensity of the appearance change. Intensity of an AU is rated from A to E. An intensity of A indicates that there is a trace of an AU. An intensity scoring of E indicates that the AU is present with its highest levels. The scale of evidence and the respective intensity scoring is given in Fig. 3.4. The other terms used in describing the peak intensity of an AU are the apex

of an action. The apex of an action is the point of the greatest excursion or change within that action. For example, if an AU increases the aperture of the eye, the apex is the moment when the maximum eye aperture first occurs. The apex is not an absolute amount of change, but the greatest change that occurs for a particular AU in a particular event (43). It is the time when the action is strongest. In many research areas, the first frame when the apex of an action occurs is of importance, while in others the duration of the apex, i.e. the time elapsed between the start and end of the apex is measured. The type of scoring varies from application to application.



**Figure 3.4:** Scale of Evidence and Intensity Scores Relation

Other important information while scoring is that of unilateral actions. These actions occur only on one side of the face and they use an abbreviation “L” for the left and “R” for the right depending on the presence of AUs on the face. The abbreviation “L” or “R” is used in front of the AU number. In case of action which occurs on both sides of the face, but its presence is stronger on one side of the face than the other than we use a prefix “A”.

### 3.2.4 Reliability of FACS

The fundamental reliability issue in FACS is to know whether independent persons would agree on the scoring that is given by them for a particular event. More emphasis is laid on learners who learned FACS without any help from the developers themselves, and to see whether they agree among themselves and/ or with the developers. The reliability issues are basically divided into two categories; one based on description and the other based on location. The reliability issue based on description includes; what happened, what are the AUs responsible for

a behavior that is observed on the facial region etc. Reliability issue based on location includes; when did a particular change happened, at what moment in time did the change started and ceased. For example, the lips have moved; in order to describe this action one has to study the type of movement that was involved. Movements like; were the lips parted, or pressed together, were the lip corners droop/rose. To locate this movement one has to determine the video frames which depict the start and end of the movement. These two questions are to some extent independent of one another. One may have a high reliability on description but low on location. For many applications and in particular to the FACS system itself, the reliability on description has received more emphasis. The reliability for both description and location is evaluated using two terms:

- agreement among independent coders
- agreement between a learner and an expert.

### 3.3 Application of FACS in Computer Vision

FACS has been used in many applications besides acting as a basic framework for emotion recognition. In this section we review some of the areas where FACS has been used. The study of reflexive responses on the facial region caused by pain among patients with Alzheimer’s disease is important for pain assessment. This is due to the diminishing cognitive abilities in these patients. In (5) the use of FACS as an index for pain variability has been successful. They were able to investigate the differences in facial expressions as function of level of discomfort among older adults with and without Alzheimer’s disease.

The use of FACS is also seen in deception studies. An entire show “Lie to me” by Tim Ross is now underway, in which Ekman, the developer of FACS himself studies the micro expressions, the tone of subjects voice and the choice of words in deciding whether a subject is lying.

Over the years FACS has also been used as a basic system for development of new systems for specific groups. Examples of such systems are baby FACS and ChimpFACS. The development of baby FACS (91) was motivated by the observation that the emotions generated by adults are different from that generated

### 3.4 Suitability of FACS for Measuring Facial Expressions

---

by infants. The difference in the distribution of facial tissues, which makes the appearances of some muscular actions different, paved the way for the baby FACS system that is basically a version of the original FACS itself. It was developed by Harriet Oster and studies the infant's faces. FACS was also used to develop ChimpFACS (79). The objective was to develop a tool for measuring facial movements in chimpanzees based on the original FACS. Such a tool enables the study of structural comparison of facial expression between humans and chimpanzees in terms of their common underlying musculature (79).

### 3.4 Suitability of FACS for Measuring Facial Expressions

Research in FER field has seen the use of FACS as an intermediary framework for expression recognition. This was partly due to the level of detail FACS provides in describing expressions and partly due to its ability to code expression intensity, making it famous in the psychology community. FACS is not only able to measure the standard expressions generated by human subjects, but can also measure the changes that one can produce randomly. This makes it flexible for expression recognition in applications where other standard systems fail.

### 3.5 Alternative systems for Measuring Facial Expressions

Alternate systems that describe the facial expressions are Maximally Discriminative Facial Movement Coding System (Max) (62), Emotion EMFACS (44), AFFEX (63) and Moving Pictures Expert Group Synthetic/Natural Hybrid Coding (MPEG-4/SNHC). Maximally Discriminative Facial Movement Coding System (MAX) was developed by C.E. Izard in 1983 at University of Delaware (62). It was last revised in 1995. The principle goal for developing the coding system was to provide an efficient, reliable and valid system, for measuring the emotion signals in the facial behaviors of infants and young children. With some modifications in the description of the appearance changes, this coding system can be

### 3.5 Alternative systems for Measuring Facial Expressions

---

used to measure emotion signals at any age. Izard's MAX is theoretically derived, because it codes just those facial configurations that Izard theorized correspond to universally recognized facial expressions of emotion. The disadvantage of theoretically derived systems is that they cannot discover behaviors that were not positioned in advance. MAX uses facial movement formulas derived from prototypical, universally recognized adult facial expressions to specific appearance changes of eight emotions such as surprise, joy, sadness, anger, fear, disgust and contempt. Izard based his formula on observation of infants. Like AUs in FACS, MAX analyses each emotion individually in terms of appearance change (AC). Like FACS AUs, MAX ACs are anatomically based on the muscles of the face. However, MAX ACs fails to comprehensively describe the full range of visibly distinctive facial movements, which FACS AUs are able to accomplish.

EMFACS was introduced in (44). EMFACS is a system to score only the FACS AUs that are relevant for detecting emotions. The goal of EMFACS is to reduce the scoring time when the investigator is only interested in emotion signals. As in the FACS system, the actions that can be scored and that cannot be scored are previously defined in the EMFACS. The coder makes no inference about the meaning of any facial behaviors defined in the EMFACS. This system requires the knowledge and the ability to identify a) the AUs involved in any facial movement b) the intensity of the AUs involved and c) the degree of asymmetry. The AUs that do not affect the interpretation of the emotion under study are not noted. However, this system is not as descriptive as FACS since it is only based on emotions.

AFFEX (63) is a system for identifying expressions by holistic judgment. This system is basically used for scoring infants emotions and their expression behavior. Using this system, the expressive response to many different situations was measured. Moving Pictures Expert Group Synthetic/Natural Hybrid Coding (MPEG-4/SNHC) standard is another approach to describe the facial movements. It is used for face animation using facial definition parameters (FDP) and facial animation parameters (FAP) streams. MPEG-4 SNHC uses 68 FAPs which are not comprehensive enough to describe the facial movements. However, MPEG-4 SHNC is widely used to animate computer generated graphics.

## 3.6 Summary

In this chapter, the basic goal of FACS and its design is explained. This chapter forms the framework on which we will base our recognition of AUs. FACS provides a system which describes the subtle changes that occur in the face region. Since we will be looking at the recognition of FACS AUs rather than emotions, the understanding of FACS becomes crucial for this thesis as a whole.

The basic face regions and the AUs associated with these face regions form the basis of FACS system. The different AUs, the underlying muscles that are responsible for their activation and the procedure to mimic these AUs is explained. Once the of an AU is known, its intensity scoring becomes crucial. The different levels of intensity and the procedure for scoring the same are also described. In addition, the different terminology used to describe the relation between AUs when they occur in combination is described. Alternative systems such as MAX, EMFACS, AFFEX and MPEG-4 SNHC are also mentioned, including the advantage of FACS over these standards.

+



# Chapter 4

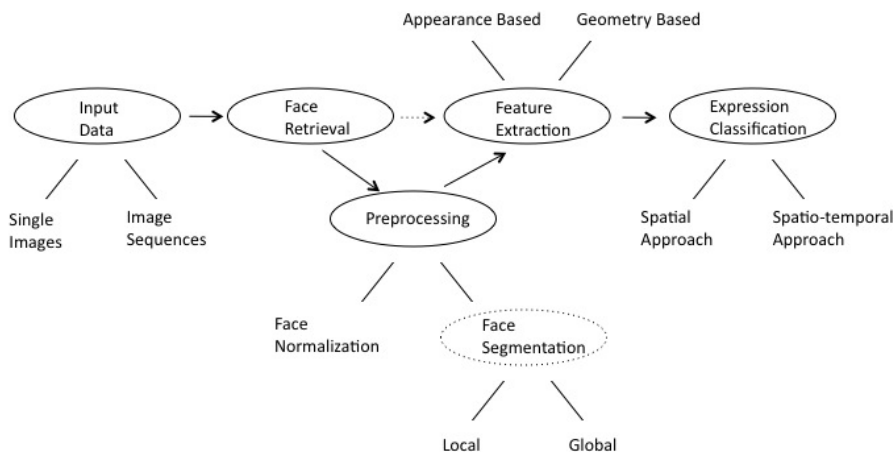
## Literature Review

### 4.1 Introduction

Facial expression recognition has been a major research area in the field of computer vision and image processing. Many new methods have been proposed to the existing algorithms as well as novel approaches have been researched upon to improve the standard of FER. Pantic and Rothkrantz (94) has carried out a state-of-art literature review in FER. In this chapter we review the existing literature on FER. This survey includes the major contributions that impacted the development of FER systems, along with some less known algorithms to present a more complete and comprehensive review. In accordance with our work, we place emphasis on the use of static and time variant data, type of feature extraction and the role of spatial and spatio-temporal classifiers on the performance of classification models.

### 4.2 Facial Expression Analysis

Facial expression analysis is a complex task because of the variations in faces based on the individuals, age, ethnicity, gender etc. The complexity is further increased by the background noise and different lighting conditions. The general framework for FER is given in Fig. 4.1. Different researchers may opt to avoid some of the stages such as face normalization and segmentation. In our litera-



**Figure 4.1:** Overview of an general FER system

ture review, we discuss the work carried out by different researchers that have significantly improved the state-of-the-art in FER. Automatic FER systems can be broadly divided into different stages; input data, face retrieval, preprocessing, feature extraction and expression classification.

### 4.2.1 Input Data

Based on the use of different types of input data, the systems can be broadly divided into two models: one that uses single static images, and the other that uses image sequences. In our work, the image sequences are of high significance because of their ability to provide temporal information.

### 4.2.2 Face Retrieval

In the face retrieval stage, an automatic face detector is used to locate faces and isolates them from the background. The complexity of the face retrieval stage is governed by the requirements of an analysis algorithm. Most of these algorithms need an exact location of the face to perform accurate face retrieval. Face detection algorithms should also be able to distinguish between face like images such as photo frames.

### 4.2.3 Preprocessing

Preprocessing is performed in two stages on the retrieved face region; face normalization and face segmentation. The number of steps in each category depends on the requirement of the analysis algorithm and that of the classification model. The preprocessing step, however is optional. Face normalization is performed to scale, rotate and standardize images. This is required to offset the normalization factors that may be problematic for classification. Face segmentation is a process that provides data on a global or a local level for expression classification.

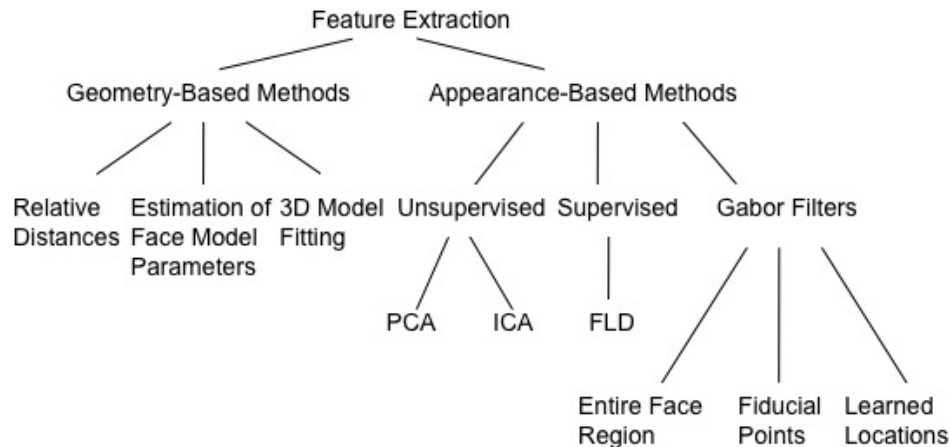
### 4.2.4 Feature Extraction

Feature extraction can be categorized into two approaches: appearance-based methods and geometry-based methods. The appearance-based methods use the color information of image pixels to extract information about the facial features, while the geometry-based methods analyze the geometric relationship between the key feature points, known as fiducial points, on the face.

### 4.2.5 Expression Classification

Expression classification is the last stage of an automatic FER system. This process can be divided into the type of analysis and the approach of a classifier. The measurement of facial expressions which is an important criteria can be done using judgment-based and sign-based approaches (50). In judgment-based approach facial expressions are classified into a set of predefined emotions usually known as basic emotions introduced by Ekman and Friesen (42). On the other hand sign-based approaches measure the facial expressions in terms of facial actions describing the possible visible changes that can occur on a face region. Facial action coding system developed by Ekman and Friesen (43), is one of the sign-based system that gained interest among many researchers in the recent years, because of its ability to recognize subtle changes on the facial region.

The classification approach can also vary based on the data and classifier used such as spatial-based and spatio-temporal based approaches. Spatial-based approach involves the use of a classifier such as a neural network. Here, the direct



**Figure 4.2:** Overview of an general FER system

use of face images forms the input data. Spatio-temporal approach involves the use of RNNs or HMMs capable of modeling the dynamics of the facial actions.

## 4.3 Different Approaches

In this thesis, the focus is on the use of different types of input data, the use of feature extraction/selection techniques and the use of different classifiers. In Sections 4.4-4.6, we first review the literature based on the use of feature extraction techniques used; geometry-based and appearance-based methods. A general taxonomy based on different feature extraction techniques is given in Fig. 4.2. A comparative analysis is provided based on both appearance based and geometry based methods. We then review hybrid architectures using appearance and geometry based methods. It also becomes vital to review the literature based on the classifiers and their ability to handle spatio-temporal data, as this is the main thrust of our work. In Section 4.9 we review the spatio-temporal approach adapted by different classifiers. Section 4.10 then sheds some light on the use of RNNs in facial expression and FACS AU recognition.

## 4.4 Geometry Based Methods

Geometric positions of key facial feature points and their relative distances to one another have been used in developing many FER systems. These key facial feature points on the face are referred to as fiducial points of the face. Fiducial points are located along the eye, eyebrows, forehead and mouth. However, many systems identify the set of fiducial points on the entire face. The use of location and relative distances of fiducial points as input vectors for expression recognition is motivated by fact that the expressions have a substantial affect on the movement and size of the facial features.

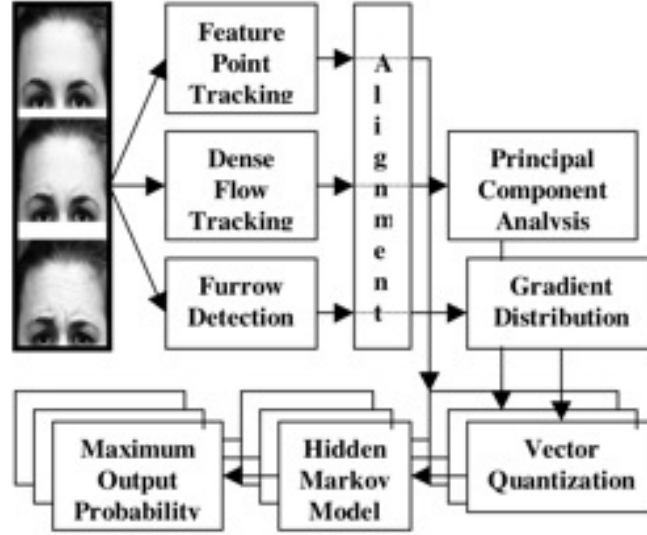
In a broad sense, the input feature vectors for the located fiducial points and the method for conversion of their relative distances into a feature vector mainly affect expression recognition using geometric methods. The localization of the fiducial points and the process of locating them exactly will affect performance of the system. One should use reliable methods for locating fiducial points to ensure good recognition rates. Converting relative distances of fiducial points into feature vectors should be done quickly. Here, we discuss few methods out of the existing ones, which are used in FER.

### 4.4.1 Relative Distances

A straightforward method of using location and displacement of facial features is to convert them directly as a feature vector. Sako and Smith (104) developed their system using the automatic measurements of the facial feature dimensions and their positional relationship such as distance between eyes, height and width of mouth and face to construct their feature vector. They tested their system with samples from one subject. They achieved an average recognition rate of 70.6% for expressions normal, angry, surprise, smile and sad for that specific subject.

Lien et al. (77) developed their system using feature point tracking, dense flow tracking and furrow detection (shown in Fig. 4.3) for the recognition of upper face expressions. The feature vectors were calculated by the displacement of 6 fiducial points located at the upper boundaries of the left and right eyebrows. The displacement of each fiducial point was calculated by its position in the neutral video frame to its position in the present frame, which forms the feature vector.

They classified three AU expressions such as AU4, AU1+4 and AU1+2 using HMM as classifier and achieved an average recognition rate of 85%.

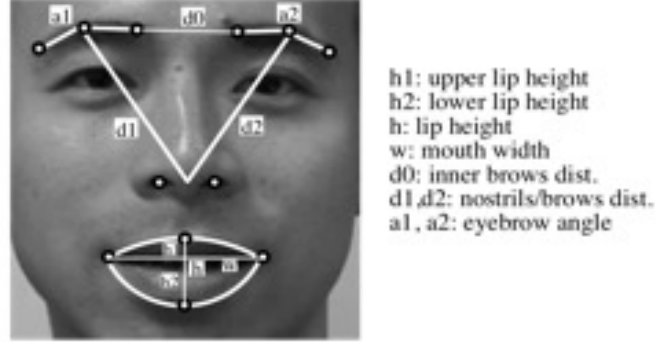


**Figure 4.3:** System structure for Lien et al. (77)

Cohn et al. (29) used feature point tracking for the recognition of 15 AU and AU combinations present in FACS. They located 37 fiducial points (shown in Fig. 4.4) in the first frame and tracked them in the image sequence using a hierarchical algorithm and then classified them using discriminate analysis. The average recognition rate for the AU and AU combinations in the brow, eye and mouth regions were 91%, 88% and 81%, respectively.

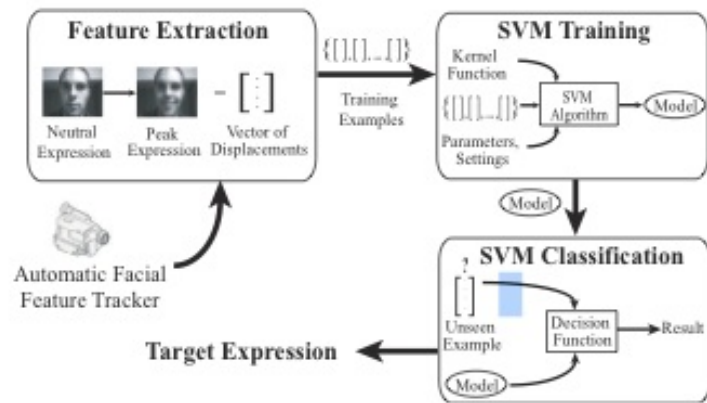


**Figure 4.4:** Manual tracking of fiducial points in neural frame and automatic tracking in the subsequent frames adopted in Cohn et al. (29)

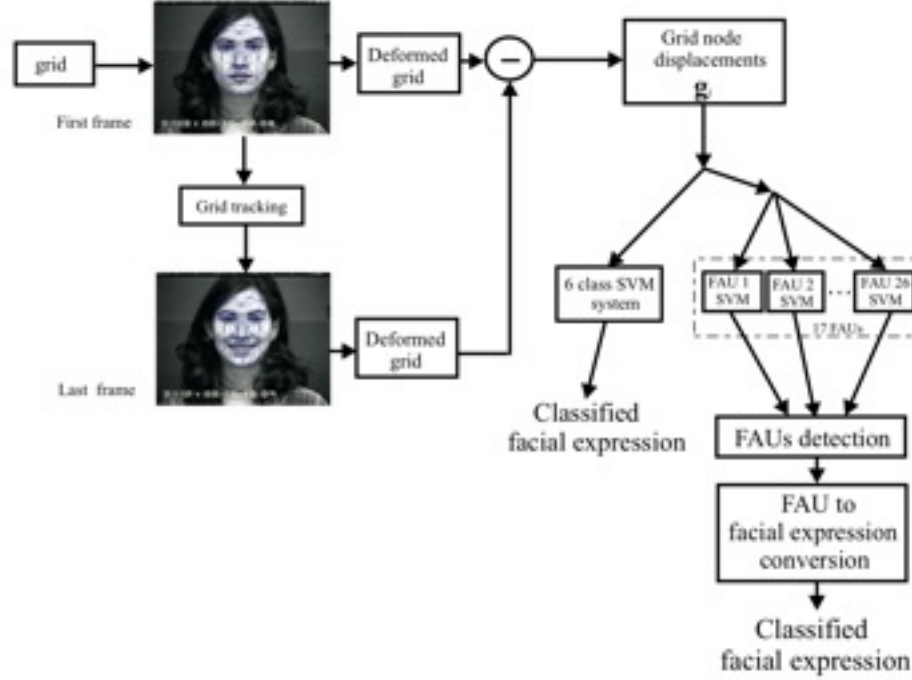


**Figure 4.5:** Geometric face model used to produce the feature vectors in (22)

Bourel et al. (21), (22) tracked the facial landmarks (shown in Fig. 4.5), and measured the change in distance from one frame to the next frame, which are then used to form the feature vector. Their approach was robust to partial occlusion when compared to other FER approaches. Using a nearest neighbor classifier, they classified six basic prototypic expressions with a recognition rate of over 80% for all the six expressions without any occlusion. Philipp and Rana (97) used feature displacements in the video stream as the input for SVMs ( shown in Fig. 4.6) for the recognition of six basic expressions and obtained a recognition accuracy of 71.8% for person-independent classification.



**Figure 4.6:** SVM based automated FER system using feature displacements (97)



**Figure 4.7:** System architecture used by Kotsia and Pitas (70) for FER

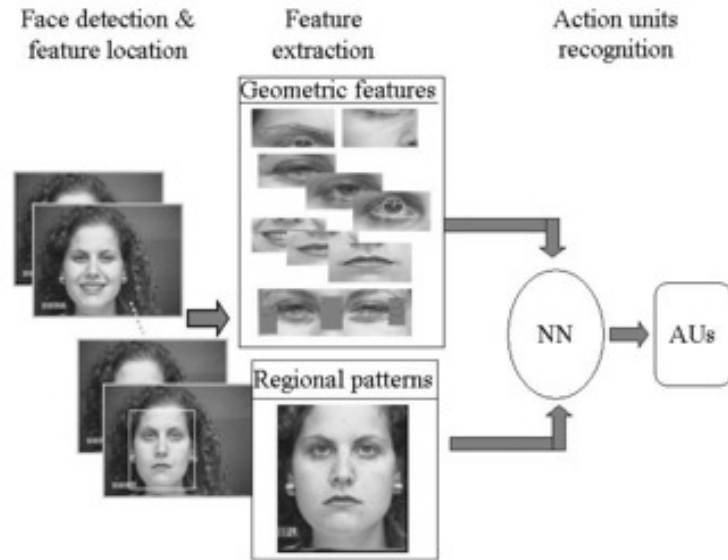
Seyedarabi et al. (107) designed their system for the recognition of facial expressions using feature point tracking and RBF neural networks for the recognition of six facial expressions. The facial features were extracted from a frontal image and were automatically tracked in the image sequence. They achieved an average recognition rate of 91.6% using RBF neural networks. Jose et al. (64) developed their system for feature point based tracking to recognize six prototypic facial expressions under two different spatial resolutions of the frames. They achieved an average recognition rate of 91.4%. Kotsia and Pitas (70) used grid tracking and a deformation system for recognizing six basic facial expressions and a set of facial AUs (shown in Fig. 4.7). The difference of the node coordinates of the selected nodes from the first frame and the highest intensity frame are used as the input vector. They achieved a recognition accuracy of 99.7% for the six facial expressions using multi-class SVMs as classifiers. The use of a displacement vector has also been studied for the recognition of expressions on partially visible faces by Theekapum et al. (109). A 2.5D partial face image is captured



from a viewpoint between  $\pm 45$  degrees. This data is then used to construct a 3D virtual expression. The results presented show the effectiveness of using displacement vectors for the recognition of four basic facial expressions.

### 4.4.2 Estimation of Face Model Parameters

In contrast to the direct use of location and relative distances of the fiducial points, many FER systems use these displacements to estimate the parameters of a face model. These estimated parameters are then used as input feature vector for the expression recognition. Black and Yacoob (18) used such a method for their recognition system. These locations and relative distances were used to estimate the model parameters of a perspective projection model. These model parameters were then converted into intermediate level parameters, which describe the movement of facial muscles. These intermediate level parameters were then used for the classification of facial expression. They achieved a recognition rate of 92% for seven prototypical expressions using a rule-based classifier.



**Figure 4.8:** Feature based system used in Tian et al. (111)

Tian et al. (111) used multi-state models of head and face (shown in Fig.

4.8) to calculate 16 upper and eight lower face parameters. The 16 parameters in the upper face describe shape, motion, eye state, motion of brow and cheek, and furrows, while the eight parameters in the lower face describe shape, motion, lip state and furrows. Separate neural networks were used for the recognition of seven upper face AUs (including neutral expression) and 11 lower face AUs (including neutral expression). The average generalization capability achieved by their system was 96.4% for the upper face AUs and 96.7% for the lower face AUs.

### 4.4.3 3D Model Fitting

The use of 3D model fitting to estimate the parameters is another method which is based on the movement of facial muscles and their activation. Essa and Pentland (41) used this method to recognize a set of five facial expressions using a database of 52 video sequences. The process of manually translating, rotating and deforming the 3D facial model to fit the face in an image is carried out by using view based and modular Eigen space methods. Using these methods, the positions of eyes, nose and lips are extracted from the images. These coordinates were then used to fit a 3D model onto the face in every video frame (shown in Fig. 4.9). The expressions were recognized by using two methods based on the movement of the facial muscles of the physical model and by the use of 2D spatio-temporal motion energy templates. Both methods achieved 98% recognition rate for the five prototypical expressions.



**Figure 4.9:** Feature tracking to fit a 3D model on to a face (41)

Braathen et al. (23) developed a system for the recognition of spontaneous facial expressions including out-of-plane head motions with the use of 3D model



**Figure 4.10:** 3D model fitting used in (23)



**Figure 4.11:** Candide face model used in Graves et al. 2008 (57)

fitting (shown in Fig. 4.10). Using Gaussian radial basis function SVM, they achieved a recognition rate of 90.5% for blinks and 84.5% for brow raises. In (47) a 3D wireframe face model "Candide" (1) was used to track the deformations in video frames. The estimated facial actions were then used to recognize a set of facial expressions using LDA. Using this approach they achieved an overall recognition rate of 92.3% for six basic expressions. The work by (57) used similar Candide model (shown in Fig. 4.11) in conjunction with a learned objective function for estimating the model parameters. These parameters were then classified using a RNN for the recognition of basic facial expressions.

## 4.5 Appearance Based Methods

A different approach used in automatic FER is appearance-based methods. These methods use the color of the image pixels to extract salient features to classify the facial expressions. A wide range of algorithms have been used in this category such as optical flow, dimensionality reduction using PCA and independent component analysis (ICA) which are unsupervised methods, fisher discriminate analysis (FDA) which is a supervised approach and Gabor filters.

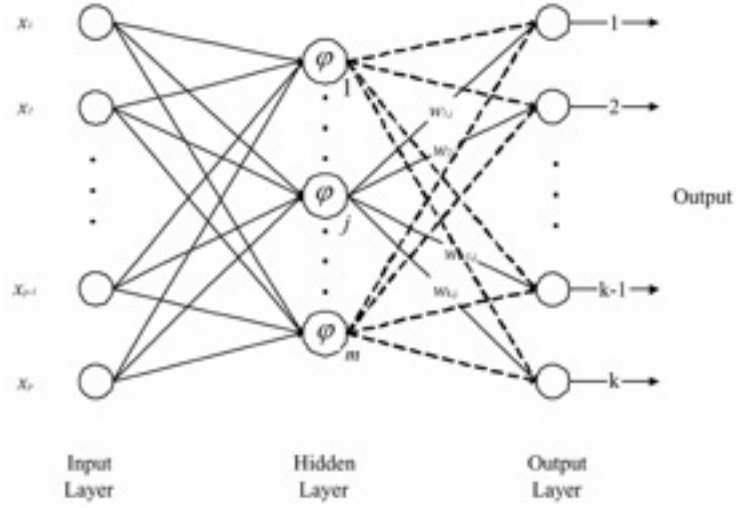
### 4.5.1 Unsupervised Methods for Dimensionality Reduction

In appearance based methods, the information about the features depends on the number of pixels in the image. These pixels are in the order of hundreds even at low resolutions. A significant number of these pixels contain information which may be less useful for the classification of expressions or FACS AUs. In most cases, the values of some pixels do not change from one expression to another and also many pixels are entirely dependent on their neighbors, which make the extracted feature vectors redundant. Removing this redundant data may improve the classification performance and also reduce the dimensionality by many orders. PCA and ICA are most widely used unsupervised methods to remove redundant data and also decrease the dimensionality. In this section we review the FER approaches that were based on the use of PCA and ICA for reducing the dimensionality.

Principle component analysis performs feature selection and reduces dimensionality. When applied to a  $n$  dimensional dataset, we get  $p$  projections where  $p \ll n$  known as principle components. Thus, the resultant components retain most of the original variance and also produce a dataset with a much smaller dimension. However, the major disadvantage of PCA is that it may lose important information for discrimination between different classes.

Donato et al. (37) classified six upper and six lower face AUs using PCA and template matching classifier. The first 30 principle components calculated using difference images were used. They reported a best average performance of

79.3% for the 12 AUs. Along similar lines, Bartlett et al. (7) classified six upper and six lower face AUs using 50 principle components of difference images and a neural network as their classifier. They achieved a recognition rate of 88.6%. They attributed this high recognition rate either to the use of difference images or smaller original image size. Similar experiments were performed by Fasel and Luttin (49) to classify individual AU and AU combinations on a different dataset. They reported a recognition rate of 79% for nine individual AUs and 74% for 16 AU combinations using nearest neighbor classifier.



**Figure 4.12:** Architecture of radial basis function network used in Daw-Tung Lin (35)

Daw-Tung Lin (35) have used PCA to recognize seven facial expressions based on face, lips and eye regions. A hierarchical radial basis function network (HRBFN) (shown in Fig. 4.12) was used to classify seven basic facial expressions based on local features extraction by PCA technique. An average classification rate of 89.21% was reported (top two matches) using face region.

In the FER literature, ICA has outperformed PCA and this was attributed to its capability to learn higher order dependencies in addition to the second order dependencies learned by PCA (37). In expression recognition, most of the information such as nonlinear relationships among pixel gray values, for example edges and elements of shape and curvature are concentrated in the higher

order dependencies (37). However, unlike PCA, inherent ordering of the independent components is missing in ICA. One commonly used ordering parameter is the class discriminability of each component, which is defined as the ratio of between-class to within class variance of independent components. Using the above method, Bartlett et al. (8) and Donato et al. (37) selected 75 components using class discriminability and obtained a performance of 95.5% using cosine similarity measure and nearest neighbor classifier for six upper and six lower face AUs. Using a similar approach Fasel and Lutton (49) obtained performance of 74% for nine individual AUs and 83% on 16 AU combinations using nearest neighbor classifier.

### 4.5.2 Supervised Methods for Dimensionality Reduction

Many researchers have used supervised methods for the recognition of facial expressions and FACS AUs. The most widely used supervised method is the Fisher linear discriminate (FLD), which is discussed below.

The goal of FLD is to project images into a  $c-1$  dimensional subspace in which the  $c$  classes are maximally separated. The use of this classic pattern recognition technique has shown to improve the identity recognition performance over PCA (16). Based on the studies by (16), Bartlett et al. (8) developed a FER system using FLD. They first applied PCA and then FLD to find the projection matrix. Thirty principle components were chosen that gave the best performance, which were further reduced to five dimensions using FLD. They achieved a performance of 75.7% using Euclidean distance similarity measure and template matching classifier.

### 4.5.3 Gabor Filters

Recently, wavelet decompositions have been widely used as an alternative to the other holistic analysis. One of the most commonly used wavelets are Gabor decompositions, which have proved to be one of the successful appearance-based methods. The Gabor decomposition of an image is computed by filtering the input image with a Gabor filter which is tuned to a particular frequency and orientation.

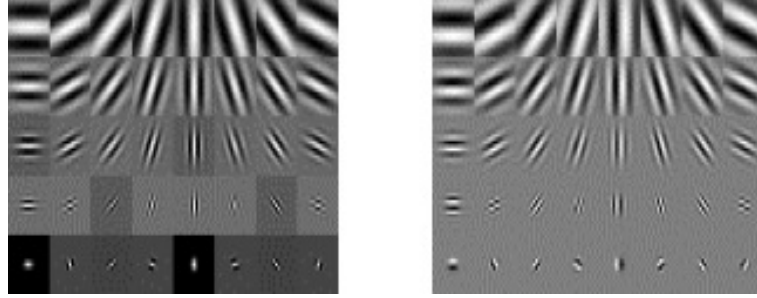
The Gabor filter can be represented in space domain as

$$p_k(x) = \frac{k^2}{\sigma^2} \exp\left(-\frac{(k^2 x^2)}{(2\sigma^2)}\right) (\exp(ik \cdot x) - \exp(-\frac{\sigma^2}{2})) \quad (4.1)$$

where  $k$  is the characteristic wave vector.

In FER however, multiple Gabor filters are used for feature extraction where each filter is tuned to a characteristic frequency and orientation. The combined response is called a Gabor jet. Fig. 4.13 shows the Gabor jet calculated at five spatial frequencies space at half octaves and eight different orientations. The response image of the Gabor filter when applied on an input image using the Gabor kernel is given as

$$a_k(x_0) = \int I(x) p_k(x - x_0) dx \quad (4.2)$$



**Figure 4.13:** The real and imaginary components of the Gabor jet with five spatial frequencies spaced at half-octaves and eight different orientations

Lades et al. (74) were the first ones who showed that Gabor filters are a good representation for facial image processing. Gabor filters can be used to extract the features from a face image using different approaches. One approach is to apply the Gabor filters on the whole face image; a second option is to apply the Gabor filters only at a few selected fiducial points. In the second case, the Gabor response is calculated directly in space domain by convolving each filter at the desired location on the face. However, in the first case it is much faster to use Fourier transforms where the Fourier transformed image is multiplied by the Gabor filter. The result is then inverse transformed back into the space

domain. The filter response is a complex value and usually the magnitude is calculated prior to classification because it changes very slowly with position whereas phases are very position sensitive. A further approach is a combination of the above two where a subset of the Gabor responses at learned locations over the entire face region, frequencies and orientations is selected. All these approaches will be discussed in this section. Apart from how the Gabor responses are generated, one more important aspect of Gabor filters is their configuration. The optimal combination of frequencies and orientations that are suitable for the facial expression or FACS AU recognition plays a very important role in the performance of the recognition system.

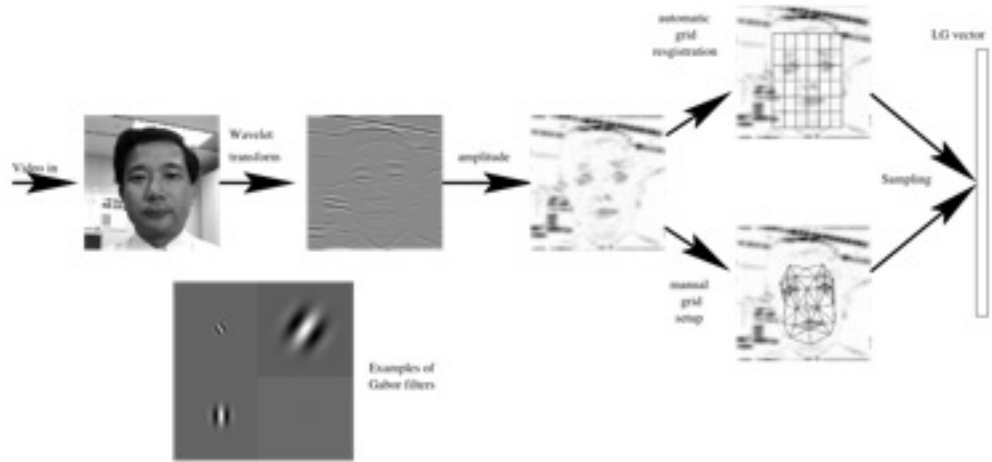
### 4.5.4 Gabor Responses Over Entire Face Region

Gabor filters applied over the entire image regions have resulted in some of the highest recognition rates in the FER literature. A recognition system using Gabor filters over the entire image region and nearest neighbor classifier was developed by Bartlett et al. (8) and Donato et al. (37). A set of five spatial frequencies and eight different orientations have been used. Further, the Gabor responses were down sampled by a factor of 16 to reduce the dimensionality and then normalized to unit length. Their system achieved a performance of 95.5% for six upper and six lower AUs which was comparable to the results obtained by ICA in the same study. In their follow-up work, Bartlett et al. (9) developed a recognition system for spontaneous facial actions, which include posed expressions, and out-of-plane head movements. They achieved a classification rate of 90.6% for AU 1+2 (brow raise) and 75.0% for AU4 (brow lower) using SVMs and HMMs. Littlewort et al. (80) focused on detection of genuine smiles using normalized Gabor filters and linear SVMs. Their system recognized 75% of the test images correctly whereas only 60% of the smiles were detected correctly by human subjects on the same test images. Following (9), Smith et al. (108) developed their FACS AU recognition system using neural networks as their classifiers. Applying Gabor filters with eight different orientations and five spatial frequencies on the difference images of the whole face region, they obtained mean and joint recognition rates of 98%



and 93% respectively for six upper face AUs.

#### 4.5.5 Gabor Responses at Fiducial Points



**Figure 4.14:** Gabor wavelet representation of a facial image as shown in Lyons et al. (82)

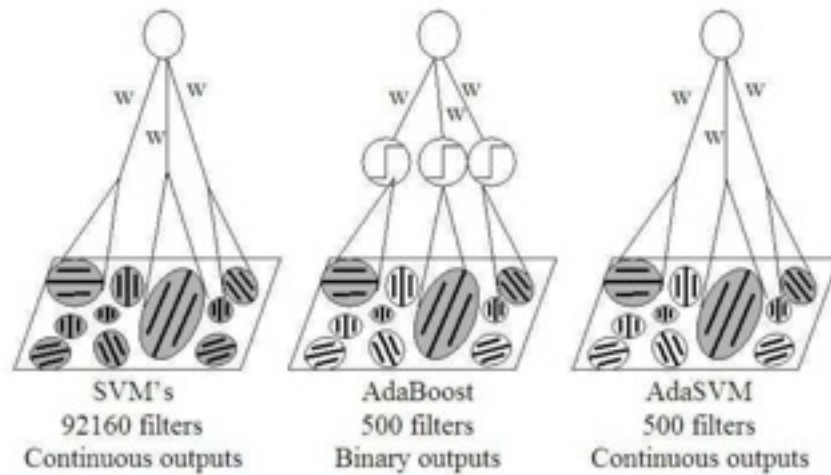
Gabor filters can also be used to compute Gabor responses at fiducial points. Zhang et al. (125) and Zhang (124) were the first ones to use this technique for prototypic expression recognition. Gabor responses were calculated at 34 fiducial points using three different frequencies and six different orientations. Classification was performed using a multilayer perceptron with seven hidden units. They achieved an overall recognition rate of 90% for the six prototypical expressions. Lyons et al. (82) developed a FER system which collects the Gabor responses at the grid points (shown in Fig. 4.14) and these Gabor vectors were further subjected to PCA analysis as a dimensionality reduction step. The responses were then classified using LDA. By empirically choosing the eigenvectors to optimize the classification performance, they achieved a recognition rate of 92%.

Computing Gabor responses at fiducial points for the recognition of FACS AUs, Tian et al. (110) developed a model to recognize three eye states from

image sequences. Using Gabor jets with three spatial frequencies and six different orientations they computed the Gabor responses at three fiducial points i.e. inner corner, outer corner and center of the eye. They achieved an average recognition rate of 83% for the three AUs 41, 42 and 43. In later studies, Tian et al. (112) developed a model which computes the Gabor responses at 20 fiducial points concentrated around eye, brows and forehead for the recognition of nine upper face AU and AU combinations with head motion and non-homogeneous subjects. For these image sequences of increasing complexity, they reported an average recognition rate of 32%.

In Bashyal and Venayagamoorthy (14), a set of 34 fiducial points were located for the classification of seven different facial expressions. Gabor filter based feature vector extraction was used in combination with learning vector quantization (LVQ). The results in (14) indicated that LVQ performs better than a multilayer perceptron in recognizing fear expression. They achieved a recognition rate of 90.22% for classifying seven different facial expressions.

### 4.5.6 Gabor Responses at Learned Locations



**Figure 4.15:** The Gabor, Adaboost and AdaSVM used in Littlewort et al. (81)

In addition to the computation of Gabor responses over entire image regions

and/or at fiducial points, many researchers adopted a new approach which combines the advantages of the above mentioned methods. This technique involves the selection of a subset of Gabor responses from learned locations, frequencies and orientations from the Gabor responses over the entire image region. Littlewort et al. (81) employed two approaches for the selection of Gabor responses; in the first approach they classified the Gabor responses using PCA feature selection and in another they used the AdaBoost algorithm (shown in Fig. 4.15). They also compared the performance of these feature selection techniques on LDA and SVM classifiers. They reported a recognition rate of 93.3% for the combination of AdaBoost and SVM known as AdaSVMs for the seven prototypical expressions. However an interesting point to note was the incompatibility of PCA with SVMs where the performance was reduced drastically when compared to the classification performance without any feature selection. This was not the case when using LDA as a classifier.

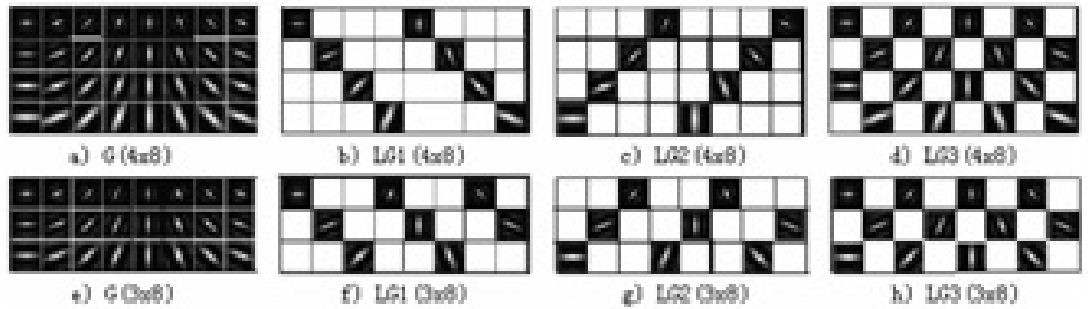
### 4.5.7 Configuration of Gabor Jets

The number and combination of scales and orientations used for a Gabor jet plays an important role in the task that is on hand. In FER, researchers have used different sets of scales and orientations to form a Gabor jet. The most widely used combinations are 3x8, 5x8 and 8x9. However, few studies were designed to verify the combination that works best in FER. While most studies for expression and FACS AU recognition used eight spatial orientations spaced at radians (8), (37), (104) there is no optimal limit on the number of frequencies used. Few studies e.g. (36) suggested that for FER, a combination of five spatial frequencies and eight different orientations works best.

### 4.5.8 Importance of Scales

In the last decade, more emphasis has been laid on the importance of different scales and their effect on the overall performance. Many studies have been aimed at investigating which frequency scales work better for FER and FACS AU recognition e.g. (36), (112). Selection of a few scales for FACS AU recognition is not only to investigate the effect of different scales on recognition performance, but

is also a step in reducing the dimensionality of the Gabor coefficients. Donato et al. (37) tested the importance of different scales by dividing the five spatial frequencies into two sets, one having the higher three spatial frequencies and the other having the lower three spatial frequencies. They obtained a recognition rate of 92.8% using the higher three spatial frequencies and a performance of 83.8% by the use of lower three spatial frequencies for the classification of six upper face AUs and six lower face AUs. Their study suggested that the higher spatial frequency bands of the Gabor filter representation contain more information than the lower frequencies. The results by Donato et al. (37) also indicated that the use of all five frequencies works better than the use of subsets alone. However, this was not the case for the results reported by Tian et al. (112). Their results indicated that the subset of spatial frequencies perform better than the use of all the spatial frequencies, which they attributed to the reduction in the redundant data. They divided the five spatial frequencies into three subsets, containing first three frequencies, middle three frequencies and last three frequencies. Their study indicated that the middle three frequencies perform better for the upper face AUs and the last three spatial frequencies perform better for the lower face AUs. There is no standard to know which set of frequency components work better for the classification of FACS AUs.



**Figure 4.16:** Examples of local Gabor filters proposed by Deng et al. (36)

However, one study by Deng et al. (36) proposed a new approach based on local Gabor filters (shown in Fig. 4.16); instead of selecting a subset of spatial frequencies, using the coefficients from all the frequency scales is also efficient in

dimensionality reduction. They achieved an average recognition rate of 97.33% using Gabor responses at five spatial frequencies and eight different orientations using nearest neighbor classifier with distance measure Euclidean.

### 4.5.9 Feature Selection Techniques for Gabor Coefficients

Owing to the huge dimensionality of the data generated by Gabor filters, feature selection techniques have gained importance where Gabor coefficients are used as features. The most widely used methods in FER literature are PCA and LDA. This section reviews the mentioned feature selection methods and their affect on performance of the FER systems.

Principle component analysis has been one of most widely used feature selection methods in real world applications. In FER, PCA has been used as a feature selection technique in conjunction with Gabor coefficients to reduce the dimensionality of the data. Bartlett et al. (9) and Littlewort et al. (81) used PCA as a feature selection technique on the Gabor coefficients. Using a LDA classifier, they achieved 44.4% accuracy without any feature selection techniques in place, whereas applying PCA before expression classification by the LDA classifier, the recognition rate increased to 80.7% for six basic emotions. However, similar studies performed with linear SVM as a classifier indicated that the use of PCA as a feature selection technique degraded the performance from 88% to 75.5% for the classification of six basic emotions. They attributed this reduction to the incompatibility of PCA with SVM for the particular task. Tian et al. (112) have used PCA as a feature selection technique to reduce the dimensionality of the data derived from few fiducial points, which failed to increase the performance of the system using neural networks. Deng et al. (36) also used PCA as a feature selection technique to classifying expressions. Their results indicated that nearest neighbor classifier using Euclidean (L2) performs better than Cityblock (L1) distance measure in conjunction with PCA. No results were reported comparing the performance obtained with PCA and without PCA. The performance of PCA differs from one instance to another. Its performance not only depends on the application that it is used for, but also the classifier, the size of training data and the complexity (9).

## **4.6 Comparison of Appearance Based and Geometry Based Methods**

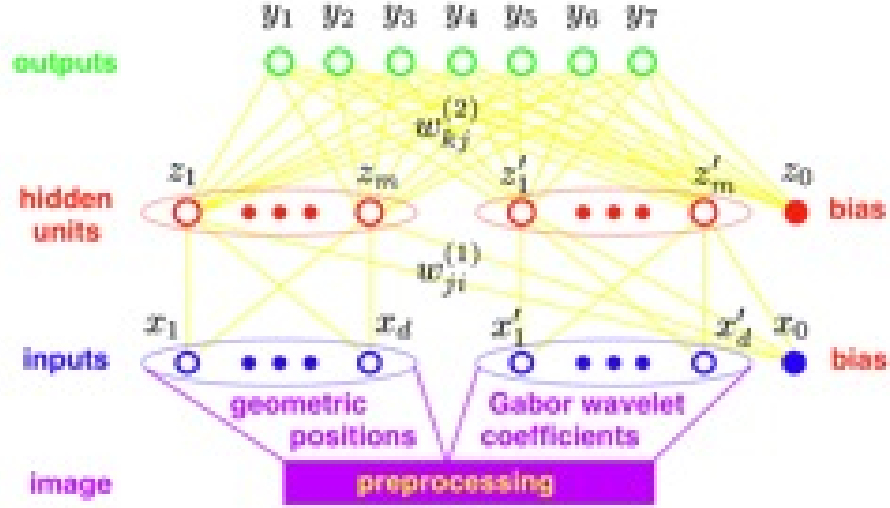
Linear discriminate analysis was also used in conjunction with PCA as a feature selection technique. In most real world applications, it is not possible to have enough number of training samples which equate to the dimensions of the feature. In order to solve this problem Deng et al. (36) used PCA+LDA for feature selection. The features are first selected from the Gabor responses using PCA, and the selected features are processed using LDA to reduce the dimensionality further. In (36), the results indicated that the use of PCA+LDA feature selection method performs better than the use of PCA alone. They achieved highest recognition rate of 97.33% for the classification of facial expressions using nearest neighbor classifier with Euclidean (L2) distance measure.

## **4.6 Comparison of Appearance Based and Geometry Based Methods**

Recently, FER studies have compared the performance of appearance based to geometric based methods. In this section we study models based on their use.

Zhang (124) and Zhang et al. (125) (shown in Fig. 4.17) compared the classification performance of appearance-based method using Gabor wavelets and geometric based methods for the recognition of prototypical expressions. The image database was homogeneous containing only frontal faces. For Gabor responses, they use three spatial frequencies and six different orientations and for geometric representation they used 34 fiducial points distributed on the face. They employed a neural network as a classifier for both methods. Their results show that the Gabor based method clearly outperformed the geometric method with a nearly 30% higher recognition rate with 20 hidden units.

However, the high performance of Gabor responses claimed by Zhang et al. (125) was disputed by Tian et al. (111). Similar experiments by Tian et al. (111) on a more heterogeneous database with head movements for the recognition of FACS AUs showed that geometric based approach outperforms Gabor based approach. When classifying AUs with complex combinations with other AUs, they achieved a recognition rate of 87.6% whereas the recognition rate fell to 32% using Gabor based method. However, they did not perform any pre-processing



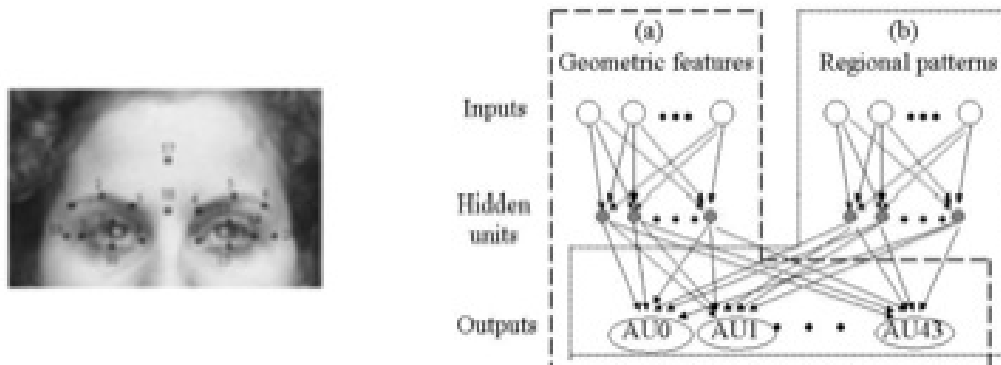
**Figure 4.17:** System architecture used in Zhang et al. (125)

step prior to Gabor based method and the responses were also not measured on the entire face, which was reported to increase the performance of Gabor based methods (36).

## 4.7 Hybrid Systems

To take advantage of both the appearance and geometry-based methods, systems have been developed combining both methods. The model developed by Zhang (124) classified seven prototypical expressions combining Gabor responses and the geometric positions calculated at 34 fiducial points. Using a neural network with seven hidden neurons, they achieved recognition rates of 73.3% using geometric approach alone, 92.2% using Gabor coefficients alone and a recognition rate of 92.3% when combined. However, their system did not show any significant performance increase when combined features were trained compared to training only the Gabor coefficients.

A similar approach by Tian et al. (112) for the recognition of FACS AUs using 20 fiducial points (shown in Fig. 4.18) in the upper half of the face revealed a



**Figure 4.18:** 20 fiducial points in the upper half of the face region and the neural network architecture used in Tian et al. (112)

different behavior. They combined the Gabor coefficients and geometric positions calculated at 20 fiducial points located near the eyes, brows and forehead. Using a neural network (shown in Fig. 4.18) for the recognition of nine upper face FACS AUs, they achieved a recognition rate of 87.6% using geometric features alone, 32% using Gabor based approach and 92.7% for a combined approach.

## 4.8 Image Sequences with Increasing AU Complexity

Recently, more emphasis has been placed on the use of image sequences for the recognition of facial expressions and FACS AUs instead of single frame images. One common approach is to calculate the difference images where the initial frame (neutral image) is subtracted from the other images in the sequence. Donato et al. (37) developed their model using image sequences that contained six frames. In the preprocessing step, the eye and mouth coordinates were used to align and scale the faces. To control the variations in light, logistic filters were used and the Gabor filters were applied on the difference image. Using cosine similarity method and nearest neighbor classifier, they obtained a recognition rate of 95.5%. However, their image sequences contained only the requested action with no rigid



head motions. In Tian et al. (112) a model was developed using images sequences with increasing complexity containing single AU, AU combinations and head motions. Using a neural network, they obtained a recognition rate of 32% using Gabor coefficients computed at 20 fiducial points. They attributed the decrease in the performance to the use of heterogeneous database with no preprocessing steps involved. Following (10), Smith et al. (108) developed their FACS AU recognition model. They used neural networks as their classifiers. Applying Gabor filters with eight different orientations and five spatial frequencies on the difference images of the whole face region, they obtained mean and joint recognition rates of 98% and 93%, respectively, for six upper face AUs.

## 4.9 Classification

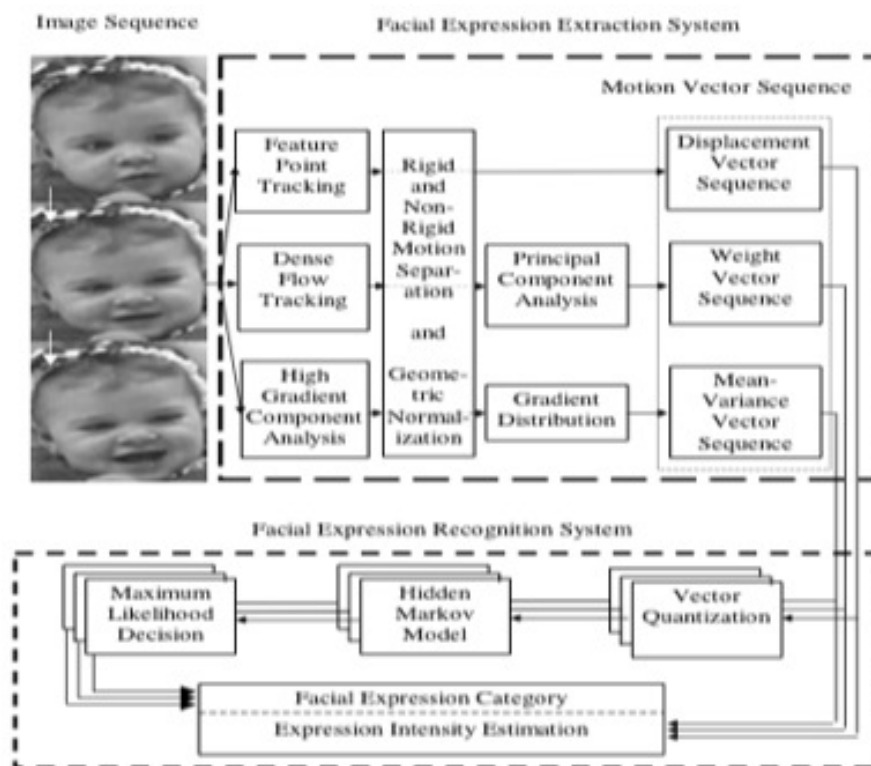
Classification is the last stage in facial expression analysis. As discussed in Section 4.2.5, the classification stage can be used to classify actions based on judgment or sign based systems. Facial expression analysis can also be modeled to interpret the judgment based basic emotions using sign-based frameworks. A different approach towards classification is the use of spatial or spatio-temporal approach. In this section we review some of the literature based on this distinction.

### 4.9.1 Spatial Approach

In the literature spatial approach often used neural networks, SVMs and LDA for facial expressions classification (31), (112), (81). Single images were directly used or a set of feature extraction techniques such as PCA (37), Gabor filters (81) were applied before feeding the data to the above mentioned classifiers. These models however, only depended on the peak intensities depicted in the frame used for classification. The knowledge regarding the formation of a particular action is absent.

### 4.9.2 Spatio-temporal Approach

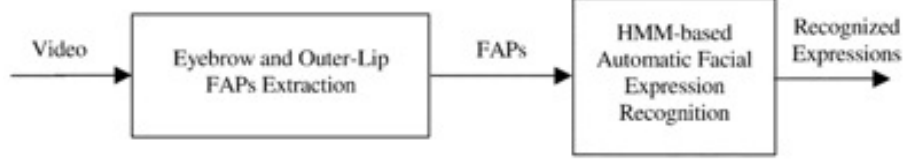
The use of spatio-temporal approach in conjunction with HMMs and RNNs has been studied in the field of speech recognition and gesture recognition. Spatio-



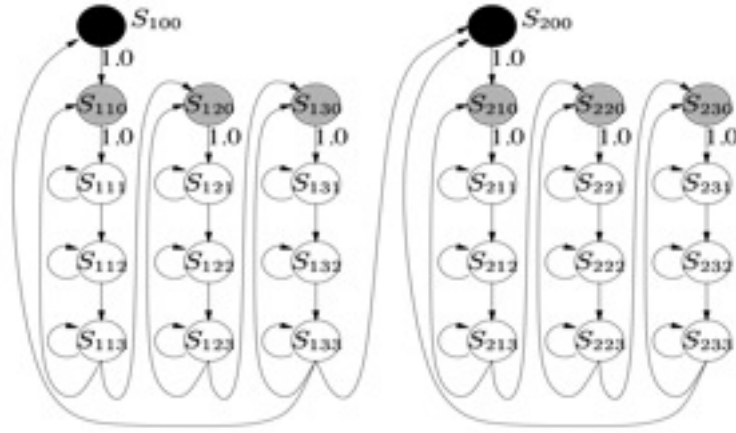
**Figure 4.19:** Block Diagram of the recognition system used in Lien 1998 (76)

temporal approach using HMMs and RNNs has also gained interest in the field of FER. Some of the models using HMMs for FER (76), (92), (28), (90) and (95) are found in the literature. In (76) and (77) an automatic FER model using HMMs was developed. The block diagram of the recognition model used in (76) is shown in Fig. 4.19. The features from an image sequence were extracted using a high gradient component analysis, which upon further processing, were fed to a HMM for the recognition of nine facial expression units based on FACS.

Peter and Aggelos (95) used HMMs to classify the FAPs of MPEG-4 standard as shown in Fig. 4.20. The FAPs they employed describe the movements of the outer-lip contours and eyebrows. These FAPs were then used to model six basic facial expressions. The classification model which was based on the use of HMM. Muller et al. (88) also used a HMM model for the classification of four basic expressions; anger, surprise, disgust and happiness. The HMM structure used by



**Figure 4.20:** Block Diagram of the HMM based FER system used in (95)

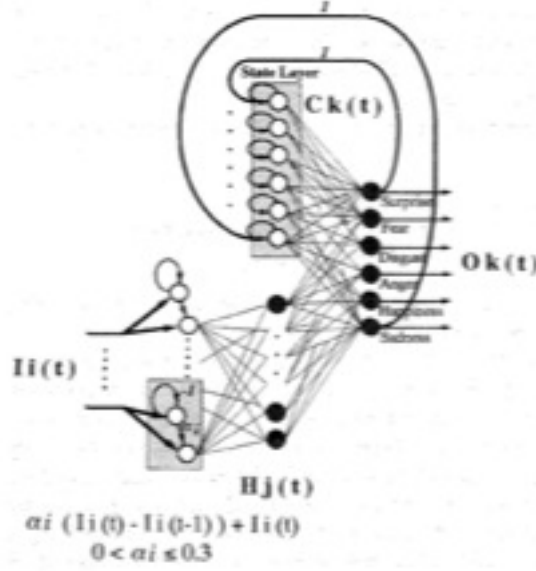


**Figure 4.21:** HMM structure capable of modeling whole image sequence used in (88)

them to model the four basic expressions is given in Fig. 4.21.

Recurrent neural networks were also used in place of HMM for the classification of facial expressions (68), (102), (58), (25), (57). Kobayashi and Hara (68) used a discrete-time recurrent neural network as shown in Fig. 4.22 to recognize dynamic facial expressions. They compared the recognition results of the facial expressions by the use of RNN with that of human beings. Their results indicated that RNNs based dynamic recognition of facial expressions is capable of showing similar characteristics as those seen in the dynamic recognition of facial expressions by humans.

In Rosenblum et al. (102) some basic enhancements were made to the Radial basis function network (RBFN) to handle the temporal relations associated with the use of image sequences. They used six such individual networks to train six



**Figure 4.22:** Discrete-time recurrent neural network used in (68)

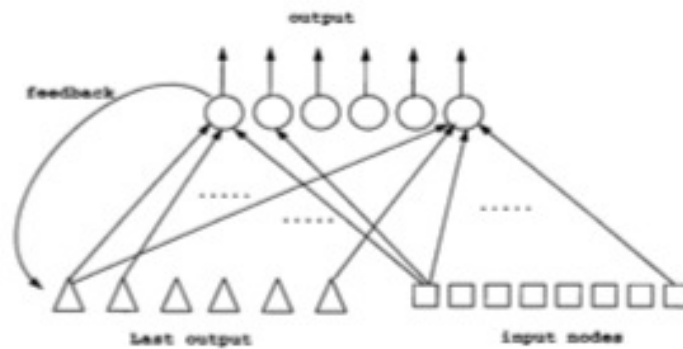
basic emotions such as happiness, disgust, anger, sadness, surprise and fear (as shown in Fig. 4.23). Using an absolute mode where all the network outputs were compared, a winner is picked. Their experiments suggested that the emotions which depicted more pronounced motion were better understood by the networks.

In Hai Tao et al. (58) the use of facial animation parameters (FAPs) of MPEG4 standard were used to train a RNN for the recognition of six emotions. In their study the emphasis was on finding a new representation of the FAPs with lower spatial dimension, which would reduce the complexity of the recognition model. In spatio-temporal pattern recognition models using HMMs and RNNs, the computational complexity and the training data size are significantly affected by the spatial dimension of the input data (58). Their compression method was able to downsize the 68 FAPs to just eight components, which were then used to train a RNN. A RNN with eight input nodes for reading the inputs and six output nodes for the recognition of the six emotions was used (as shown in Fig. 4.24).

The use of RNN for the recognition of facial expressions had also been investigated in the bi-model systems (25). Caridakis et al. (25) used a simple RNN



**Figure 4.23:** Hierarchy of networks based on decomposition of emotions used in (102)

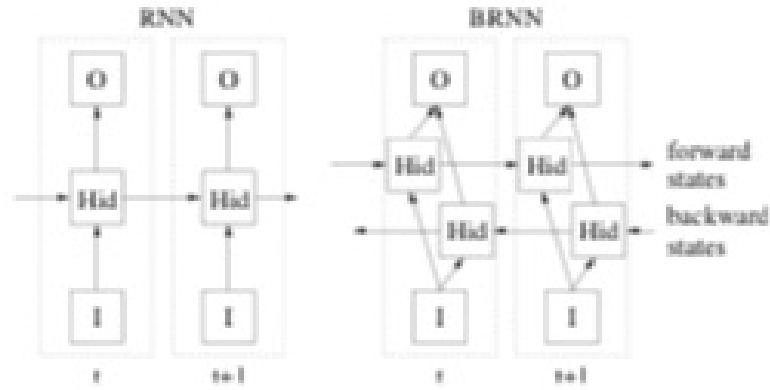


**Figure 4.24:** RNN structure used in Hai Tao et al. (58)

for modeling dynamic changes in both a subjects facial expressions and speech. Feature extraction to track the changes in the facial area involves the location and tracking of MPEG4 FAPs. These were then fed to an Elman network along with the speech features. They achieved a classification efficiency of 67% using only facial data, where as the performance raised to 79% by combining both the modalities.

In Graves et al. (57), an automatic FER system using RNNs was developed for handling temporal data. They used two different variants; unidirectional and bi-directional LSTMs for the experiments. Long short-term memory are capable

of handling information over long periods of time, thus extending the temporal context available to the network (57). A bidirectional network is able to provide both the future as well as the past context. The uni and bi-directional LSTMs used in (57) are shown in Fig. 4.25 .Their results indicated that a bi-directional network gives a significantly better performance than the use of a uni-directional LSTM.



**Figure 4.25:** Uni-directional and bi-directional networks used in (57)

## 4.10 FACS AU Recognition Using RNNs

To data, however, the use of spatio-temporal approach has not been used for FACS AU recognition. In the current work we propose the use of a recurrent approach for handling time varying data in the field of FACS AU recognition. This would provide a model for classification of subtle changes making it useful for many real world applications. We also hypothesize that our approach will perform better than the use of classic classifiers such as SVMs using single static images.

## 4.11 Conclusion

In this chapter we have discussed about various approaches for FER and the different feature types that can be used. We have discussed about appearance based

and geometry-based feature systems. Different models and their performance when using different feature types were briefed and the strengths of such systems were discussed. Systems that use both the feature types were also explained in detail. Then the importance of time depend data for FER was emphasized and the classifiers that have been used in this regard were explained. The classifiers like neural networks provide a non-recurrent approach of reading time series data, where as the classifiers such as RNNs with their ability to handle dynamic data are a good choice for FER when the input data is time dependent. Given the high performance of appearance-based methods such as Gabor filters with other classifiers, it would be interesting to investigate the use of RNNs with time series data and compare its performance with that of other systems.

## 4.12 Summary

Here in this chapter we have discussed about the automatic FER systems used in the recent times. We have focused on the two feature selection types, appearance-based and geometry-based methods. We then focused on the use of time series feature data, which would help the system to learn from the previous time steps. Reccurent neural networks have been successfully used in speech recognition, signature verification and forecasting for handling time variant data. In this thesis we proposed the use of RNNs for FACS AU recognition.

# Chapter 5

## Feature Extraction Techniques

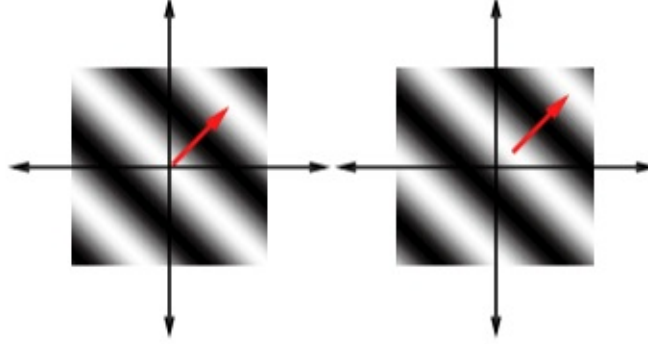
### 5.1 Introduction

In the field of FER, successful classification is achieved by the use of good feature extraction techniques. The literature shows that the use of a particular feature extraction either improves the models performance or deteriorates by a great degree. The ability of an extraction technique towards providing a better performance also depends on the classification model used, the structure of data, the presence of noise and the extraction techniques ability to handle the presence of noise. The classical feature extraction techniques used over the years are PCA, LDA, Gabor filters and Haar filters etc. Some of these techniques such as PCA has also been successfully used for feature reduction. Here in this chapter we review the above techniques as this forms a crucial part of this thesis.

### 5.2 Gabor Filters

Gabor filter, named after Dennies Gabor, is a linear filter used in for edge detection. They have been one of the widely used feature extraction techniques for expression recognition. The work by (37) also suggested that Gabor filters provided the best performance for expression recognition. A tutorial by Movellan (87) provides a detailed overview of both 1-D and 2D Gabor filters. In this section we review only a 2D Gabor filter.





**Figure 5.1:** The real and imaginary parts of a complex sinusoid with  $u_0 = v_0 = 1/80$  cycles/pixel and  $P = 0$  deg. as shown in (87)

A 2D Gabor filter in spatial domain is given as the product of a complex sinusoid function and a 2D Gaussian function (87). Owing to the convolution theorem, the Fourier transform of a Gabor filter impulse response is the convolution of the Fourier transform of the sinusoid function and the Fourier transform of the Gaussian function. The complex Gabor function in space domain is given as

$$g(x, y) = s(x, y)w_r(x, y) \quad (5.1)$$

where  $s(x, y)$  is a complex sinusoid and  $w_r(x, y)$  is a 2D Gaussian shaped function. The complex sinusoid function is known as the *carrier* and the 2D Gaussian function is known as the *envelop*.

The complex sinusoid is given as

$$s(x, y) = \exp(j(2\pi(u_0x + v_0y) + P)) \quad (5.2)$$

where  $(u_0 + v_0)$  and  $P$  define the spatial frequency and phase of the sinusoid respectively. This function is further given as two separate real functions, located in the real and imaginary of a complex function (shown in Fig. 5.1).

The real and imaginary part of this sinusoid function is given as

$$\text{Re}(s(x, y)) = \cos(2\pi(u_0x + v_0y) + P) \quad (5.3)$$

$$\text{Im}(s(x, y)) = \sin(2\pi(u_0x + v_0y) + P) \quad (5.4)$$

The parameters  $u_0$  and  $v_0$  define the spatial frequency of the sinusoid in Cartesian coordinates. The spatial frequency in polar coordinates is given as magnitude  $F_0$  and direction  $w_0$ :

$$F_0 = \sqrt{u_0^2 + v_0^2} \quad (5.5)$$

$$w_0 = \tan^{-1}\left(\frac{u_0}{v_0}\right) \quad (5.6)$$

*i.e.*

$$u_0 = F_0 \cos w_0 \quad (5.7)$$

$$v_0 = F_0 \sin w_0 \quad (5.8)$$

Using this representation the complex sinusoid is given as

$$s(x, y) = \exp(j(2\pi F_0(x \cos w_0 + y \sin w_0) + P)) \quad (5.9)$$

The Gaussian function is given as

$$w_r(x, y) = K \exp(-\pi(a^2(x - x_0)_r^2 + b^2(y - y_0)_r^2)) \quad (5.10)$$

where  $(x_0, y_0)$  is the peak of the function.  $a$  and  $b$  are the scaling parameters of the Gaussian and the subscript  $r$  stands for a rotation operation which is

$$(x - x_0)_r = (x - x_0) \cos \theta + (y - y_0) \sin \theta \quad (5.11)$$

$$(y - y_0)_r = -(x - x_0) \sin \theta + (y - y_0) \cos \theta \quad (5.12)$$

The complex Gabor function in space domain is thus given as

$$g(x, y) = K \exp(-\pi(a^2(x - x_0)_r^2 + b^2(y - y_0)_r^2)) \exp(j(2\pi(u_0x + v_0y) + P)) \quad (5.13)$$

or in the polar coordinates

$$g(x, y) = K \exp(-\pi(a^2(x - x_0)^2 + b^2(y - y_0)^2)) \exp(j(2\pi F_0(x \cos w_0 + y \sin w_0) + P)) \quad (5.14)$$

## 5.3 Principle Component Analysis

Principle component analysis has been widely used for feature extraction and selection. In FER, dimensionality of data plays a vital role and PCA has been successfully used towards reducing the dimensionality of input data.

Consider a set of  $N$  images  $x_1, x_2, \dots, x_N$  represented by a  $p$ -dimensional Gabor feature vector. Principle component analysis can then be used to find the linear projections mapping the original  $p$ -dimensional data onto a  $q$ -dimensional feature space, where  $q \ll p$  for most real time applications. The new feature vector  $y_i \in R^f$  is defined as

$$y_i = W_{pca}^T x_i \quad (5.15)$$

where  $i = 1, 2, \dots, N$  is the number of sample images and  $W_{pca}$  is the linear transformation matrix.

$W_{pca}$  is a  $r$ -dimensional column matrix where  $r$  is number of eigenvectors associated with the  $r$  largest Eigen values of the scatter matrix  $S_T$  defined as

$$\sum_{i=1}^N (x_i - x_m)(x_i - x_m)^T \quad (5.16)$$

where  $x_m$  is the mean image of all the sample images. One of the disadvantages of PCA is that, it may lose important information for discrimination between classes.

## 5.4 Linear Discriminant Analysis

Linear discriminant analysis and the related Fisher's linear discriminant (FLD) are the methods to find a linear combination of features which characterize two

or more classes of events. This also makes them useful as a linear classifier for dimensionality reduction.

Consider a set of feature  $x$  for each sample of an event with known class  $y$ . This set of samples constitutes the training set. The classification problem is to find a good predictor for the class  $y$  of any sample of the same distribution given only a feature  $x$ . Using LDA for this problem one assumes that the conditional probability density function  $p(x|y = 0)$  and  $p(x|y = 1)$  are both normally distributed with mean and covariance parameters  $(\mu, \sum_{y=0})$  and  $(\mu, \sum_{y=1})$  respectively. Based on this assumption, the Bayes optimal solution is to predict points as being from the second class if the ratio of the log likelihood is below some threshold  $T$ .

## 5.5 AdaBoost

Adaboost is a machine learning algorithm formulated by Freund and Schapire (53). Adaboost is less susceptible to the problem of over-fitting face by other classifiers. However, it is sensitive to noisy data.

Adaboost calls a weak classifier repeatedly in a series of rounds  $t - 1, t, t + 1, \dots, T$  from a total of  $T$  classifiers. For each call a distribution of weights  $D_t$  is updated that indicates the importance of examples in the dataset for the classification [Refs]. In each round, the weights of each incorrectly classified observation are increased, so that the new classifier focuses on those observations

In this section we will be reviewing Adaboost as a feature selection algorithm. The following algorithm presents Adaboost as a feature selection technique.

- Given the training set  $(x_1, y_1), \dots, (x_n, y_n)$  where  $x_i$  is the data of the  $i^{th}$  example, and  $y_i = 0, 1$  for imposters and clients respectively.
- Initialize weights  $w_{i,j} = \frac{1}{2m}, \frac{1}{2n}$  for  $y_i = 0, 1$  respectively, where  $m$  and  $n$  are the number of imposters and clients respectively.
- For  $t = 1, 2, \dots, T$ :
  - Normalize the weights  $w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{i=1}^m w_{t,i}}$  so that  $w_t$  is a probability distribution.

- For each feature  $j$ , train a classifier  $h_j$  which uses a single feature. The error is evaluated with respect to  $w_t, \varepsilon_j = \sum_{i,t} w_{t,i} |h_j(x_i) - y_i|^2$
- Choose the classifier  $h_t$ , with the lowest error  $\varepsilon_t$ .
- Update the weights:  
If there is only one lowest error,  
 $w_{t+1,i} = w_{t,i} \beta_t^{1-e_i}$   
where  $e_i = 0$  if example  $x_i$  is classified correctly,  $e_i = 1$  otherwise, and  
 $\beta_t = \frac{\varepsilon_t}{1-\varepsilon_t}$

## 5.6 Haar Filters

Haar filters are an alternative to Gabor filters which have proven to be successful for facial expression analysis. However, Haar filters don not face the disadvantage of huge dimensionality face by Gabor filters. Haar filters are based approximately of the Haar wavelet decomposition.

A one dimensional Haar wavelet decomposition of an array of size  $n$  is computed recursively using averaging and differencing. In the averaging stage input array is reduced in length by half by averaging the value of every pair of neighboring values. In this process information is lost. The lost information is recovered by appending  $\frac{n}{2}$  detail coefficients to the output array during the differencing stage. Each detail coefficient  $d$  is the amount by which the first element in the averaged pair exceeds that pair's average. This appending of detail coefficients resize the length of the array to  $n$ . The process of averaging and differencing are then repeated on the first half of the array. The next level of recursion will consist of averaging first quarter of the array. The recursive process is completed after  $\log n$  levels when only one pair of numbers is averaged. Here we illustrate the process using a generic array of 4 elements  $[x_1, x_2, x_3, x_4]$ . Upon averaging and differencing during iteration 1, we get:

$$\left[ \frac{x_1 + x_2}{2}, \frac{x_3 + x_4}{2}, x_1 - \frac{x_1 + x_2}{2}, x_3 - \frac{x_3 + x_4}{2} \right] \quad (5.17)$$

After 2nd iteration of averaging and differencing, we get:

$$\left[\frac{x_1 + x_2 + x_3 + x_4}{4}, \frac{x_1 + x_2 - (x_3 + x_4)}{4}, \frac{x_1 - x_2}{2}, \frac{x_3 - x_4}{2}\right] \quad (5.18)$$

A two dimensional Haar decomposition can be obtained by generalizing the one dimensional Haar decomposition using two methods known as standard and non-standard decomposition. In standard decomposition, the transform is first applied to each row of the input matrix and then it is applied to each column. In non-standard decomposition, the transform is alternately applied to rows and columns at each recursive level of the transform. The transform then proceeds again on the rows at the next recursive level.

Assume a square image of size  $n^2$  pixels. The two dimensional Haar decomposition of this image will consist of  $n^2$  distinct Haar wavelet coefficients. The first wavelet will be the mean pixel intensity value of the whole image. The other wavelets represent the difference in the mean intensity values of the horizontally, vertically or diagonally adjacent squares consisting of black and white regions. The Haar coefficient of a particular Haar wavelet is then computed as the difference in average pixel value between the image pixels in the black and white regions.

A two dimensional Haar decomposition of an image with  $n^2$  pixels contains exactly  $n^2$  coefficients and is thus said to be exactly complete. For each Haar wavelet the  $(x, y)$  locations, its width and height which are a power of 2, acts as constraints to the wavelet. These constraints however, can be relaxed based on the application. Haar features can be computed using few CPU instructions using the “integral image” technique demonstrated on (114).

## 5.7 Summary

This chapter is aimed at reviewing the basic algorithms for the most widely used feature extraction and feature selection techniques. The use of feature extraction and reduction techniques is widely used in our work. Thus it becomes important to review the basic algorithms of these techniques. We review the most commonly used PCA, LDA, Gabor filters, Haar filters and AdaBoost. Some of these algorithms are used for both feature extraction and feature selection.

# Chapter 6

## Modeling

### 6.1 Introduction

This chapter will concentrate on the basics of the classifiers we intend to use in our work. Section 6.2 focuses on neural networks, their learning algorithms and some applications of the same. This will act as our base for a detailed overview of RNNs, which will be discussed in Section 6.4.

Apart from neural networks, SVMs have also gathered considerable attention in recent years. The basic principle of SVM is to maximize the distance between two classes of data points in the input space. Support vector machines enjoy several advantages like: handling high dimensional data without any considerable affect on the training time, power and flexibility offered through the use of kernel trick. In the field of FER, this capability of SVMs proved to be very helpful in handling high dimensional data generated with feature extraction techniques like Gabor filters (11). Section 6.3 focuses on SVMs.

Neural networks and SVMs are classic classifiers which are capable of handling data without any capability of handling time variant sequences commonly known as spatial classifiers. In real world applications, however, the data is rarely present in one single instance. One needs classifiers which are capable of handling data that varies over a period of time. This format of data not only helps in gaining more information but also the changes that take place over a period of time. Section 6.4 and Section 6.5 focuses on classifiers like RNNs and HMMs which are

capable of handling time varying sequences. At the end, section 6.6 focuses on LSTMs which are capable of handling large sequences of data over time.

## 6.2 Artificial Neural Networks

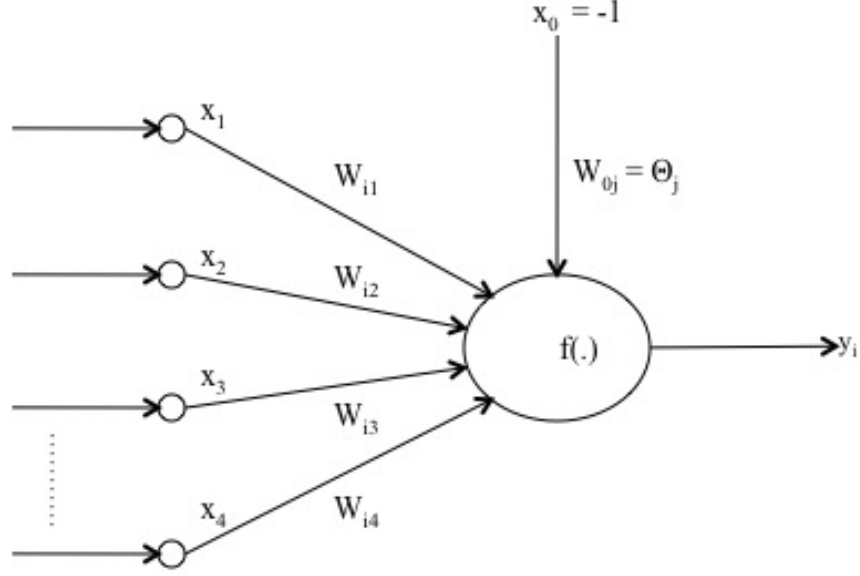
One of the popular data-driven techniques used by various researchers in machine learning is an Artificial Neural Network (ANN). An ANN roughly replicates the behavior of the human brain when processing the information. Each neuron in the network acts as a biological neuron, which helps an ANN to reason like a human. A neuron is a single processing unit that accepts the input from other neurons or an external source and computes their weighted sum. Depending on the location in the network a neuron can act as an input, hidden or an output neuron. The neural network trains itself with the data that is available in the training set and tests on the test set. Ideally both the training and test sets are taken from the same group of the information but are disjoint. The network learns by updating the weights associated with the neurons so that the weighted sum of the inputs is converging towards the known output. Once the training is performed, the verification is very fast. The reliability, robustness and the classifying capability of the ANNs depend on the source, range, quantity and quality of the data. ANNs have been applied to many real world problems like speech recognition and gesture recognition.

An ANN acts like a human brain. The cell membrane receives the incoming information (impulses) via dendrites that typically acts like a receiver. If the number of incoming impulses exceeds a certain threshold value a neuron will send the information to the other neurons through its synapses. Threshold value determines the impulse frequency at which a neuron fires off.

### 6.2.1 Processing Unit

In a typical ANN, a node has synaptic weights connecting to the input nodes, a summation function associated with the node and a transfer function. Fig. 6.1 depicts the structure of a single neuron.





**Figure 6.1:** Structure of a single neuron

Synaptic weights are characterized by their value (strength) that corresponds to the importance of the input information to the neuron. The summation function calculates the weighted sum of the inputs and the transfer function, which is also known as the activation function, and transforms the summed input (from the summation function) into output. There are 2 kinds of transfer function: linear and non-linear.

*Linear Function:* A Linear function is described as

$$f(y_i) = y_i \quad (6.1)$$

where the output of the node replicates its input.

*Non-Linear Function:* A typical example of a non-linear function is a sigmoid function, which is described as follows

$$f(y_i) = \frac{1}{(1 + e^{-y_i})} \quad (6.2)$$

Another example of a non-linear transfer functions is Tan-Hyperbolic functions  $\tanh(y_i)$ .

From Fig. 6.1, the weighted sum of the inputs to the neuron  $i$  is given as,  
Weighted sum of the neuron  $i$ =

$$i = \sum_{j=1}^n w_{ij}x_j \quad (6.3)$$

where  $n$  represents the number of inputs to the neuron.

The net input activation value of neuron  $i$ , which has  $N$  number of input impulses is given as

$$y_i = \sum_{j=1}^n w_{ij}x_j - \theta_{0j} \quad (6.4)$$

where  $\theta_{0j}$  represents the bias. The transfer function  $f(y_i)$ , which can be either a linear or a non-linear function, then computes the output  $O_j$  of the neuron  $i$  as

$$O_i = f(y_i) \quad (6.5)$$

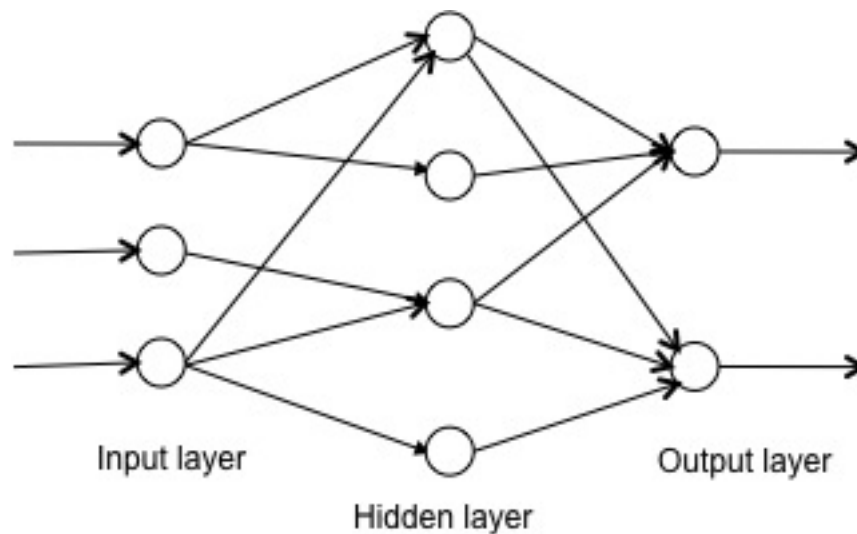
### 6.2.2 Multilayer Perceptron

Neural networks are structured in the form of layers of neurons. Every network consists of an input layer and an output layer. But in most of real time problems there arises a need of the hidden layer, which adds additional complexity to the network. Thus, the networks are divided into two types depending on the type of layers present: single layer networks which contain no hidden layers and multi layer networks which have one or more hidden layers.

Training of neural networks forms one of the other important aspects to look at. Training of neural networks can be categorized into two types: supervised learning and unsupervised learning. Supervised learning is a machine learning technique for learning from a given set of training data. The training set consists of pairs of input vectors and desired output vectors. The task of the supervised learning is to predict the value of the function for any valid input after having seen a number of training examples. To accomplish this task, the learning algorithm has to generalize from the present data to unseen samples in a reasonable way. Unsupervised learning is the other machine learning technique where one seeks

to determine how the data is organized. It is different from supervised learning in that the learning algorithm is given only the input vectors.

The next important aspect is the connection pattern between the different layers present in the networks. Networks are classified into two groups depending on the connection patterns: feed-forward neural network and feedback (recurrent) neural network. Feed-forward networks contain only open loop interconnections between the layers. Each of the nodes in the hidden and output layers have connections with their corresponding weights to each node of the preceding layer. Fig. 6.2 depicts the data flow of a feed-forward network. The input vector given to the network is passed to the input layer without any computation. The hidden layers, if any, provide the additional computation and then the output layer generates the output.



**Figure 6.2:** Network Topology of a Feed-forward Network

Feed-forward networks have applications in solving time invariant problems. Some of its applications include gesture recognition, signature verification, face and expression recognition. The most popular types of feed-forward networks are error back propagation (EBP) and radial-basis function (RBF) networks.

Error back propagation networks have wide range of applications because of their robustness. The task of an RBF network is that of supervised learning,

where given a set of input vectors and their associated target vectors, the network tries to learn the functional relationship between the input and the target vectors. These networks operate in two modes: mapping and learning. In mapping mode each input vector is analyzed and then the network estimates the outputs. This output from the output node is compared with the target output associated with the input vector and the difference (error) is propagated into the network to adjust the weight vector which is the learning mode. Error back propagation networks use nodes with sigmoid transfer functions. Sigmoid transfer functions provide a bounded output.

The dynamics of the EBP network are as follows: Given the input vector  $X = (x_1, x_2, \dots, x_n)$ , the output at the hidden node is given as:

$$y_i = f\left(\sum x_j w_{ji} + \theta_{0j}\right) \quad (6.6)$$

where  $j = 1..N$ ,  $N$  being the number of inputs.  $w_{ij}$  is the weight from node  $j$  to node  $i$ . The hidden node output will be the input to the output nodes if there is only one hidden layer. The output of the output node is calculated as:

$$O_i = f\left(\sum y_j w_{ij} + \theta_{0j}\right) \quad (6.7)$$

where  $j = 1..K$ ,  $K$  being the number of output units. Here the transfer function is a sigmoid function which bounds the output values in the range of  $[0,1]$ .

The mean square error is then calculated as:

$$E = \frac{\sum \sum (O_{input} - t_{output})^2}{2MN} \quad (6.8)$$

where  $N$  is the number of examples in the training set and  $M$  is the number of outputs in the network.  $t_{input}$  is the target of the  $i^{th}$  target output for the  $n^{th}$  example.

In learning mode, the network is optimized by decreasing the mean square error so that  $E$  is minimum. By knowing the slope of the error surfaces weights are adjusted after every iteration according to gradient descent. The weight is updated as:

$$\Delta w(t) = -n \frac{\partial E}{\partial w} + \mu \Delta w(t-1) \quad (6.9)$$

where  $n$  is the learning rate and  $\mu$  is the momentum value.

Radial-basis function is the other type of feed-forward neural network. Typically an RBF consists of one input layer, one hidden layer and one output layer. Here the number of hidden layers can not be more than one. The networks learning rate is much higher than the other networks if the number of input variables is not too high. However, the required number of hidden units increases geometrically with the number of input units. The hidden layer nodes use Gaussian functions.

### 6.2.3 Learning Algorithm

Learning in neural networks is a task to approximate the output of the network when a set of inputs is given. This is achieved by adjusting the weights of the network according to the learning algorithm. The learning of the weights continues till a predefined criterion is met. In most applications the criteria for the network to stop learning is defined in terms of network output error, i.e., when the network error is less than the minimum error acceptable for the task at hand, the learning stops. The data available for training the network provides two possibilities of training the network. They are supervised learning and unsupervised learning.

Supervised learning is a learning process where along with the input data, the expected output is presented to the network. The weights in the network are thus adjusted such that when the specific input is given, the network output is similar to that of the desired output. However, it may not be possible to know the expected output for a specific input beforehand. In this situation, one uses unsupervised learning technique where the network is trained with the specific input data on hand. The network weights are adjusted such that the network produces the best possible output for the given input.

The goal of a learning algorithm is to optimize the network weights to produce the expected output vector when given a set of input vectors. Gradient descent is one of the simplest function optimization techniques which has been widely used

for learning in neural networks. The goal of gradient descent learning rule is to optimize the weights between nodes of two layers in the network. The objective function in this algorithm measures the difference in the desired output and the actual output given by the network for a set of input vectors.

The learning algorithm which uses gradient descent to update the weights of the single layer network is called the delta-learning rule. The delta-learning rule tries to optimize the network weights by adjusting them using gradient descent function optimization technique. The network error  $E$  with  $i$  number of output units is given as

$$\frac{1}{2}(d_i - t_i)^2 \quad (6.10)$$

where  $d_i$  is the desired output and  $t_i$  is the actual output given by the network.

The goal of the learning rule is to find the optimal weight for a neuron in the weight space in proportion to the gradient of the error function with respect to each weight. The partial derivative of the each error function with respect to each weight  $j$  is given as

$$\frac{\partial E}{\partial w_{ij}} = -(d_i - t_i)g'(o_i)x_j \quad (6.11)$$

where  $\eta$  is the learning rate.

Delta-learning rule can also be extended to multi layer networks. Back propagation through time (117) is a learning rule that employs gradient descent for training neural networks. However, gradient descent suffers from getting stuck in the local minima when searching for an optimal solution or global minima. The network by this nature of gradient descent gives a poor performance for training and testing. However, this disadvantage of performance degradation due to being stuck in local minima can be avoided by using a group of networks which are trained with different initial weights and the best result is voted as the final solution. This generally gives a much better result.

### 6.2.4 Theoretical Properties

In this section we review the different terms used to define the theoretical properties of neural networks such as computational capability, capacity, convergence

and generalization.

Computational capability of a neural network depends on the topology of the network. The number of neurons or the initial setting of the network greatly affects the computational capability of the system. However, there is no rule as to how many neurons will be required by a network to perform at its best capability.

Capacity of a neural network is its ability to model any given function. It is related to the amount of information that can be stored in the network. Simple networks with less neurons can successfully approximate a simple function, where as real time applications need complex networks with higher number of neurons and network layers.

Network convergence depends on a number of factors. Convergence is affected by the number of local minima that exist, which in turn depends on the error function and the network model. The second factor is the optimization method used which might not guarantee a convergence when far away from the local minima. The third factor is that most theoretical guarantees for a convergence are unreliable for practical applications that have huge amounts of data.

Generalization ability of a network is a measure of how well a network can generalize for unseen samples of data. It has been shown that for good generalization, size of the weights is more important than the size of the network (6). The problem of over training emerges when a network is over complex or over specified. There the capacity of the network exceeds the needed parameters. Over training can be avoided with the use of increased number of samples for training. The other popular solution for the problem of over fitting is the use of weight decay. Weight decay is a gradient descent method which fractionally decreases the weights during training. The other factor that affects the generalization ability of the network is under fitting of data. In contrary to over fitting under fitting occurs when the network is less complex with less free parameters.

## 6.3 Support Vector Machines

Support vector machines have been one of the widely used techniques in many real world applications. They were first introduced in (20), and since then they

have found a significant place in machine learning literature. The basic idea is to maximize the distance between two classes in the input space that one wishes to classify. This principle was highly successful in classification and regression problems surpassing other state-of-the-art methods. One of the main advantages of SVMs which made them popular in applications like FER is their ability to handle high dimensional feature vectors without affecting the time taken for training. This made the use of SVMs widely popular in many machine learning applications. In this section we provide the basic concept behind SVMs; the kernel trick which provides flexibility of replacing the default linear kernel with that of RBF, sigmoidal, polynomial and many others. We then shed some light on multi-class SVMs and regression. Theoretical information on SVMs is available in (73).

### 6.3.1 Basic Principle

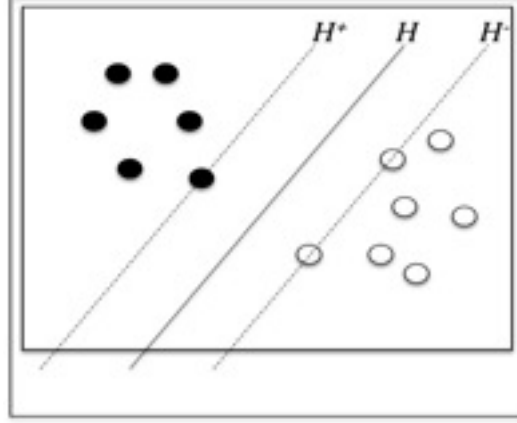
The basic idea behind SVMs is to know whether a test sample belongs to one of the two classes, defined by the given training data. Instead of single data points, here we view data point as a  $p$ -dimensional vector. Assuming the training samples are of the form:  $(x_i, y_i), i = 1, 2, \dots, n$  and  $x_i \in \mathbb{R}^d, y_i \in (-1, +1)$  where  $(\mathbf{x}_i)$  are called the co-variants or input vectors and  $(\mathbf{y}_i)$  the response variables or labels. We first look at a simple case where the data is in fact linearly separable. This makes it possible to have a hyperplane  $H$  consisting of a set of points  $x$  such that

$$H : w \cdot x + b = 0 \tag{6.12}$$

where  $\cdot$  denotes that dot product. The vector  $w$  is a normal vector perpendicular to the hyperplane and the parameter  $\frac{b}{\|w\|}$  is the offset of the hyperplane  $H$  from the origin along the normal vector  $w$ . This hyperplane  $H$  is able to divide the points having  $y_i = 1$  from those of having  $y_i = -1$ .

For the above hyperplane  $H$ , we can formulate an infinite number of equations by scaling both  $w$  and  $b$ . Here we need to choose the maximum margin (the distance between the hyperplanes) so that they are as far as possible while still separating the data. Two such hyperplanes can be described by the following equations





**Figure 6.3:** Maximum margin hyperplane  $H$  that separates the hypothetical training data points

$$H^+ : w \cdot x + b = 1 \quad (6.13)$$

and

$$H^- : w \cdot x + b = -1 \quad (6.14)$$

where  $H^+$  is a hyperplane consisting of all the data points having  $y_i = 1$  and  $H^-$  is a hyperplane consisting of all the data points having  $y_i = -1$  as shown in Fig. 6.3.

Since the data is assumed to be linearly separable, selection of these two hyperplanes should be such that no data points fall between them. To prevent data points falling into the margin, we require that:

$$w \cdot x + b \geq 1 \quad (6.15)$$

for all  $x_i$  data points in  $H^+$ .

or

$$w \cdot x + b \leq -1 \quad (6.16)$$

for all  $x_i$  data points in  $H^-$ .

The above two conditions can be combined and rewritten as:

$$y_i(w \cdot x_i) \geq 1 \quad (6.17)$$

for all  $1 \leq i \leq n$ .

The hyperplane  $H$  with maximum margin should be identified. The margin is equal to the distance between the hyperplanes  $H^+$  and  $H^-$ . The distance of  $H^+$  from origin is  $\frac{(1-b)}{\|w\|}$ , and the distance between hyperplane  $H^-$  to the origin is  $\frac{(-1-b)}{\|w\|}$ . Therefore the margin is equal to  $\frac{2}{\|w\|}$ . The margin can thus be minimized by minimizing  $\|w\|$  or by minimizing  $\frac{1}{2}\|w\|^2$ .

The optimization problem can then be defined as:

Minimize in  $(w, b)$

$$\frac{1}{2}\|w\|^2 \quad (6.18)$$

Subject to (for any  $t = 1, 2, \dots, n$ )

$$y_i(w \cdot x_i + b) \geq 1 \quad (6.19)$$

This way we can maximize the margin subject to the constraints so that all training data points fall on either side of the support hyperplanes  $H^+$  and  $H^-$ . The data points that lie on the hyperplane are called support vectors, as they support the hyper planes and hence determine the solution.

The solution to the above optimization problem can be solved using Lagrangian multipliers, which is a mathematical optimization technique that provides a strategy for finding the maxima and minima of a function subject to constraints. Considering an optimization problem such as

$$\begin{aligned} &\text{Maximize } f(x, y) \\ &\text{Subject to } g(x, y) = c. \end{aligned}$$

A new variable  $(\alpha)$  called a Lagrangian multiplier is introduced and the Lagrangian function is then defined as

$$L(x, y, \alpha) = f(x, y) + \alpha \cdot (g(x, y) - c) \quad (6.20)$$

Introducing a vector of Lagrangian multipliers  $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_l)$ , the Lagrangian function for our optimization problem is given as

$$L(w, b, \alpha) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^l \alpha_i y_i (w \cdot x_i + b) - 1 \quad (6.21)$$

The solution to the above problem can be found at the saddle point according to the optimization theory. The function must now be minimized with respect to  $w$  and  $b$  and simultaneously maximized with respect to  $\alpha$ , subject to  $\alpha \geq 0$ . Taking partial derivatives with respect to  $w$  and  $b$  and setting them to zero, we get

$$\frac{\partial}{\partial w} L(w, b, \alpha) = w - \sum_{i=1}^l \alpha_i y_i x_i = 0 \quad (6.22)$$

i.e.

$$w = \sum_{i=1}^l \alpha_i y_i x_i \quad (6.23)$$

Substituting  $w = \sum_{i=1}^l \alpha_i y_i x_i$  in the original Lagrangian function we get:

$$L_D = \frac{1}{2} \left( \sum_{i=1}^l \alpha_i y_i x_i \right)^2 - \sum_{i=1}^l \alpha_i (y_i \left[ \left( \sum_{i=1}^l \alpha_i y_i x_i \right) \cdot x_i + b \right] - 1) \quad (6.24)$$

$$L_D = \frac{1}{2} \sum_{i,j=1}^l \alpha_i \alpha_j y_i y_j x_i \cdot x_j - \sum_{i,j=1}^l \alpha_i \alpha_j y_i y_j x_i \cdot x_j - b \sum_{i=1}^l \alpha_i y_i + \sum_{i=1}^l \alpha_i \quad (6.25)$$

$$L_D = \sum_{i=1}^l \alpha_i - b \sum_{i=1}^l \alpha_i y_i - \frac{1}{2} \sum_{i,j=1}^l \alpha_i \alpha_j y_i y_j x_i \cdot x_j \quad (6.26)$$

Applying the second required differentiation with respect to  $b$ , we get

$$\frac{\partial}{\partial b} L(w, b, \alpha) = \sum_{i=1}^l \alpha_i y_i = 0 \quad (6.27)$$

Substituting  $\sum_{i=1}^l \alpha_i y_i = 0$  in the above equation we get:

$$L_D = \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i=1}^l \alpha_i \alpha_j y_i y_j x_i \cdot x_j \quad (6.28)$$

subject to  $\alpha \geq 0$ . However, the function should still be maximized with respect to the Lagrangian multipliers  $\alpha$ . Now  $L_D$  is a quadratic form and the constraints are all linear in  $\alpha_i$ . This is a quadratic programming problem and can be optimized using different computer packages.

Considering we have an  $\alpha^o$  which maximizes  $L_D$  subject to the given constraints and  $w^o$  and  $b^o$  the parameters of the corresponding optimal hyperplane, we get:

$$w^o = \sum_{i=1}^l \alpha_i^o y_i x_i \quad (6.29)$$

To calculate  $b^o$  we must make use of *Karush-Kuhn-Tucker* complementarity conditions. For any nonzero  $\alpha_i^o$ , we get

$$y_i(w^o \cdot x_i + b^o) = 1 \quad (6.30)$$

Since  $y_i = \pm 1$ , knowing the values of  $w^o$  and particular data point  $x_i$  will retrieve the value of  $b$ .

Once we know the separating hyperplane, the new points,  $x$  can be classified. The process is to just determine the associated  $y$  value which is evaluated as follows:

$$y = \text{sgn}(w^o \cdot x + b^o) \quad (6.31)$$

This makes the support vector machine a classifier.

### 6.3.2 Kernel Trick

The kernel trick (20) has gained interest for its application in generalizing the SVMs to higher dimensional feature spaces. The motivation comes from (59) which states that the data that is not linearly separable in a low-dimensional space can be transformed into a higher-dimensional feature space using transformation which yields linearly separable data. So primarily the use of kernel trick is

motivated for use in case of data which is not linearly separable in the input space. The kernel trick roots are based on Hilbert-Schmidt operators and reproducing kernel Hilbert spaces. A Hilbert space is an inner product space with the additional property of completeness. The essence of kernel trick drives from Mercers theorem. Mercers theorem in simplified terms states that for any bounded, non-negative function  $K(x, y)$  which is symmetric in its arguments  $(x, y) \in G$ , there exists a Hilbert space  $H_K$  (called the feature space), and a mapping  $\phi : G \rightarrow H_K$  such that  $\phi(x)\phi(y) = K(x, y)$  for all  $(x, y) \in G^2$  (72). Such a  $K$  is called a kernel function. Mapping of  $\phi$  from input space to high dimensional space will more likely yield linear separability of our transformed data points  $\phi(x_1), \dots, \phi(x_l)$  than the original data points. Thus when  $G$  is not an inner product space we cannot even define a hyperplane and hence linear separability in  $G$ .

### 6.3.3 Handling Linear Inseparability

One of the limitations of original SVMs relates to its inability in handling training data that is not linearly separable in the feature space corresponding to the chosen kernel  $K$ . However, this linear inseparability can be handled using *soft margin* hyperplane (30). Although kernel trick aims at making data linear separable in the feature space, cases where separability can not be achieved still persist.

This approach involves introducing *slack variables*  $\xi = (\xi_1, \xi_2, \dots, \xi_l)$ . The constraints function is then given as:

$$y_i(w \cdot x_i + b) \geq 1 - \xi_i, i \in (1, \dots, l) \quad (6.32)$$

$$\Rightarrow w = \sum_{i=1}^l \alpha_i y_i x_i \quad (6.33)$$

for  $\xi_i \geq 0$ . In order to penalize the errors, the objective function is changed to

$$\frac{1}{2} \|w\|^2 + C \sum_{i=1}^l \xi_i \quad (6.34)$$

where  $C$  determines the extent to which the errors are penalized.

The Lagrangian function for this optimization problem is then given as:

$$L(w, b, \xi, \alpha, \eta) = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^l \xi_i - \sum_{i=1}^l \alpha_i \{y_i(w \cdot x_i + b) - 1 + \xi_i\} + \sum_{i=1}^l \eta_i \xi_i \quad (6.35)$$

where  $\mu = (\mu_1, \dots, \mu_l)$  multipliers are introduced to enforce the constraint  $\xi_i \geq 0$ . Substituting these constraints into the original optimization problem

$$\frac{\partial}{\partial w} L(w, b, \xi, \alpha, \eta) = w - \sum_{i=1}^l \alpha_i y_i x_i = 0 \quad (6.36)$$

$$\frac{\partial}{\partial b} L(w, b, \xi, \alpha, \eta) = \sum_{i=1}^l \alpha_i y_i = 0 \quad (6.37)$$

$$\frac{\partial}{\partial \xi_i} L(w, b, \xi, \alpha, \eta) = C - \alpha_i - \eta_i = 0 \quad (6.38)$$

Now taking partial derivatives with respect to  $w, b$  and  $\xi$  and setting the value to zero, we get:

$$L_D(b, \xi, \alpha, \eta) = \frac{1}{2} \left( \sum_{i=1}^l \alpha_i y_i x_i \right)^2 + C \sum_{i=1}^l \xi_i - \sum_{i=1}^l \alpha_i (y_i [(\sum_{j=1}^l \alpha_j y_j x_j) \cdot x_i + b] - 1 + \xi_i) - \sum_{i=1}^l \eta_i \xi_i \quad (6.39)$$

$$\begin{aligned} L_D(b, \xi, \alpha, \eta) = & \frac{1}{2} \sum_{i,j=1}^l \alpha_i \alpha_j y_i y_j x_i \cdot x_j + C \sum_{i=1}^l \xi_i - \sum_{i,j=1}^l \alpha_i \alpha_j y_i y_j x_i \cdot x_j - \sum_{i=1}^l \alpha_i y_i b \\ & + \sum_{i=1}^l \alpha_i + \sum_{i=1}^l \alpha_i \xi_i - \sum_{i=1}^l \eta_i \xi_i \end{aligned}$$

$$L_D(b, \xi, \alpha, \eta) = \sum_{i=1}^l \alpha_i + \sum_{i=1}^l \xi_i (C - \alpha_i - \eta_i) - \frac{1}{2} \sum_{i,j=1}^l \alpha_i \alpha_j y_i y_j x_i \cdot x_j - b \sum_{i=1}^l \alpha_i y_i \quad (6.40)$$

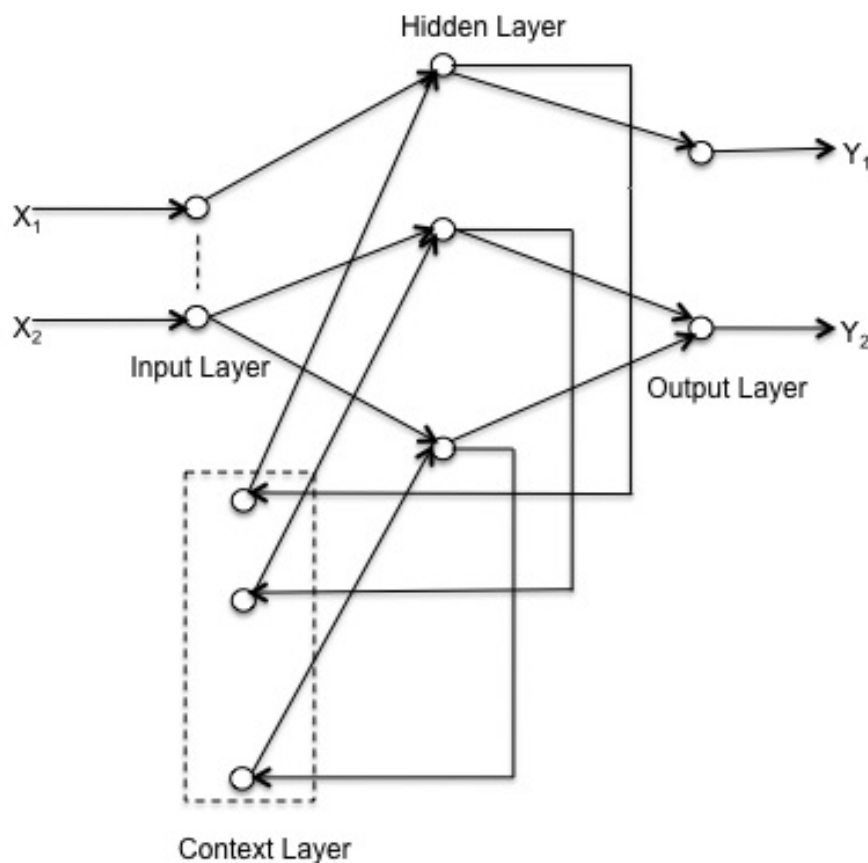
The further substitutions  $\sum_{i=1}^l \alpha_i y_i = 0$  and  $C - \alpha_i - \eta_i = 0$ , we get:

$$L_D(b, \xi, \alpha, \eta) = \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i,j=1}^l \alpha_i \alpha_j y_i y_j x_i \cdot x_j \quad (6.41)$$

The above quadratic programming problem can now be computed efficiently.

## 6.4 Recurrent Neural Networks

### 6.4.1 First-Order Recurrent Neural Networks



**Figure 6.4:** Architecture of First Order Recurrent Neural Network

First-order RNNs have been applied to many real world applications like speech recognition (113), (3), signature verification (99), gesture recognition (26) and financial forecasting (69). First-order RNN uses context units to store the previous time step data with a time lag as shown in Fig. 6.4.

The hidden unit activations in case of first-order RNN at time  $t + 1$  is given as

$$S_j^{t+1} = g\left(\sum_{i=1}^K V_{ji} I_i(t) + \sum_{i=1}^N W_{ji} S_i(t)\right) \quad (6.42)$$

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

The different architectures available in the first-order RNNs are classified based on the values taken by the context units. In Elman architecture the context units takes the output values of the state units with a time lag, and are then used along with the input as inputs to the state units in the next time step. Thus in Elman networks the number of context units is equal to the number of state units. In Jordan networks (65), however, the output values of the output units are stored in the context units with a time lag.

### 6.4.2 Second Order Recurrent Neural networks

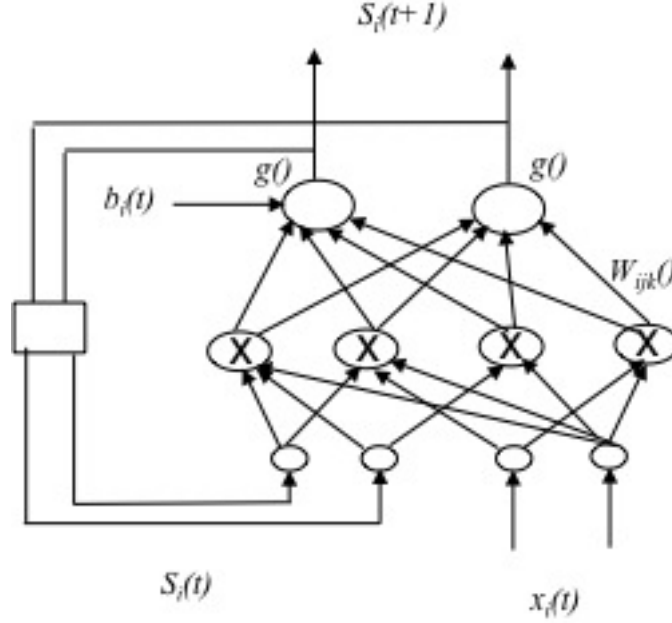
Second order RNNs were first proposed by Giles et al. (55) and were shown to be more useful in the modeling of finite state automaton than first order RNNs. In these networks the product of the input and the state units with all combinations is used as the input to the output units in the output layer. The activation function for the output units is given as

$$S_i(t+1) = g\left(\sum_{j=1}^m \sum_{k=1}^n W_{ijk} S_j(t) x_k(t)\right) \quad (6.43)$$

where  $m$  is the number of input units and  $n$  the number of state units.

Fig. 6.5 show the architecture of second order recurrent neural network with two input units and two state units.  $g()$  is the transfer function,  $x_k(t)$  is the input to the network at time  $t$  and  $w_{ijk}$  represents the weights associated with the network. Taking the number of neurons into consideration, first order networks have shown to have better generalization capability than second order networks which was attributed to the increased number of weight connections in second order RNNs (55).





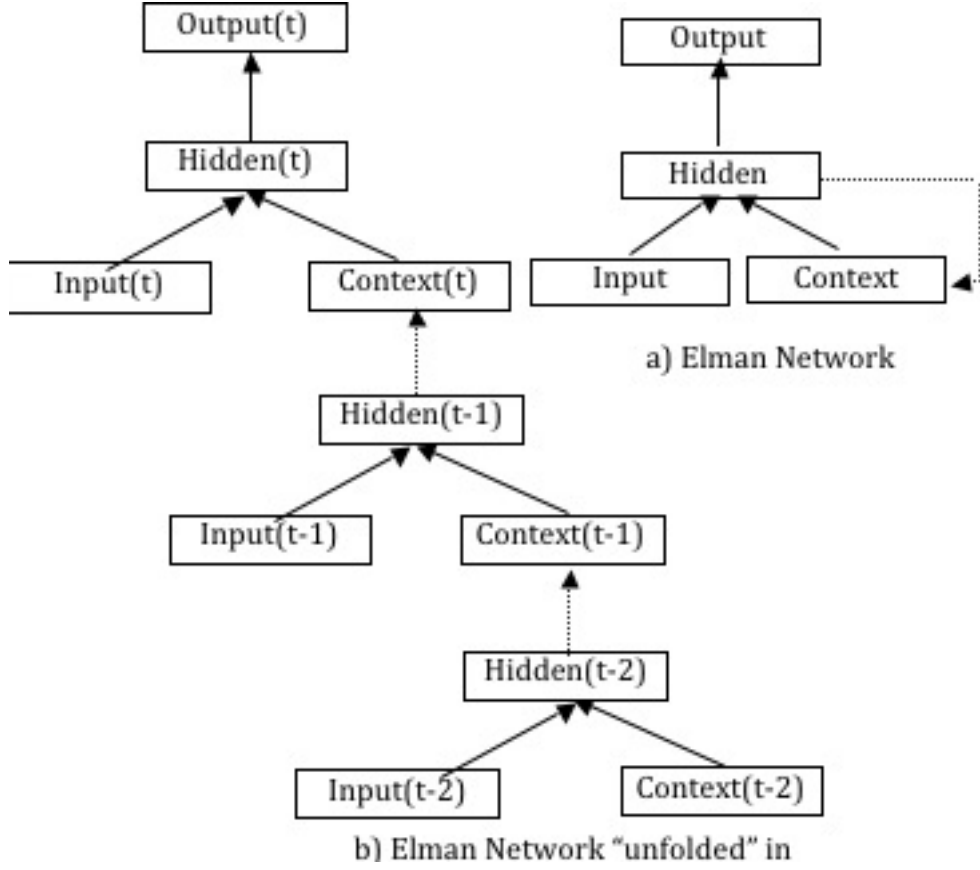
**Figure 6.5:** Architecture of Second Order Recurrent Neural Network

### 6.4.3 Back Propagation Through Time Learning Algorithm

Back propagation through time training algorithm is the extension of the standard back propagation algorithm. The Elman networks proposed by Jeff Elman (39) were first trained using truncated back propagation algorithm where the output values of the state units at time  $t - 1$  were simply regarded as additional input units at time  $t$  and the error generated at the state units was used to modify the weights to these additional input units. Later it was shown that the error can be propagated even further which lead to the design of BPTT algorithm (117).

The basic principal of BPTT algorithm is to unfold the network. Fig. 6.6(a) shows a RNN and (b) shows the same network unfolded in time. Algorithm 6.1 gives the general layout of the BPTT algorithm. In supervised learning methods the role of the training algorithm is to adjust the system weights so that the output values at the output nodes are equal to the target values at specific time.

Algorithm 6.1 starts with constructing the network with the desired number



**Figure 6.6:** Recurrent Neural Network a) Elman Network b) Elman Network “un-  
folded” in time for 2 time steps

of input, hidden, context and output units. Then all the network weights are initialized to small random values usually between  $[-1,1]$ . The main loop iterates over the entire training examples. For each training example, the network calculates the network output at time  $t$ , copies the output values of hidden units at time  $t$  to the context units at time  $t + 1$  and again calculates the network outputs at time  $t + 1$ . This step continues until the end of string is reached. Since in most cases the expected output after reading of the entire string is given, all the network outputs at the intermediate stages are assumed to be zero. The final network output is then used to compute the gradient with respect to the output error. This gradient is used to compute weight updates for each layer in the network, which are further propagated back to the previous time steps, and

thus for the entire string length. The decision to update the weights depends on whether we are employing *batch learning* or *pattern learning*. In case of pattern learning the weights are updated after each pattern is read and in case of batch learning the weights are updated after all the patterns in the training set are read. Algorithm 6.1 employs a batch learning process. This gradient descent step is repeated until the network learns all the training samples.

### Back propagation through time (Training samples, $\eta$ )

*Each training sample constitutes of 2 values which is in the form  $(x, d)$  where  $x$  is a vector of input values at different time steps and  $d$  is the vector of desired output values.  $\eta$  is the learning rate, usually a value between 0 and 1*

- Create the network with input, hidden, context and output values
- Initialize the network weights to small random values usually in the range  $[-1, 1]$
- Until the condition of termination is met, Do
  - For each training sample  $(x, d)$  of length  $l$ 
    - Propagate the input forward through the network
      - \* *Input the instance of the input  $x$  at time  $t = 1$ , context unit values all set to 0 and compute the output at the output units*
      - \* *Copy the output values of the hidden units at  $t = 1$  to context units at time  $t = 1$ , input the instance of input  $x$  at time  $t = t + 1$ , and compute the output*
      - \* *Repeat the above step till all the instances are fed*
      - \* *Compute the value of the output units at the last instance of the input vector  $x$  and store*
    - Propagate the error backward through the network
      - For each training sample  $(x, d)$  of length  $l$* 
        - \* *Compute the error  $\delta_j^L$  for each neuron  $j$  in every layer  $L$*

- \* *Propagate the error back to layers  $l = l - 1$*
- Update the network weights
  - \* *Update each network weight  $W_j^L$  at each layer  $L$*

**Algorithm 6.1: The BPTT algorithm for RNNs employing batch learning. The same algorithm can be used for pattern learning by combining steps 2 and 3 into one loop**

The time complexity of the BPTT algorithm is given as  $O(d^3)$  where  $d$  is the depth of the error back propagated. Below are the training equations for the BPTT algorithm using batch learning. Here the time  $t$  corresponds to the layers unfolded in time.

The error of the output node at time  $t$  is given as

$$F\_yhat_i = y_i - d_i \quad (6.44)$$

where  $y_i$  is the output of node  $i$  at time  $t$  and  $d_i$  is the desired or the target output at time  $t$ . The total error over all the output units for training sample  $x$  is given as:

$$E_x = \sum_{t=0}^t (y_i - d_i)^2 \quad (6.45)$$

The error gradient at the output units is given as

$$\delta_i = (d_i - y_i)y_i(1 - y_i) \quad (6.46)$$

And the error gradient at the hidden units is calculated as

$$y(i)(1 - y(i)) \sum_{j=1}^m \delta_j W_{ij} \quad (6.47)$$

The weight updated is then given as

$$\Delta W_{ij} = \eta \delta_i d_{ij} \quad (6.48)$$

Back propagation through time algorithm implements gradient descent to search for the possible network weights in weight space. This is achieved by reducing the error  $E$  between the desired output for the training samples and the network output values. However, it is fairly possible that the weight space contains a number of local minima's and thus the network in most cases is struck in the local minima than the global minima. In order to improve the generalization performance using back propagation we can include a *momentum* term in the weigh update where the weight update during  $n^{th}$  iteration is partially depend on the  $(n - 1)^{th}$  iteration. The weight update equation using the momentum term is given as

$$\Delta W_{ij}(n) = \mu \delta_i(x_{ij}) + \eta \Delta W_{ij}(n - 1) \quad (6.49)$$

where  $\Delta W_{ij}(n)$  is the weight update performed during the  $n^{th}$  iteration.  $\mu$  is the momentum term which is usually in the range  $[0,1)$ . The momentum term provides a definite direction to the search in the weight space in situations where the direction of the local gradient changes rapidly. The other method of improving the generalization performance is to train different networks with the same data but network weights initialized with different small random values. A similar approach for calculating the gradients like BPTT is used in another famous approach known as RTRL discussed in the next section.

### 6.4.4 Real Time Recurrent Learning Learning Algorithm

The other popular implementation for obtaining the derivatives is Real time recurrent learning. The RTRL algorithm was first proposed by Williams and Zipser (121). In RTRL, weights are updated concurrently with network execution and the derivatives of the node outputs with respect to the weights are calculated in the forward pass. Thus RTRL is also called as forward method. There is no unfolding of the network performed in RTRL algorithm. The disadvantage of RTRL is that its computational complexity which is in the order of  $O(n^4)$ ,  $n$  being the number of nodes in the network.

The RTRL learning algorithm for RNNs is based on minimizing the instantaneous squared error at the output of the first neuron in the RNN (33), which is given as

$$\min(e^2(k)) = \min([s(k) - y_i(k)]^2) \quad (6.50)$$

where  $e(k)$  is the instantaneous error at the output neuron and  $s(k)$  is some teaching (desired) signal.  $y_i(k)$  is given as

$$y_i(k) = \eta(v_i(k)), i = 1, 2, \dots, N \quad (6.51)$$

where

$$v_i(k) = \sum_{l=1}^{p+n+1} w_{i,l}(k) u_l(k) \quad (6.52)$$

and

$$u_i^T(k) = [s(k-1), \dots, s(k-p), 1, y_1(k-1), y_2(k-1), \dots, y_N(k-1)] \quad (6.53)$$

$w(k)$  is the weight vector,  $p$  is the number of external neurons and  $N$  is the number of feedback connections and vector  $u(k)$  comprises of both the external and feedback inputs to a neuron.  $\eta$  is the non-linear activation function of a neuron and is assumed to be continuously differentiable.

Hence from the original equation, the correction for the  $i^{th}$  weight of the neuron  $n$  at the time instant  $k$  can be given as

$$\Delta w_{n,l}(k) = -\eta \frac{\partial}{\partial w_{n,l}(k)} e^2(k) = -2\eta e(k) \frac{\partial e(k)}{\partial w_{n,l}(k)} = 2\eta e(k) \frac{\partial y_k(k)}{\partial w_{n,l}(k)} \quad (6.54)$$

This can be further evaluated as

$$\frac{\partial y_1(k)}{\partial w_{n,l}(k)} = \phi'(v_1(k)) \frac{\partial v_1(k)}{\partial w_{i,j}(k)} = \phi'(v_1(k)) \left( \sum_{\alpha=1}^N \frac{\partial y_{\alpha}(k-1)}{\partial w_{n,l}(k)} w_{l,\alpha+p+1}(k) + \delta_{nl} u_1(k) \right) \quad (6.55)$$

where

$$\delta_{nl} = 1, n = 1 \quad (6.56)$$

or

$$\delta_{nl} = 0, n \neq 1 \quad (6.57)$$

Considering the commonly used assumption for gradient based algorithms which is also used in RTRL (89), (121), when the learning rate  $\eta$  is sufficiently small we have

$$\frac{\partial y_\alpha(k-1)}{\partial w_{n,l}(k)} \approx \frac{\partial y_\alpha(k-1)}{\partial w_{n,l}(k-1)} \quad (6.58)$$

and the recursive equation for gradient updates becomes

$$\pi_{n,l}^j(k+1) = \phi'(v_j) \left[ \sum_{m=1}^N w_{j,m}(k) \pi_{n,l}^m(k) + \delta_{nl} u_1(k) \right] \quad (6.59)$$

where the initial condition is

$$\pi_{n,l}^j(0) = 0 \quad (6.60)$$

with the values for  $j$ ,  $n$  and  $l$  given as below

$$\pi_{n,l}^j(k) = \frac{\partial y_j(k)}{\partial w_{n,l}(k)}, 1 \leq j, n \leq N, 1 \leq l \leq (p+1+N) \quad (6.61)$$

Finally the correction to the weight  $w_{n,l}(k)$  can be given as

$$\Delta W_{n,l}(k) = 2\eta e(k) \pi_{n,l}^l(k) \quad (6.62)$$

### 6.4.5 Applications

Recurrent neural networks are a class of neural network where connections between units form a directed cycle. This creates an internal state of the network which allows it to exhibit dynamic temporal behavior. This characteristic of RNNs has found applications in speech recognition, forecasting and signature

verification. In this section we review the different areas where RNNs have been successfully used such as speech recognition, signature verification and forecasting.

The ability of RNNs to treat and store time depend information enables them to learn space-time relationships. This ability to learn space-time relations makes RNNs useful in speech recognition, where the examples are space-time patterns. The main goal of automatic speech recognition is to design automatic systems that are capable of interpreting the vocal signs coming from a source (human speaker) in terms of linguistic categories. The task of speech recognition is divided into two steps: 1) feature extraction, where the features are extracted from the stream of data and 2) designing of an system to model the extracted features. Feature extraction is important as the speech sequences contain irrelevant information like background noise etc. Recurrent neural networks have been successfully used for speech recognition (113), (2).

Signature verification systems are designed for authentication of a signature. The verification system is composed of two units 1) the pre-processing unit which extracts the feature of a signature which include the timing information and the positioning of the pen point which making signatures and 2) a modeling unit which learns the extracted features. Recurrent neural networks have been successfully used in signature verification (4). 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. These systems provide a reliable means of personal identification in almost any large modern organizations like airports, banks, and other high security areas.

Financial forecasting is one of the applications that needs great precision. Recurrent neural networks because of their dynamic behavior have been successful in forecasting the financial trading (96). In (96), time series data and other factors are fed 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 (58), (25), (57). It was shown that RNNs perform similar in dynamic recognition of facial expressions to that of human recognition of facial expressions.

### 6.4.6 Computation Capability of RNNs

The computational capability of the RNNs mainly depends on the specific task they are being used for. The task itself will not be able to give a better understanding of the learning algorithm used and networks learning capability. The wide use of RNNs in speech recognition, gesture recognition and forecasting require different feature extraction techniques which may not provide the necessary information for the understanding of network and its fundamental characteristics. We thus require structures that can be used to understand the capabilities of RNNs.

In this regard, finite automata and the languages represented by them are known to be the fundamental models used to represent the knowledge learned by the RNNs. There is no need of feature extraction and the knowledge represented by the network is the relational representation of the finite automata. Also since RNNs are complex networks to work with, it is important to be able to understand the network architecture and its behavior with the help of training with finite automata. If a network can represent a finite automaton that represents a language, the network should be able to learn that language.

### 6.4.7 Vanishing Gradient Problem

Recurrent Neural Networks are capable of handling temporal dependencies. This tendency makes them useful in applications which include temporal delays of signals like speech, gesture, forecasting and time series analysis. The network must learn to assign a set of inputs for the desired outputs which is performed using gradient descent. Gradient descent method involves sending the current signal back in time through the feedback connections to previous inputs to gain adequate knowledge. Conventional back propagation algorithm suffers from long learning times. This makes error signals flowing backwards in time to vanish which is a common phenomenon observed in BPTT and RTRL. This “*Vanishing*

*of Gradients*” makes it hard to learn long-term dependencies due to insufficient weight changes (106).

Vanishing gradient corresponds to vanishing information in the internal states of a RNN. To overcome this, four types of solutions are given:

- Methods which do not use gradients:

Global search methods such as multi-grid random search (17) and random weight guessing (105) do not use gradient information. These solutions are good for handling long term dependencies for simple problems such as nets with very few parameters and the absence of precise computation.

- Methods which enforce higher gradients:

Time weighed Pseudo- Newton optimization and discrete error propagation (17) can be used to enforce large values of gradient. However, these methods encounter problems learning to store precise real values information over a period of time.

- Methods which operate on higher levels:

Methods such as Kalman filters can be used for training RNNs.

- Methods which use special architectures:

In time delay neural networks (TDNN), the net activations from the previous times steps are fed back into the net using fixed delay lines. So the error decrease is slowed down because it uses shortcuts as it gets propagated back. However there is one trade off i.e. increasing the length of delay line increases the error flow but the net has more parameters/units. Long short term memory uses a special architecture to enforce a constant error flow through special units. Here the perturbations by the current irrelevant signals are prevented by multiplicative units.

### 6.4.8 Network Optimization using Weight Decay

The generalization ability of an artificial neural network depends on the balance between information available in the training examples and the complexity of the

networks (15). A very complex network for generalizing a training set containing little information will lead to over-fitting of the data where as a simple network for generalizing a training set containing complex relations will under-fit the data. Both these scenarios will lead to bad generalization (71).

A successful way of avoiding the above mentioned problems is to limit the growth of the weights through some kind of weigh decay (71). This mechanism will prevent the weights from growing too large. This is achieved by adding a term to the cost function. This cost function is designed to penalize large weighs. The error measure is thus given as

$$E(w) = E_{\alpha}(w) + \frac{1}{2}\lambda \sum_i w_i^2 \quad (6.63)$$

where  $E_{\alpha}$  is the error measure usually calculated as the sum of squared errors and  $\lambda$  is the parameter which determines the penalizing effect on the large weights. This is usually known as decay constant.  $w$  is the weight vector.

Using gradient descent, we have

$$w_i \alpha - \frac{\partial E_{\alpha}}{\partial w_i} - \lambda w_i \quad (6.64)$$

The use of weight decay has shown to increase the generalization capability of classification systems (61), (71). However, there are a couple of disadvantages associated with weight decay. Firstly, convergence time for training with weight decay increases with increasing decay rate. Secondly, one needs to set the decay rate prior to training (56). There is no algorithm which provides a solution to obtain a best decay constant for a given network.

### 6.4.9 Network Optimization using Weight Elimination

A simple technique was proposed by (56) which improves the performance significantly over the networks that were trained using weight decay. The results also showed that training is faster due to shrinking network size and there is no need for determining a decay rate prior to training (56). The goal of the pruning heuristic is to train networks of small size with improved generalization performance. The process involves training a large network for a known regular grammar and

then applying the pruning heuristic. Whenever the training is successful the state neuron with the smallest weight vector is removed and the network is retrained using the same training set. This process of pruning the state neuron is repeated until either a network with satisfactory generalization performance is obtained or until the retraining fails to converge within a certain number of epochs (56). The pruning cycle is stopped when the current network fails to converge and the previously pruned network is considered as the solution network.

## 6.5 Hidden Markov Models

### 6.5.1 Definition

A hidden Markov model (HMM) is a finite set of states. Each state is usually associated with a multi-dimensional probability distribution (100). 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 generated associated with the probability distribution. This outcome from a state is visible to the user but not the state themselves, hence the hidden Markov model is given to these systems.

An HMM is defined by

- The number of states given by  $N$
- The number of observations given by  $M$ .  $M$  is infinite if the observations are continuous
- Set of state transition probabilities given as  $\Lambda = \{a_{ij}\}$  where  $a_{ij} = p\{q_{t+1} = j | q_t = i\}$ ,  $1 \leq i, j \leq N$ ,  $q_t$  denotes the current state.

Here the transition probabilities are given as

$$a_{ij} \geq 0, 1 \leq i, j \leq N \quad (6.65)$$

and

$$\sum_{j=1}^N a_{ij} = 1, 1 \leq i \leq N \quad (6.66)$$

- The probability distribution for each of the state is given as

$$B = \{b_j(k)\} \quad (6.67)$$

where

$$b_j(k) = p\{o_t = v_k | q_t = j\}, 1 \leq j \leq N, 1 \leq k \leq M \quad (6.68)$$

$v_k$  denotes the  $k^{th}$  observation symbol and  $o_t$ , the current parameter vector. Here the following should be satisfied

$$b_j(k) \geq 0, 1 \leq j \leq N, 1 \leq k \leq M \quad (6.69)$$

and

$$\sum_{k=1}^M b_j(k) = 1, 1 \leq j \leq N \quad (6.70)$$

In case the observations are continuous, the continuous probability density function is given as

$$b_j(o_t) = \sum_{m=1}^M c_{jm} \eta(\mu_{jm}, \sigma_{jm}, o_t) \quad (6.71)$$

where

$\eta$  is the Gaussian distribution

$\mu_{jm}$  is the mean vectors

$\Sigma_{jm}$  is the Covariance matrices and

$c_{jm}$  the weight coefficients

where  $c_{jm}$  satisfies the following conditions

$$c_{jm} \geq 0, 1 \leq j \leq N, 1 \leq m \leq M \quad (6.72)$$

and

$$\sum_{m=1}^M c_{jm} = 1, 1 \leq j \leq N \quad (6.73)$$

- The initial state distribution is given by  $\pi = \{\pi_i\}$

where

$$\pi = p\{q_t = i\}, 1 \leq i \leq N \quad (6.74)$$

A HMM with discrete probability distribution can now be denoted as

$$\lambda = (\Lambda, c_{jm}, \mu_{jm}, \sum_{jm}, \pi) \quad (6.75)$$

### 6.5.2 Assumptions in HMMs

A set of assumptions are made in the theory of HMMs for the sake of mathematical and computational traceability

- The Markov assumption: It is assumed that next state is dependent only on the current state.
- The stationary assumption: It is assumed that 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.

### 6.5.3 Basic Problems in HMMs

The problems that are associated with a HMM are as follows

- The evaluation problem
- The decoding problem
- The learning problem

Here we concentrate on the learning problem. Solution to this problem is required if one wants to use HMMs for recognition tasks. The learning problem is defined as

- Given a model  $\lambda$  and a sequence of observations  $O = o_1, o_2, \dots, o_T$ , how should the model parameters be adjusted in order to maximize  $p\{O, \lambda\}$ ?

In simple terms, the learning problem is to adjust the HMM parameters so that the given set of observations i.e. the training set is optimally represented by the model for a specific application. This also involves choosing the best optimization criteria out of the available options depending on the specific application. In literature two such criteria are available such as Maximum Likelihood (ML) and Maximum Mutual Information (MMI). Here we concentrate on the Baum- Welch algorithm under ML criteria.

### 6.5.4 Maximum Likelihood (ML) Criterion

Under this criterion one tries to maximize the probability of a given set of observations  $O_w$  wrt the HMM model  $\lambda_w$  for a given class  $w$ .

The probability is thus given as

$$L_{tot} = p\{O|\lambda\} \tag{6.76}$$

given that only one class  $w$  is considered at a time. To maximize the value  $L_{tot}$  we choose model parameters that are locally maximized using Baum-Welch method or gradient based method.

## 6.6 LSTM Recurrent Neural Networks

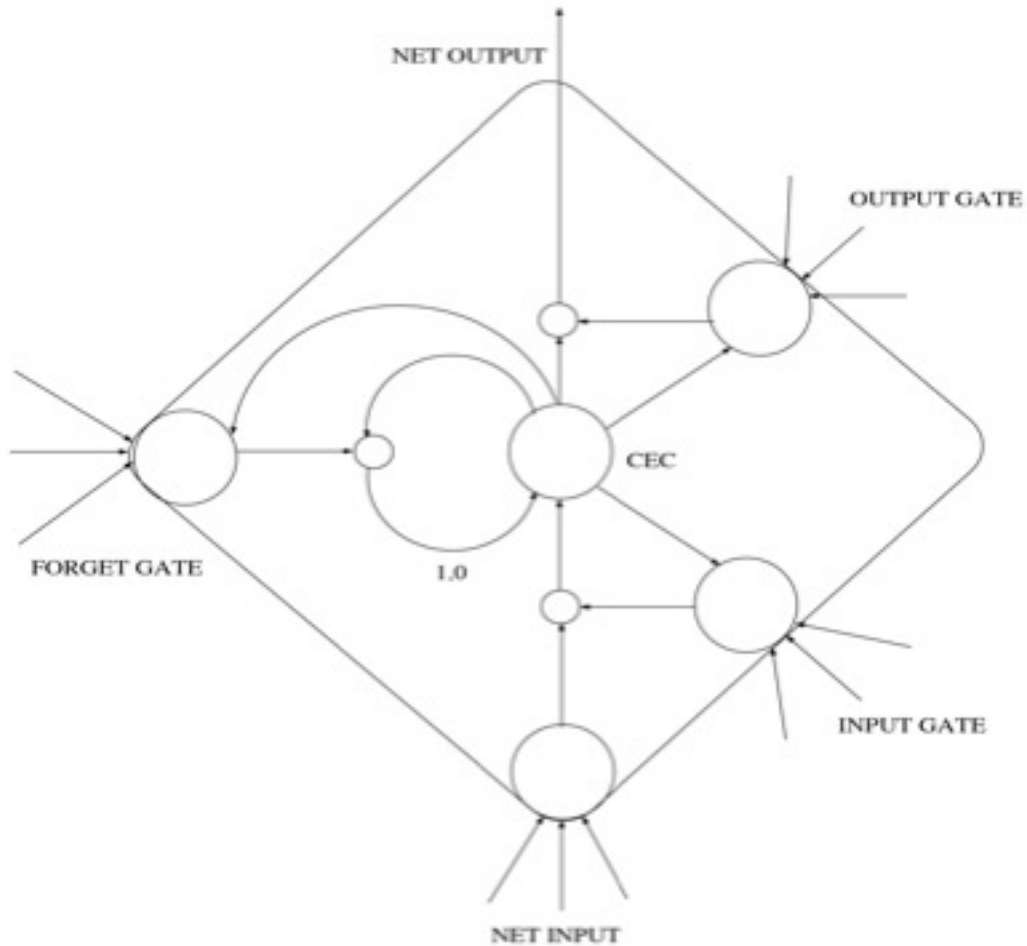
Recurrent neural networks are capable of learning dynamic behavior of systems with gradient descent without much problem. However, they fail to successfully learn long-term dependencies (51). They fail to propagate the information over long time steps using gradient descent learning mechanism. The gradient tends to zero after certain time steps making RNNs non capable of learning long-term dependencies. To overcome this problem we need networks which can successfully propagate the information over long time steps. Long short term memory was proposed to overcome the problem of long-term dependencies (54). They are composed of memory cells and gates.

### 6.6.1 Architecture of LSTM

Fig. 6.7 shows the structure of an LSTM with a memory block with single cell and three adaptive, multiplicative gating units shared by all cells in the block. Each memory cell has at its core a recurrently self-connected linear unit known as Constant Error Carousel (CEC). By circulating the activation and error signals indefinitely, CEC provides short-term memory storage for extended time periods. The input, forget and the output gate are trained to learn information that needs to be stored in the memory, how long to store it and when to read it out. Combining memory cells into blocks allows them to share the same gates and thus reducing the number of adaptive units.

The general notation used is as follows:  $j$  denotes indexed memory blocks,  $v$  denotes indexes memory cells in block  $j$  (with  $s_j$  cells), such that  $c_j^v$  denotes the  $v^{th}$  cell of the  $j^{th}$  memory block,  $w_{lm}$  is the weight connecting unit  $m$  to unit  $l$ . Index  $m$  ranges over all source units and  $y_m(t-1)$  refers to the source unit activation of the input unit. The output  $y_c$  of the memory cell  $c$  is calculated based on the current cell state  $s_c$  and four sources of input;  $z_c$  is the input to the cell itself, while  $z_{in}$ ,  $z_\phi$  and  $z_{out}$  are fed into the input, forget and the output gate respectively.





**Figure 6.7:** LSTM Architecture with Memory Cell

## 6.7 Learning Algorithm

Long short-term memory operated in discrete time steps  $t = 0, 1, 2, \dots, n$ , each involving the updating of all unit activations known as *forward pass* followed by computation of error signals for all the weights known as *backward pass*.

Forward pass consists of three steps calculating: input, cell state and output.

*Input:* During each forward pass, net cell input is calculated as

$$z_{c_j^v}(t) = \sum_m w_{c_j^v m} y_m(t-1) \quad (6.77)$$

where  $z_{c_j^v m}$  is input to the  $v^{th}$  of the  $j^{th}$  memory block,  $w_{c_j^v m}$  is the weight on the connection from unit  $c_j^v$  to unit  $m$  and  $y_m(t-1)$  is the unit activation which refers to an input unit. We then apply the input squashing function  $g$ . The result is multiplied by activation of the memory blocks input gate calculated by applying a logistic sigmoid squashing function  $f_{in}$  with range  $[0,1]$  to the gates net input  $z_{in}$ , we get:

$$y_{in_j}(t) = f_{in_j}(z_{in_j}(t)) \quad (6.78)$$

$$z_{in_j} = \sum w_{in_j m} y_m(t-1) \quad (6.79)$$

The activation  $y_{in}$  of the input gate multiplies the input to all cells in the memory block, and thus determines which activity patterns are stored into it. During training, input gate learns to open ( $y_{in} \neq 1$ ) so as to store relevant inputs in the memory block. Similarly it learns to close ( $y_{in} \neq 1$ ) so as to shield it from irrelevant ones.

*Cell state:* At  $t = 0$ , the activation or state denoted by  $s_c$  of a memory cell  $c$  is initialized to zero. Subsequently, the recurrently self-connected linear unit of the memory cell accumulates a sum, discounted by the forget gate over its input. The memory blocks forget gate activation is then given as:

$$y_{\phi_j}(t) = f_{\phi_j}(z_{\phi_j}(t)) \quad (6.80)$$

$$z_{\phi_j}(t) = w_{\phi_j m} y_m(t-1) \quad (6.81)$$

where  $f_{\phi}$  is a logistic sigmoid function with range  $[0,1]$ . The new cell state is then obtained by adding the squashed, gated cell input to the previous state multiplied by forget gate activation given as:

$$s_{c_j^v}(t) = y_{\phi_j} s_{c_j^v}(t-1) + y_{in_j}(t) g(z_{c_j^v}(t)) \quad (6.82)$$

$$s_{c_j^v} = 0 \quad (6.83)$$

Thus the activity circulates in the self-connected linear unit of the gate as long as the forget gate remains open ( $y_\phi \neq 1$ ). The input gate learns what to store in the memory block, the forget gate learns for how long to retain the information, and once it is outdated to erase it by resetting the cell state to zero. This prevents the cell state from growing to infinity and enables the memory block to store fresh data without undue interference from prior operation (54).

*Output:* Cell output  $y_c$  is calculated by multiplying cell state  $s_c$  by the activation  $y_{out}$  of the memory blocks output gate, given as:

$$y_{c_j^v}(t) = y_{out_j}(t)s_{c_j^v}(t) \quad (6.84)$$

Long short-term memories backward pass is a fusion of error back propagation for output units and output gates and is a customized version of RTRL for weights to cell inputs, input gates and forget gates.

*Output units and gates:* For the output units the standard back propagation weight changes are given as:

$$\Delta w_{km}(t) = \alpha \delta_k y_m(t-1), \quad (6.85)$$

$$\delta_k(t) = -\frac{\partial E(t)}{\partial z_k(t)} \quad (6.86)$$

where  $\Delta w_{km}$  is the change in weight from unit  $m$  to unit  $k$ ,  $\alpha$  is the learning rate,  $\delta_k$  is the negative gradient of the objective function  $E$  by gradient descent (subject to error truncation) and  $y_m(t-1)$  is the unit activation of the input unit. The customary squared error objective function based on the targets  $t_k$  yields

$$\delta_k(t) = f'_k(z_k(t))e_k(t) \quad (6.87)$$

where  $f_k$  is the output squashing function and  $e_k(t) := t_k(t) - y_k(t)$  is the externally injected error. Weight changes for connections to the output gate

of the  $j^{th}$  memory block from source units  $m$  are obtained by standard back propagation:

$$\Delta w_{out_j m}(t) = \alpha \delta_{out_j}(t) y_m(t), \quad (6.88)$$

$$\delta_{out_j}(t)^{tr} = f'_{out_j}(z_{out_j}(t)) \left( \sum_{v=1}^{s_j} (t) \sum_k w_{kc_j^v} \delta_k(t) \right) \quad (6.89)$$

where  $^{tr}$  = indicates the error truncation.

*RTRL weight changes:* The weight changes  $\Delta w_{lm}$  for connections to the cell ( $l = c_j^v$ ) input gate ( $l = in$ ) and the forget gate ( $l = \phi$ ) are given as (51):

$$\Delta w_{c_j^v m}(t) = \alpha e_{s_{c_j^v}}(t) \frac{\delta s_{c_j^v}(t)}{\delta w_{c_j^v m}} \quad (6.90)$$

$$\Delta w_{in_j m}(t) = \alpha \sum_{v=1}^{s_j} e_{s_{c_j^v}}(t) \frac{\delta s_{c_j^v}(t)}{\delta w_{in_j m}} \quad (6.91)$$

$$\Delta w_{\phi_j m}(t) = \alpha \sum_{v=1}^{s_j} e_{s_{c_j^v}}(t) \frac{\delta s_{c_j^v}(t)}{\delta w_{\phi_j m}} \quad (6.92)$$

where the internal state error  $e_{s_{c_j^v}}$  is separately calculated for each memory cell as:

$$e_{s_{c_j^v}}(t)^{tr} = y_{out_j}(t) \left( \sum_k w_{kc_j^v} \delta_k(t) \right) \quad (6.93)$$

### 6.7.1 Applications

Long short-term memory networks are well suited for learning and classifying time series when the time lags are very long. This makes them perform better than RNNs and other sequence learning methods. In this section we will briefly review the different applications such as handwriting recognition, reinforcement learning robots, music composition and speech recognition where LSTMs have been successfully used.

In the field of handwriting recognition, work by Marcus et al. (83) has shown that LSTMs outperform all the other methods on the difficult problem of recognizing unsegmented cursive handwritten text. Using a Connectionist Temporal Classification (CTC) objective function they achieved word recognition of 74% compared with 65.4% obtained using HMM based recognition system.

Robots have found their way into surgical procedures which not only automates small subtasks but also greatly reduces the surgeons fatigue and the total surgery time. One of the subtasks in surgical procedures is tying of suture knots during minimally invasive surgery. LSTMs have been used to accomplish this subtask (85) and have proved to be more successful in generalizing the unfamiliar instrument positions during the surgery. Previous work in (84) was hardwired meaning it always repeated the same prescribed motion without any generalization. On the other hand, model using LSTMs was able to perform knot tying based on previous states, i.e. the position of the instruments etc. This made it possible to select future actions appropriately.

LSTMs have also made significant contributions towards music composition. In (38) LSTMs were able to learn a form of blues music and were also able to compose novel melodies in that style. Long short term memory were able to provide a global structure where as RNNs have failed to so do (38). Once the network based on LSTM was able to find the relevant structure it did not drift from it and was able to play blues with good timing and proper structure.

Speech recognition is mainly focused on the contextual information i.e. when classifying a frame of speech data it is also helpful to look at the frames after and before the word segment. Recurrent neural networks themselves are well suited for this task, however, they lack the ability to learn time dependencies more than a few time steps long. In (3) bidirectional LSTMs outperformed both unidirectional and conventional RNNs.

## 6.8 Summary

This chapter gives a detailed overview of neural networks and SVMs which form the basis for our ground work on FACS AU recognition. Here a multilayer perceptron is explained in detail which forms the basis for RNNs. The goal of a neural

network is to approximate an output when given a set of input by adjusting the weights. Back propagation which uses gradient descent is the common algorithm that is used to train neural networks. Computational capability, capacity, convergence and generalization are the four important theoretical properties of neural networks. A detailed section on SVMs then follows. Support vector machines power and flexibility makes them one of the most widely used classifiers in the field of many computer vision applications. The next section describes RNNs and the different architectures available such as first order, second order and LSTMs. Different architectures available and learning algorithms such as BPTT and RTRL are also mentioned, followed by the different applications of RNNs such as speech recognition and forecasting. The problem of vanishing gradient faced by RNNs and steps to avoid the same are discussed. Hidden Markov models are then reviewed along with the assumptions and the basic problems faced by them. Last section in this chapter deals with LSTMs, their architecture, the learning algorithms for LSTMs and different applications such as handwriting recognition, reinforcement learning and speech recognition.

## Chapter 7

# Recognition Using SVMs

In this chapter, we discuss the use of single static images and SVMs for FACS AU recognition. Our study is to investigate the use of image sequences and the improvement in performance provided by them when compared to single images. To provide a framework for this comparison we first experiment with SVMs based FACS AU recognition model that uses single static images as its input. This would also help us in understanding the advantages and disadvantages with the use of different structures of input data. Section 7.1 reviews the Cohn-Kanade database (66) which is used for our research. We also shed some light on the formation and use of single images and image sequences.

### 7.1 Cohn-Kanade Database

For all our experiments we use the Cohn-Kanade AU-coded facial expression database (66). The public version of this database is composed of image sequences from 97 subjects performing a single AU or combination of AUs. Age of the subjects ranges from 18 to 30 years. 65% of them are female, 15% African-American and 3% Asian. The subjects are asked to directly face the camera and perform a series of 23 facial displays that may be single AU or combination of AUs. Each sequence begins with a neutral display (face containing no AU) and ends with a target display (face containing an AU at its peak intensity). Image sequences were then digitized into 640\*480 pixel arrays with 8-bit precision for

gray scale images. Over 17% of the data was comparison coded by a second certified FACS coder.

We use a subset of the Cohn-Kanade database consisting of 300 video sequences from 78 human subjects for the recognition of upper face AUs and a total of 258 video sequences from 67 subjects for the recognition of lower face AUs. Samples from subjects that do not have the minimum number of frames for the formation of an image sequence are avoided. The eye coordinates are manually located and are then used to align the frames and crop the face region in all the frames in case of an image sequence. On contrary, eyes are located in every frame used with SVMs. Further cropping is performed depending on the presence of an AU, i.e. either the upper or the lower half of the face region. The average number of samples for training and testing the upper and lower face AUs for both the classifiers (SVMs and RNNs) are given in Table 7.1 and Table 7.2 respectively.

AUs	AU0	AU1	AU2	AU4	AU5	AU6	AU7
<i>Train</i>	200	142	102	70	90	90	48
<i>Test</i>	85	70	50	38	44	46	26

**Table 7.1:** Number of samples for train and test sets used for upper face AUs

AUs	AU0	AU15	AU17	AU20	AU25	AU27
<i>Train</i>	182	70	102	64	152	90
<i>Test</i>	90	34	54	32	82	44

**Table 7.2:** Number of samples for train and test sets used for lower face AUs

### 7.1.1 Image Sequences vs Single Static Images

The format of data that is used for the recognition of FACS AUs can be broadly divided into two major categories: one is the use of single static images and the other is the use of a sequence of frames. The use of single static images is simple where one single snap shot is used to depict a neutral face or an action (in our work it relates to an AU) at its peak intensity. However a slight variant of this



form of data is to extract the first two frames representing a neutral expression and the last two frames representing an AU at its peak intensity as shown in Fig. 7.1. Each frame from this sequence is then used as a single static image for training the classifier (13). Even though use of this form of data has been proved to be successful, it lacks a time variant component. Even the approaches using entire image sequences shown in Fig. 7.1 will only be able to gain information about the final changes that took place in the facial region due to activation of an AU when compared to the neutral face.



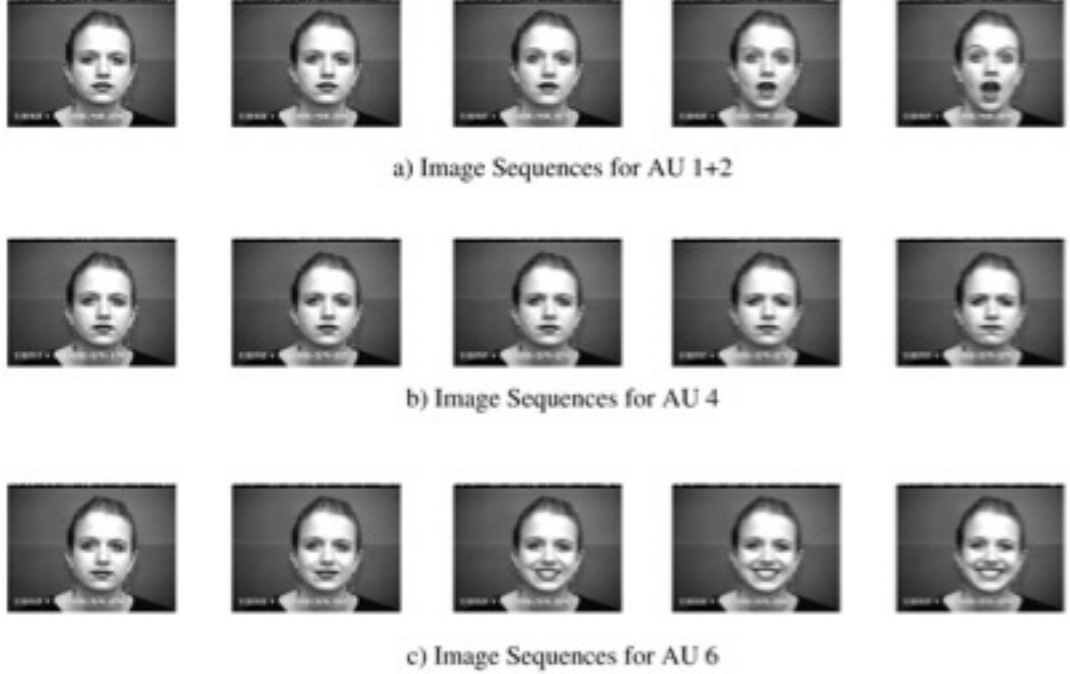
**Figure 7.1:** Image sequences with the first and last two frames from the video clip. *Pictures courtesy:* Cohn-Kanade Database (66).

In contrast, an image sequence formed using all or a part of frames present in the video clip provides a better understanding of an AU activation. This information is not available when only a single image depicting an AU at its peak intensity is studied. Fig. 7.2 shows the above difference in the formation of data. Fig. 7.2 contains time variant sequences formed using five image frames, but more or less frames can be used depending on the availability and activation of the AU under consideration. A frame sequence depicts the formation of an AU from its absence to its presence (usually at its peak intensity). This provides crucial information for understanding the appearance changes.

Any format that one uses, a positive sample always contains an AU that needs to be classified and a negative sample may contain all the other AUs as well as neutral faces. In all our experiments, we classify an AU irrespective of the presence or absence of other AUs.

### 7.1.2 11 AU and Their Description

Even though FACS defines 44 different AUs, we limit ourselves to a set of 11 AUs for our initial experiments. The selection is based on the availability of minimum

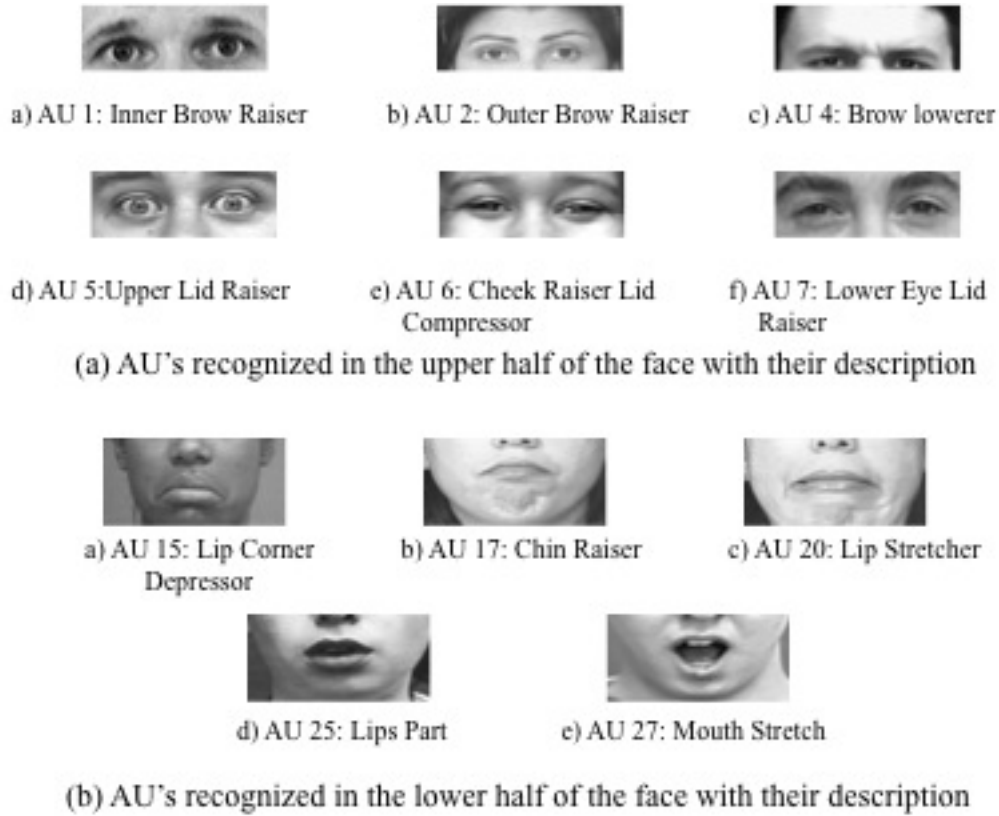


**Figure 7.2:** Image sequences for AU 1+2, 4 and 6 from the Cohn-Kanade database (66). Note that the image sequences also depict other AUs acting simultaneously.

number of samples per AU in the database and also their popularity among other researchers in the field of FER. The classified AUs and their descriptions are given in Fig. 7.3.

## 7.2 FACS AU Recognition using SVMs

The use of SVMs was studied in both facial expression and FACS AU recognition. The main reason behind SVMs being so popular for FER is their ability to handle high dimensionality of the Gabor representations without any affect on the training time for kernel classifiers (10). In our work, we performed a set of experiments involving the same dataset that we intended to use with RNNs and studied the performance provided by SVMs. This section is intended to make a comparative analysis on the use of single static images and that of using image



**Figure 7.3:** 11 FACS AU's and their description, *Pictures in Figure (a) and (b) are courtesy of Carnegie Mellon University Automated Face Analysis group, <http://www-2.cs.cmu.edu/afs/cs/project/face/www/facs.htm>*

sequences.

Even though references exist based on the same Cohn-Kanade database, most of them use large number of negative samples when compared to the number of positive samples for both training and testing. However, in our experiments we restricted ourselves to the use of equal number of positive and negative samples to keep the balance. Use of large datasets is also prevalent which often leads to use of more number of samples from each subject, with a high probability of using the same samples more than once. Keeping the data constraints faced while collecting the image sequences for use with RNNs we restricted ourselves to the use of two samples per subject; a neutral frame and a frame depicting an AU at

its peak intensity. This is to keep the data usage same for both SVMs and RNNs. Formation of more than one image sequence from each subject for use with RNNs also becomes difficult under circumstances where different AUs are activated at different instances in a video sequence.

Performance measure is a very important factor which provides a benchmark as to how the results using different methods compare. However, in the FER literature, researchers have reported their results using ROC area, equal error rate, percent correct, hit rate and n-alternative forced choice measure. In this thesis we provide the results in terms of recognition rate (RR) or hit rate (HR) which is the number of samples classified correctly by the total number of samples. The number of samples classified incorrectly by the total number of samples is given as the false alarm rate (FAR).

### 7.2.1 Data Collection

The data used with SVMs consists of single static images. From each video sample in the database we collected the first two and last two frames depicting an AUs in action as shown in Fig. 7.1. The first two frames usually depict a neutral face and the last two frames depict all the AUs that were active and at their peak intensity. Each frame is then used as a sample in the dataset. The Cohn-Kanade database contains video sequences where more than one AU is performed. In our study, however we only recognized the AU of interest irrespective of other active AUs in the frame.

### 7.2.2 Preprocessing

In this work, we performed the following preprocessing steps: 1. manual detection of facial features such as eye centers, 2. face normalization using these eye centers, and 3. face detection and segmentation. Eye coordinates were located in all the frames. These coordinates were then used to align the eye centers in the respective frames. Face detection and cropping is performed using MPISearch (48) which reduces the image size to 64\*64 pixels. Further segmentation is performed to retrieve either upper or lower half of the face region as shown in Fig. 7.4. Segmentation of face into regions is based on the knowledge that the AUs

active in the upper face have little or no effect on the appearance changes in the lower face region and vice versa (43).



**Figure 7.4:** Cropped lower face regions

### 7.2.3 Feature Extraction

Over the last decade, many feature extraction techniques have been used for the recognition of facial expressions. In (37) it was shown that Gabor filters are one of the appearance-based methods that perform better than other appearance and geometry based methods. However, Gabor filters suffer from the disadvantage of incurring high computational costs.

Gabor decomposition of an image is computed by filtering the input image with a Gabor filter tuned to a particular frequency and orientation. In the field of FER multiple Gabor filters tuned to different characteristic frequencies and orientations are used. Deng et al. (36) experimented with the use of different combinations of frequencies and orientations and their effect on recognition performance. Here, we chose five frequencies and eight orientations as in (111). In order to further reduce the dimensionality of the input, the output responses of the 40 Gabor filters were down-sampled by a factor of 16 and normalized to unit length (37).

### 7.2.4 Classification

One classifier is trained to detect the presence or absence of each AU. Action units are recognized regardless of their occurrence in combination with other AUs. We did not attempt to account for non-additive AU combinations that occur in the database. An AU is said to be active/present if output of the trained classifier is greater than or equal to zero.

Owing to a small number of samples for each AU we perform a three-fold cross validation. The data subset for each AU is divided into three parts; two parts combined form a training set and the remaining one part is used for testing. It is made sure that all the parts are disjoint with respect to the subjects. The recognition rate or the hit rate in each fold is calculated as the number of samples correctly classified by the total number of samples. The resultant recognition rate for each AU is the average over all the three folds.

### 7.2.5 Results

The performance of our SVM based model for the recognition of six upper face and five lower face AUs are given in Table 7.3 and Table 7.4 respectively. Throughout our research we keep upper and lower face AUs separate in order to access the affect of different algorithms on different face regions.

AUs	Recognition Rate(%)	False Alarm Rate(%)
AU1	85.23	9.08
AU2	89.49	6.72
AU4	77.72	6.84
AU5	79.09	7.62
AU6	76.03	7.24
AU7	73.59	11.87
<b>Average</b>	<b>80.20</b>	<b>8.23</b>

**Table 7.3:** Recognition results for the six AUs detected in the upper half of the face using SVM as a classifier

### 7.2.6 Discussion

Support vector machines achieved an average recognition rate of 79.47% with a false alarm rate of 9.22% for the 11 FACS AUs. Bartlett et al. (11) reported a recognition rate of 88% using SVMs for recognizing six basic emotions. They used Gabor wavelets to extract the features from single static images which were then fed to a SVM based classifier for training and testing generalization to new

AUs	Recognition Rate(%)	False Alarm Rate(%)
AU15	70.12	15.24
AU17	69.47	11.73
AU20	81.38	6.39
AU25	88.40	10.02
AU27	84.29	7.59
<b>Average</b>	<b>78.73</b>	<b>10.20</b>

**Table 7.4:** Recognition results for the five AUs detected in the lower half of the face using SVM as a classifier

subjects. Towards classifying facial actions Bartlett et al. (13) reported a hit and false alarm rate of 80.1% and 8.2% respectively using Gabor filters and Adaboost. Our SVM based system did not show similar performance as in (11) even though the process was the same. However, it should be noted that our model was limited to the use of one single sample from each subject. The other differences include the use of whole face image, different set of Gabor frequencies and different kernel functions used in (11) and (13). Benchmarking the results would be difficult as different researchers used different datasets, image sizes, kernel functions, Gabor frequencies and labelling of feature points.

The results using SVMs will be used for benchmarking the results that will be discussed in Chapter 8 which studies the use of RNNs for FACS AU recognition.

## 7.3 Summary

In this chapter, we provided a brief overview of the Cohn-Kanade database. A comparison between the formation of single static images and image sequences was discussed. The following sections shed some light on the 11 FACS AUs that were classified. The use of SVM with single static images for the classification of FACS AUs is then discussed. The feature extraction was performed using Gabor filters. The experiments carried here act as a benchmark for the experiments carried out using RNNs and image sequences.

# Chapter 8

## Recognition Using RNNs

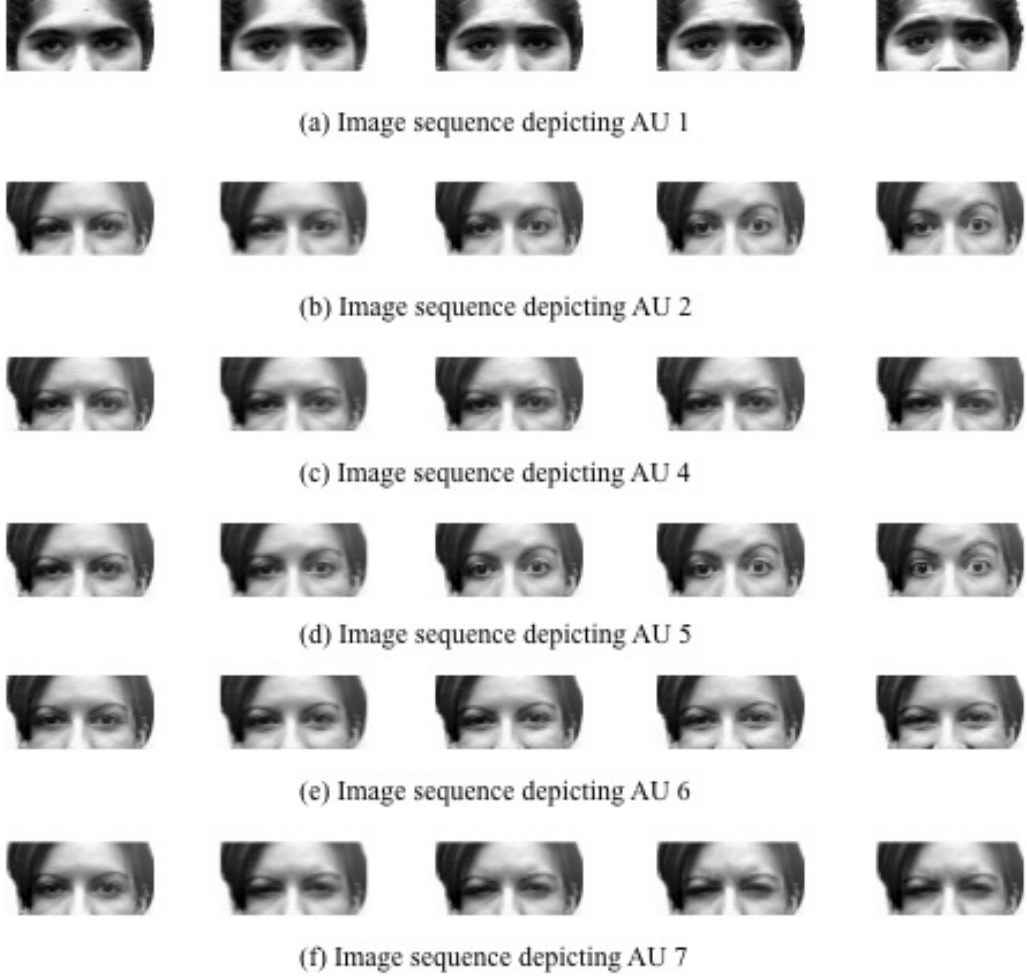
### 8.1 Introduction

In this chapter, we discuss our contributions to the field of FACS AU recognition. Our study is to investigate the use of image sequences instead of single static images. We study the use of RNNs to handle the time variant data in the form of frame sequences depicting an AU. The effect of number of hidden nodes is studied to provide an optimal network. Further experimentations involving the use of different dimensionality and feature selection techniques shed some light on the compatibility issues between RNNs and these methods. Network optimization plays a crucial role towards improving the performance of RNNs and we experiment with one of the widely used method: weight decay. At the end of this chapter we generalize our system towards recognizing most of the FACS AUs depicted in the Cohn-Kanade database. We then provide a comparative analysis on the use of image sequences and whether they are able to provide a better classification than the use of single static images discussed in chapter 7.

#### 8.1.1 Image Sequences Depicting AUs

The formation of image sequences depicting an AU forms basic data for our FACS AU recognition model. To formulate the sequences it is very important to understand the necessary muscle actions and how a particular AU increases in its



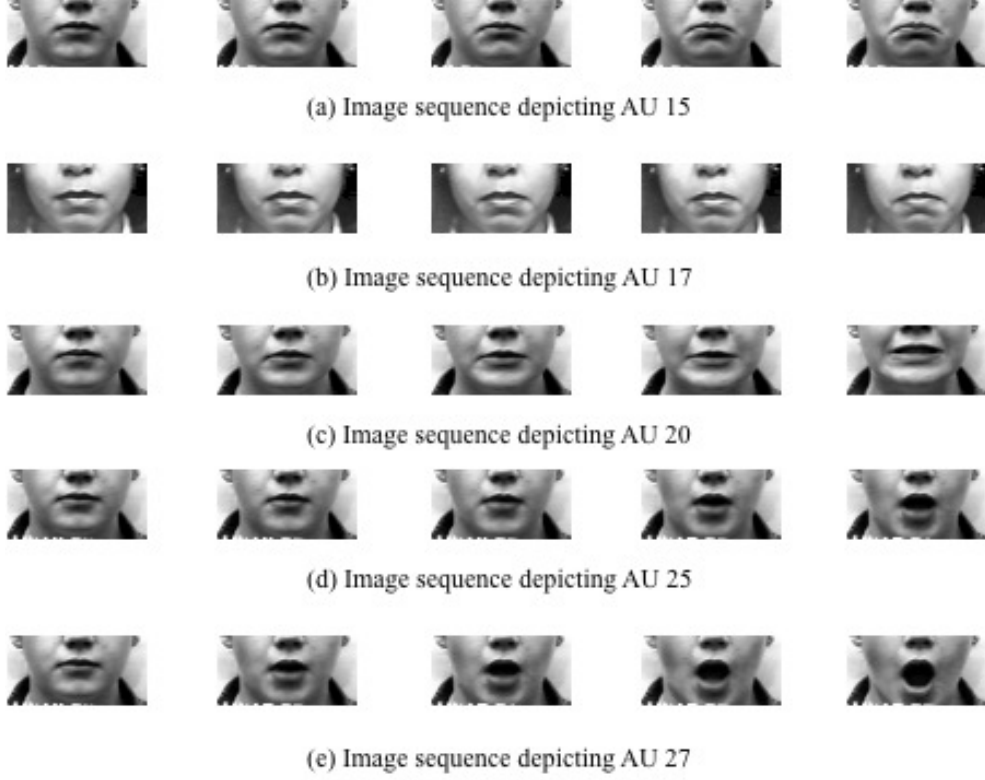


**Figure 8.1:** Image Sequences Depicting the Upper Face AUs Classified

intensity with relative to the time. Fig. 8.1 and Fig. 8.2 depicts the upper and lower face AUs respectively with the help of five frames each.

## 8.2 Baseline FACS AU Recognition Using RNN

Here we investigate the use of RNNs as a classifier for FACS AU recognition. Fig. 8.3 gives the basic structure of our FACS AU recognition model. Our model is based on the use of Gabor filters for feature extraction and RNN for classification which to our knowledge has never been used for FACS AU classification.



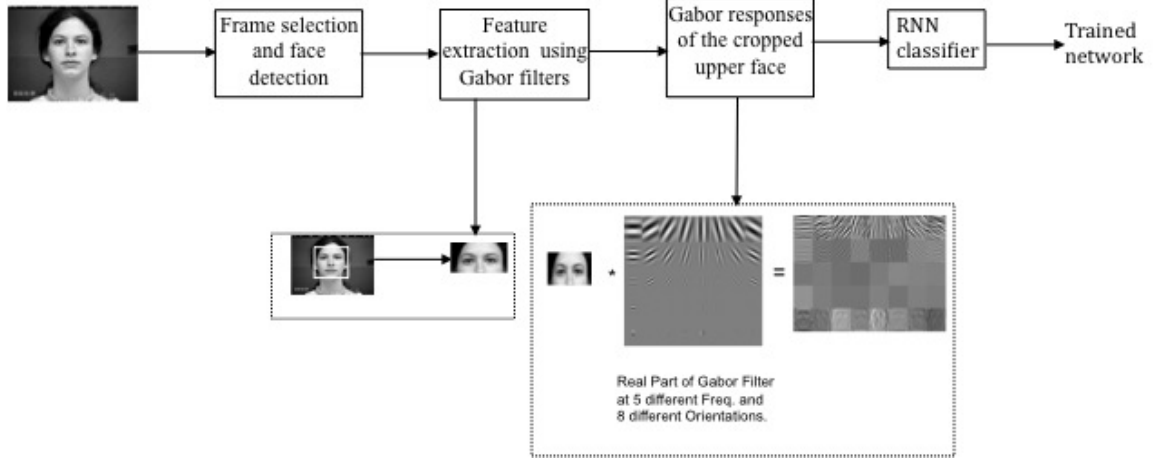
**Figure 8.2:** Image Sequences Depicting the Lower Face AUs Classified

This section is aimed at investigating the baseline recognition using RNNs. Classification process for baseline recognition involves no optimization or feature selection techniques. In the following sections we discuss the data collection, preprocessing and classification process.

### 8.2.1 Data Collection

The data for use with RNNs consists of a sequence of images as shown in Fig. 8.1. From each video sample in the database, we collect a set of frames that best depict an AU under consideration. We collect one such sequence from each subject for one single AU. However, Cohn-Kanade database contains video sequences where more than one AU is performed. So the same image sequence or an image sequence with a difference of 1-2 frames will be used for some AUs. In our study

## 8.2 Baseline FACS AU Recognition Using RNN



**Figure 8.3:** Overview of the FACS AU classification Model

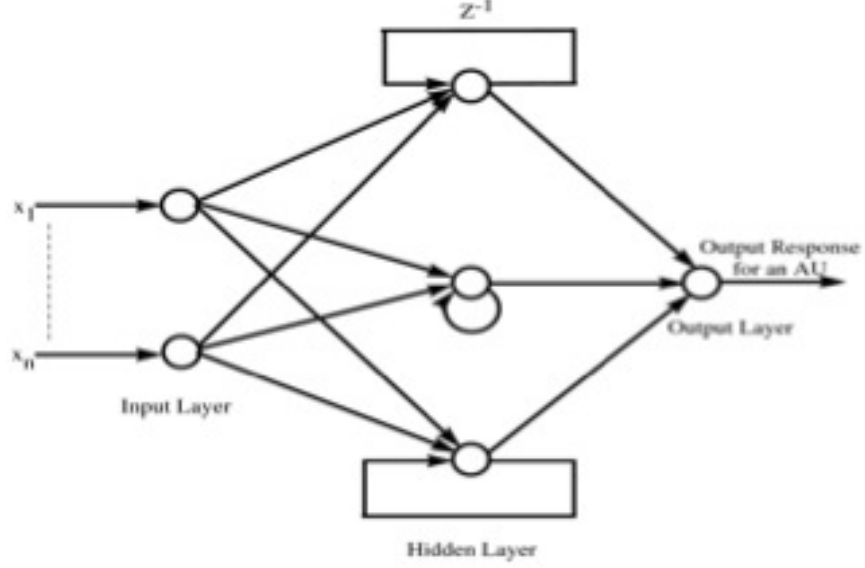
we only recognize the AU of interest irrespective of other active AUs present in the sequence.

### 8.2.2 Preprocessing

In this work we perform the preprocessing steps mentioned in Section 7.2.2. However, a slight difference exists in locating the eye coordinates. Unlike in SVM based classification, here we locate the eye coordinates only in the first frame for each sequence. These coordinates are then used for all the consecutive frames. This maintains the minor head movements in the image sequences which were absent with the use of single static images.

### 8.2.3 Feature Extraction

The process of feature extraction using a Gabor jet is similar to the one discussed in Section 7.2.3. A vector of 5120 Gabor coefficients will represent the features extracted from each frame in the image sequence. Each frame in the sequence will have a similar feature vector obtained using Gabor filters.

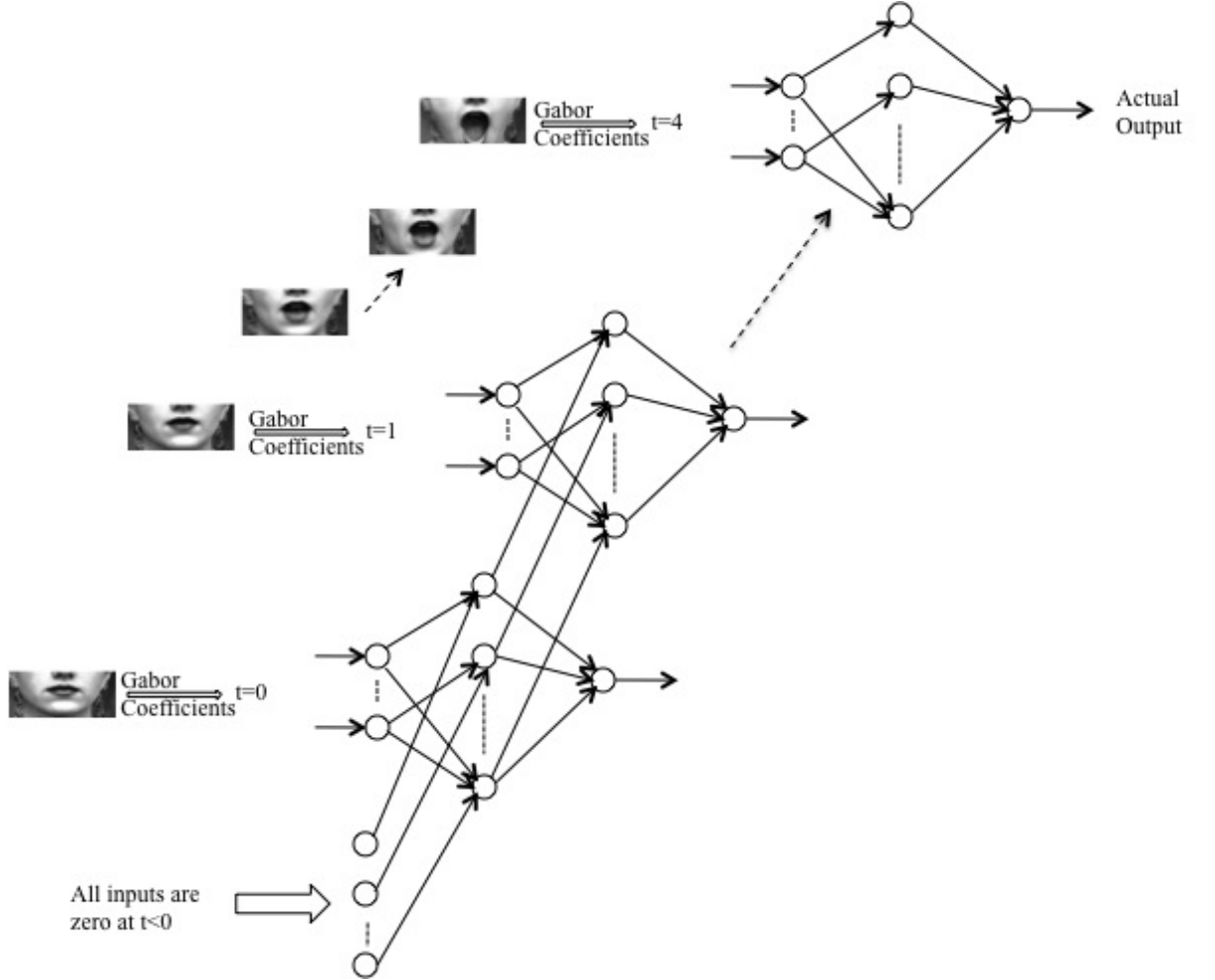


**Figure 8.4:** Elman Recurrent Neural Network for the Recognition of FACS AUs

### 8.2.4 Classification

An Elman RNN shown in Fig. 8.4 is used for the recognition of each FACS AU. Fig 8.4 depicts the same RNN expanded over time  $t$  where frame 1 is presented at time  $t=0$ , frame 2 at time  $t=1$  and so on.

The number of input units in the network is equal to the total number of Gabor coefficients (i.e. 5120). The number of output units is 1 which is trained to output a “1” if the target AU is present and a “0” if it is absent. All the intermediate outputs from the output unit are made equal to zero as we are only interested in the output (AU presence/absence) once all the frames are fed. The number of context units is equal to number of input units. For these experiments the optimal number of hidden units is found to be 15 through experimentation. Details on this will be given in Section 8.3. The network weights are set to random initial values in the range of  $[-1,1]$ . The maximum number of epochs which is the presentation of all the training samples to the network, is set to 500 and a learning rate of 0.01 is used. The training stops when the pre-defined system error is reached or the learning process continues until the maximum epochs are elapsed. No further calculations are done at the output layers; a value



**Figure 8.5:** Elman Recurrent Neural Network Unfolded in Time

0.5 or greater is taken as the target AU present and a value less than 0.5 is taken as target AU absent.

### 8.2.5 Results

We achieved an average recognition rate of 83.51% with a false alarm rate of 6.68% for the six upper face AUs and an average recognition rate of 81.98% and 8.53% false alarm rate for the five lower face AUs. The average recognition rate of 82.75% was achieved for the 11 FACS AUs. The individual results for upper

## 8.2 Baseline FACS AU Recognition Using RNN

---

and lower face AUs are given in Table 8.1 and Table 8.2, respectively.

AUs	Recognition Rate(%)	False Alarm Rate(%)
AU1	89.96	5.01
AU2	92.66	4.20
AU4	81.50	7.54
AU5	83.59	6.00
AU6	77.56	8.20
AU7	75.80	9.18
<b>Average</b>	<b>83.51</b>	<b>6.68</b>

**Table 8.1:** Recognition results for the six AUs detected in the upper face using RNN as a classifier

AUs	Recognition Rate(%)	False Alarm Rate(%)
AU15	74.71	12.97
AU17	72.33	10.45
AU20	84.27	4.04
AU25	91.10	9.30
AU27	87.50	5.86
<b>Average</b>	<b>81.98</b>	<b>8.53</b>

**Table 8.2:** Recognition results for the five AUs detected in the lower face using RNN as a classifier

### 8.2.6 Discussion

We achieved an average baseline recognition rate of 82.57% with a false alarm rate of 7.61% for the 11 FACS AUs. The average recognition and false alarm rate achieved using SVMs was 79.47% and 9.22%, respectively. The percentage increase in the recognition rate was over 3% and this increase supports the use of RNNs for FACS AU recognition. The RNN based model clearly carries the advantage of better classification with a lesser false alarm rate. The use of image sequences was successful in providing more information than the single static

images which was also reflected in the decrease of FAR. This supports our hypothesis that RNN based FACS AU recognition using image sequences performs better than the use of SVMs and single static images.

In performance analysis, statistical significance also plays an important role. Even though there was a considerable increase in the performance in terms of recognition rate, yet it was not statistically significant. One reason for this would be the small number of samples present for each AU. Except for a couple of AUs, all others had a total of 80 or less samples for training and testing.

To add more depth to our findings, we do a comparative analysis with other recognition approaches found in the literature. The average recognition rate reported by (112) using geometric features alone was 84.7% and using appearance-based method alone was 32%. Our average recognition rate was similar to the recognition rate by the use of geometric approach alone. A difference of 1-2% in average recognition rate can be attributed to the use of different networks stopped at different times. The recognition rate when compared to Gabor filters without any preprocessing in (112) also signifies the use of image normalization as preprocessing step, which substantially increases the recognition rate. However, our approach failed to reach high recognition rates stated in (9) which might be attributed to the use of huge number of samples for training, feature selection used in (9) and the difficulty faced for general training of RNNs.

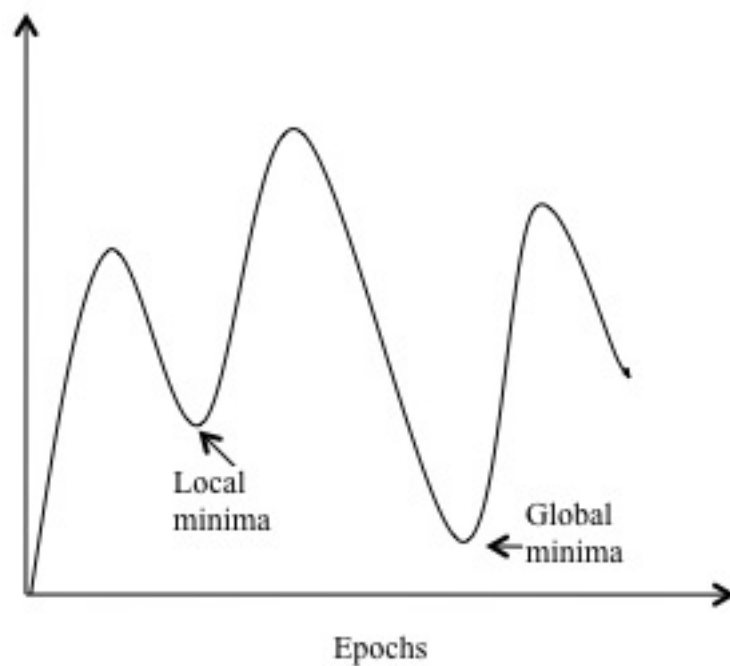
## 8.3 Effect of Number of Hidden Nodes

In this section, we perform tests to know the effect of number of hidden nodes on the models performance. There is no algorithm that could estimate the optimal number of hidden units for neural networks. Determining the optimal number of hidden nodes is best-done using trial and error method. In RNN and all other variants of neural networks, the number of hidden nodes used for a fixed number of input and output units plays a crucial role in the performance, since the number of hidden nodes greatly influences the number of network weights present in the network. We run experiments with the number of hidden nodes ranging from 1 to 20 for both upper face and lower face AUs. The objective was to find the lowest number of hidden nodes that produces the lowest possible error in the test

### 8.3 Effect of Number of Hidden Nodes

---

data. Too few hidden nodes will produce a network where the relations between the variables in the input data are not fully learned. On the other hand, too many hidden nodes will over-fit the training data producing poor results when presented with the test data. In the experiments where two network models with different number of hidden nodes gives similar performance with no significant difference, the network model with fewer number of hidden nodes is considered. It should be noted that the number of hidden units which was found to be best for our model may be just one of the local minima. A general graph representing such a situation is depicted in Fig. 8.6 . However, finding the global minima for any model is difficult in many real world scenarios. The general rule of thumb is to find a local minima and see if the recognition model is performing well enough for a given application.

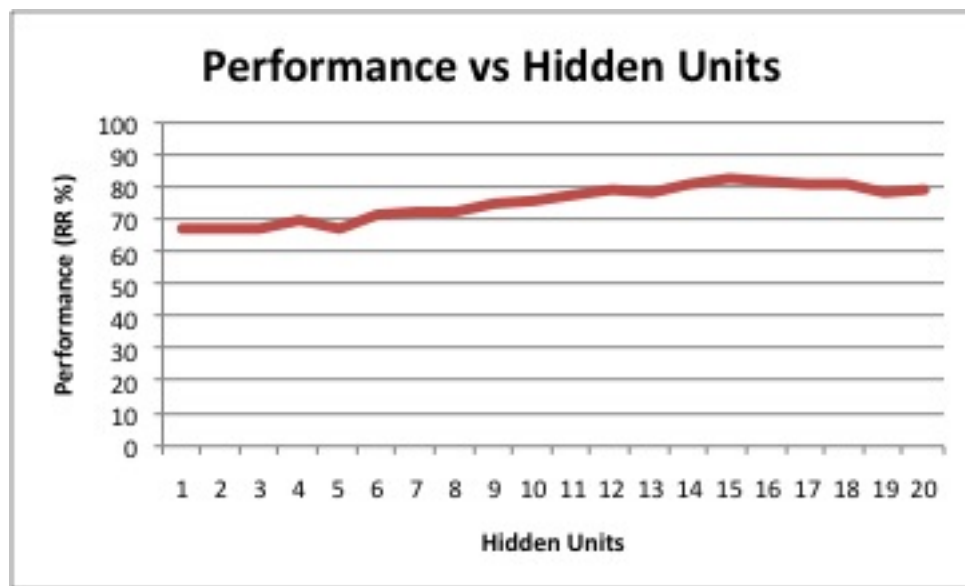


**Figure 8.6:** System error vs. Epochs.



### 8.3.1 Results

The results depicting the performance of our recognition model vs. the number of hidden nodes is illustrated in Fig. 8.7. The recognition rate on the y-axis is the average recognition rate for the 11 AUs that were classified. The x-axis are the number of hidden units from 1 to 20. After 15, the average recognition rate continuous to decrease. The model was tested till 20 hidden units.



**Figure 8.7:** Effect of number of hidden units on the performance of the model for the 11 FACS AUs

### 8.3.2 Discussion

These experiments study the relation between the number of hidden units and the corresponding performance of FACS AU recognition model. For RNN there is no set rule for finding the optimal number of hidden units. The approach is a simple trial and error where the network is trained and tested for a specific number of hidden units. The same procedure is performed using different values and the models performance is noted. For our FACS AU recognition model we have tested

from 1 to 20 hidden units. At 15 hidden units the FACS AU recognition model performed better.

## 8.4 Feature Selection

### 8.4.1 Frequency Selection

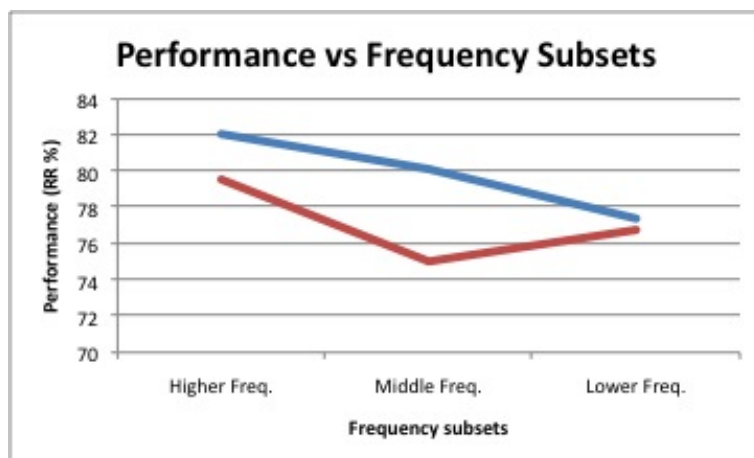
Frequency selection is one of the widely used methods for reducing the high dimensional response vectors generated by Gabor filters. Studies in (37) and (111) emphasize that not all frequency scales are required for the recognition of AUs. In (37), the set of three highest frequency scales gave a similar performance to that of using all the frequency scales. However, in (111) the set of three middle frequency scales performed better for the recognition of upper face AUs.

In our experiments, we study the effects of higher and lower frequencies on the performance. Following Tian et al. (112), we divide the five spatial frequencies  $k = (\frac{\pi}{2}, \frac{\pi}{4}, \frac{\pi}{8}, \frac{\pi}{16}, \frac{\pi}{32})$  into three subsets where first subset consists of first three frequencies  $k = (\frac{\pi}{2}, \frac{\pi}{4}, \frac{\pi}{8})$ , second subset consists of middle three frequencies  $k = (\frac{\pi}{4}, \frac{\pi}{8}, \frac{\pi}{16})$  and the third subsets consist of the last three frequencies  $k = (\frac{\pi}{8}, \frac{\pi}{16}, \frac{\pi}{32})$ .

Face Region	Higher Frequencies RR (%)	Middle Frequencies RR (%)	Lower Frequencies RR (%)
Upper Face AUs	82.07	80.10	77.40
Lower Face AUs	79.56	75.03	76.84

**Table 8.3:** Recognition rates using higher, middle and lower set of frequencies for the upper and lower face AUs

The average recognition rates for the six upper face AUs and the five lower face AUs using the three subsets of frequency components are given in Table 8.3. A graph depicting the same is shown in Fig. 8.8. From the graph, it is clear that the higher frequencies contain more information than the lower set of frequencies for the 6 upper face AUs. For the lower face AUs, however, the set of last frequencies performed better than the set of middle frequencies. In this study, we confirm



**Figure 8.8:** Significance of selection of spatial frequencies for both upper and lower face AUs

the results reported in (37) that the set of three highest frequency scales in case of upper face AUs leads to similar performance as that of using all the frequency components. In this case, there is no significant degradation in the generalization capability of the classifier. However, the performance obtained using the lower frequencies was the least. In case of lower face AUs the set of lower frequencies performed better than the set of middle frequencies. This emphasizes that not all the regions in the face provide similar information with a set of frequency components. The performance by a set of frequency components is dependent on the face region and also the AU under consideration.

The use of higher frequencies alone reduced the recognition rate from 83.51% (using all the frequency components) to 82.07% for the six upper face AUs and from 81.98% to 79.56% for the five lower face AUs. The slight drop in the recognition rate shows that the higher three frequencies can be used for the recognition of the upper face AUs without much degradation in the models performance. The selection of a subset of frequencies also reduces the number of input units from 5120 to 3072 which is considerable reduction in the dimensionality.

The recognition and false alarm rates for the six upper AUs with the use of upper frequencies are given in Table 8.4 and that of five lower face AUs are given in Table 8.5. Experiments are performed using three-fold cross validation similar

to our previous experiments. It should be noted that the models configuration has not been changed except for the number of input units.

AUs	Recognition Rate(%)	False Alarm Rate(%)
AU1	88.6	6.36
AU2	91.33	2.66
AU4	81.62	8.27
AU5	83.56	8.23
AU6	75.36	9.71
AU7	72.00	16.00
<b>Average</b>	<b>82.07</b>	<b>8.53</b>

**Table 8.4:** Recognition and False Alarm Rate for the upper face AUs with the first three higher frequencies

AUs	Recognition Rate(%)	False Alarm Rate(%)
AU15	68.05	10.3
AU17	70.75	13.69
AU20	83.42	3.0
AU25	88.2	3.64
AU27	87.50	5.86
<b>Average</b>	<b>79.56</b>	<b>7.30</b>

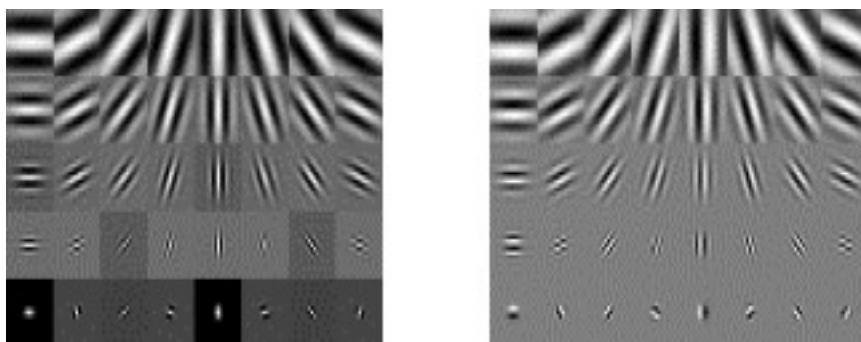
**Table 8.5:** Recognition and False Alarm Rate for the lower face AUs with the first three higher frequencies

The above results stress on a use of a method that includes all the scales of a Gabor jet, and yet reduces the dimensionality. The use of such an approach should also be able to improve or at least retain recognition performance. Deng et al. (36) proposed one such approach know as local Gabor filters.

### 8.4.2 Local Gabor Filters

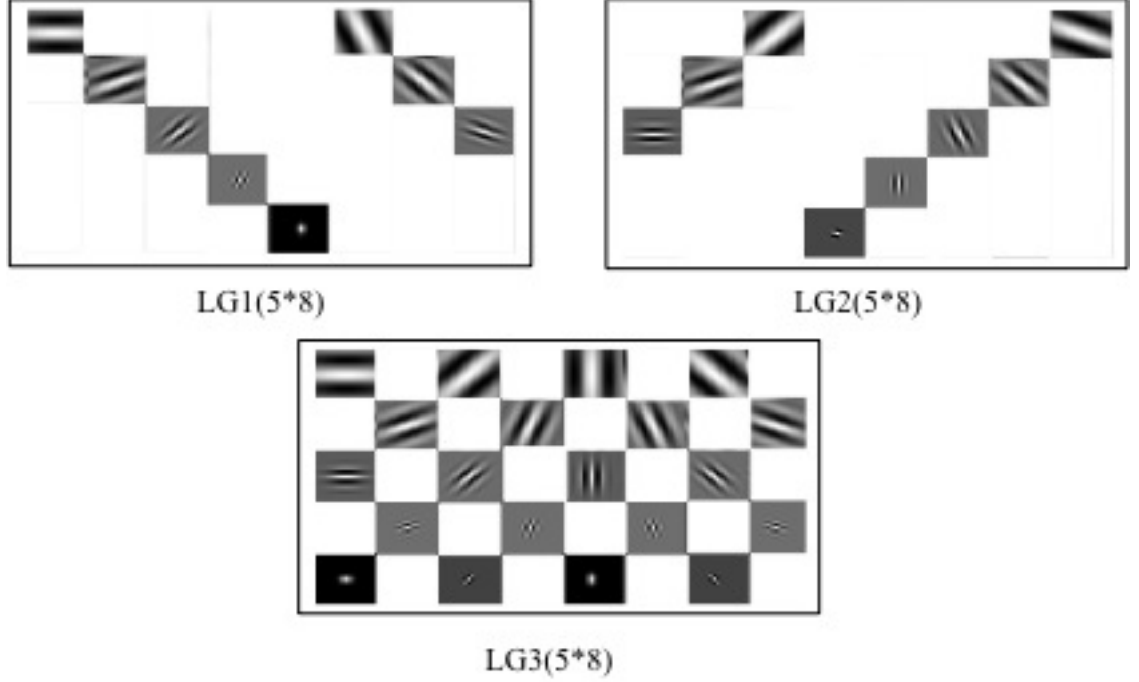
Gabor filter responses increase the size of the feature vector by a factor of 40 when five spatial frequencies and eight orientations are used. So a 64\*64 pixels

image transforms into  $64*64*40=163840$  number of Gabor coefficients. This is huge input data for any classifier. The previous method of using only a subset of frequencies does reduce the dimensionality of the data to some extent, but also decreases the performance by few percentage points emphasizing on the fact that all scales are important.



**Figure 8.9:** Real and imaginary parts of a 5\*8 Gabor jet

Looking at the Gabor jets shown in Fig. 8.9, however, it is evident that Gabor jets contain some redundant data. Gabor representations are very much similar using filters with the same orientations with two neighboring frequencies. Making use of this information Deng et al. (36) proposed a novel approach of local Gabor filter bank with a part of the entire frequencies and orientations. They denoted these local Gabor filters as LG ( $mxn$ ) where  $m$  is the total number of spatial frequencies and  $n$ , the total number of orientations. We use the same notation as in (36). To reduce the dimensionality without degrading the recognition rate local Gabor filters should cover all the frequencies and orientations but only selecting one frequency for each orientation or increase the interval between the neighboring frequencies with the same orientation (36). They proposed three such local Gabor filters; LG1 ( $mxn$ ), LG2 ( $mxn$ ) and LG3 ( $mxn$ ). LG1 ( $mxn$ ) is formulated by increasing the frequency repeatedly from minimum to maximum and orientation is incremented by one each time. LG2 ( $mxn$ ) is same as LG1 ( $mxn$ ) with a decrease in frequency from maximum to minimum. And for LG3 ( $mxn$ ), the responses are selected with an interval of one between any two filters. The response feature vectors for LG1 (5x8), LG2 (5x8) and LG3 (5x8) are shown



**Figure 8.10:** Local Gabor filters a) LG1 (5x8) b) LG2 (5x8) and c) LG3 (5x8)

in Fig. 8.10. Memory requirements of global and local Gabor filters are given in Table 8.6.

Classification is performed using an Elman RNN. Due to drastic reduction in the number of input nodes we ran experiments to obtain the number of hidden units that would give a better performance. The number of hidden units that performed best with the three variants of local Gabor jets is given in Table 8.6. To record the recognition rate and false alarm rate of our model for classifying an AU we perform three-fold cross validation. The average recognition rates for the six upper face AUs using the three variants of local Gabor filters is given in Table 8.7. From the results in Table 8.7 it is clear that the over all recognition rate with variant LG1 (5\*8) is less than the use of all the 40 Gabor coefficients. The performance was similar in the case of LG2 (5\*8) and G (5\*8). There was an increase of nearly 1% in the overall recognition rate in case of LG3 (5\*8) variant when compared to G (5\*8). This can be attributed to decreased number of input units and removal of redundant data. The individual recognition and false alarm

Gabor Jet	Original dimension (64*64)	Upper face dimension (64*32)	Sub sampling dimension (16*8)	No. of hidden units
G (5x8)	163840	81920	5120	15
LG1 (5x8)	32768	16384	1024	10
LG2 (5x8)	32768	16384	1024	10
LG3 (5x8)	81920	40960	2560	10

**Table 8.6:** Memory requirements of global and local Gabor filters and the number of hidden units used by RNN based FACS AU recognition model

rates using LG3 variant for the six upper face AUs are given in Table 8.8.

Local Gabor Configuration	Recognition Rate(%)	False Alarm Rate(%)
G (5x8)	83.51	6.68
LG1 (5x8)	79.9	10.21
LG2 (5x8)	82.61	7.59
LG3 (5x8)	84.56	6.79

**Table 8.7:** Recognition Rate and False Alarm Rate for the three variants of local Gabor filters for the six upper face AUs

Similar experiments were performed for recognizing the five lower face AUs. Table 8.9 shows the results using all the 40 Gabor coefficients and that of using the three variants of local Gabor filters. In this case none of the Gabor variants were able to outperform the use of all the 40 Gabor coefficients. Table 8.10 shows the individual AU recognition and false alarm rates for the five lower face AUs using LG3 (5\*8) variant.

In general, the three variants of local Gabor filters reduce the dimensionality by several percentage points. LG1 and LG2 variant reduce the dimensionality by 80% where as LG3 variant reduces the dimensionality by 50%. In case of upper face AUs; LG3 variant was able to increased the performance by over 1%. However, in the lower face region, performance of LG3 was nearly 1% lower to that of using all the 40 Gabor coefficients. This may be an indication of

AUs	Recognition Rate(%)	False Alarm Rate(%)
AU1	90.4	4.55
AU2	92.05	1.97
AU4	883.2	9.4
AU5	87.09	5.6
AU6	77.56	7.44
AU7	77.04	11.8
<b>Average</b>	<b>84.56</b>	<b>6.79</b>

**Table 8.8:** Recognition and False Alarm Rate for the six upper face AUs with LG3 (5\*8) variant of local Gabor filters

Local Gabor Configuration	Recognition Rate(%)	False Alarm Rate(%)
G (5x8)	81.98	8.53
LG1 (5x8)	75.2	9.1
LG2 (5x8)	78.5	8.21
LG3 (5x8)	80.85	7.76

**Table 8.9:** Recognition Rate and False Alarm Rate for the three variants of the Local Gabor filters for the five Lower Face AUs using RNN

the complexity involved in classifying lower face AUs because of more complex skin deformations (43). However, it is significant to note that in both cases LG3 variant outperformed the performance obtained by frequency scale selection, LG1 and LG2 variants of local Gabor filters. This strengthens the fact that all the frequency scales are important for a better generalization performance. However, how well a particular variant performs depends on the classification model used and the AUs that are classified.

### 8.4.3 Feature Selection using PCA

Principle component analysis is by large one of most widely used technique for feature selection and also for dimensionality reduction. It has been successfully used as a feature selection technique when input data is huge. Principle com-



AUs	Recognition Rate(%)	False Alarm Rate(%)
AU15	73.4	8.68
AU17	73.08	17.2
AU20	82.17	6.25
AU25	90	4.23
AU27	86.02	2.28
<b>Average</b>	<b>80.85</b>	<b>7.76</b>

**Table 8.10:** Recognition and False Alarm Rate for the five Lower Face AU with LG3 (5\*8) of local Gabor filters using RNN

ponent analysis has been used to select important features from the huge data generated by Gabor filters (36). However, performance of this method varied based on the classifier in use and the actions classified (facial expressions/ FACS AUs) (11). Here the set of experiments are aimed at studying the use of PCA in conjunction with RNNs as a classifier for FACS AU recognition.

The performance of Gabor filters for expression recognition is greatly affected by the lighting effects (36). To control the illumination effect we use histogram normalization following (36) for our FACS AU recognition model. After the face regions are cropped we perform histogram normalization on all the frames of an image sequence. These images are then use to extract the Gabor coefficients. There will be a total of 5120 Gabor coefficients when considering all the 40 Gabor filters. We apply PCA on these 5120 inputs. Taking all the PCA components generated we train the recognition model until the termination condition is met.

We applied PCA on the Gabor coefficients for the recognition of the six upper face AUs. The best performance was 78.6% using 30 to 40 highest Eigen vectors. The results are given in Table 8.11. The average recognition rate has reduced from 83.51% without the use of PCA to 78.6% with the use of PCA as a feature selection technique. Similar reduction in the recognition rate was observed for the lower face AUs.

From the experimental results it is observed that PCA as a feature selector failed to increase the generalization performance of the model. With the use of all PCA components the models performance was no better than the use of all the

	Upper face AUs RR (%) : FAR (%)	Lower face AUs RR (%) : FAR (%)
Without PCA	83.51 : 6.68	81.98 : 8.53
With PCA	78.60 : 11.72	73.62 : 10.20

**Table 8.11:** Performance with and without using PCA

40 Gabor coefficients. Even the removal of first few components which contain information about the orientation and lighting conditions failed to increase the performance. We attribute this to the incompatibility between PCA and RNNs for the particular task at hand. Similar incompatibility between PCA and SVMs for expression recognition was observed in (11).

## 8.5 RNN Network Optimization

Till now we have mainly focused on the feature selection and their optimization. In this section we focus on the optimization techniques for RNNs. Some of the most widely used optimization techniques in literature are early stopping and weight decay. Early stopping uses a validation set in addition to training and test sets. The learning algorithm stops when the performance on the validation set stops improving. The model is then used for test data set. However, this kind of performance evaluation is not a good measure of generalization error as it depends on the composition of the specific validation set used. Weight decay introduced by Werbos (116) decreases the weights while training them through back propagation. In the next section, we experiment with weight decay for optimizing our FACS AU classification model.

### 8.5.1 Weight Decay

Here we study the effect of weight decay on our baseline recognition model which uses no dimensionality reduction techniques mentioned in Section 8.5. Here we experiment with three different decay constants to obtain an optimal value for optimal performance. The average recognition rate and false alarm rate for the six upper face using three different decay constants is given in Table 8.12. It is

## 8.5 RNN Network Optimization

concluded that a decay constant of 0.0001 works better in terms of recognition and false alarm rates for the six upper face AUs. The individual recognition and false alarm rate for the six upper face AUs with a decay constant of 0.0001 are given in Table 8.13.

	Recognition Rate (%)	False Alarm Rate (%)
Decay constant=0.0001	85.84	6.41
Decay constant=0.001	78.23	14.48
Decay Constant=0.01	No convergence	-

**Table 8.12:** Average recognition and false alarm rates for the six upper face AUs using three different decay constants

AUs	Recognition Rate(%)	False Alarm Rate(%)
AU1	89.57	3.495
AU2	94.72	3.95
AU4	81.18	10.66
AU5	90.1	2.17
AU6	80.59	6.65
AU7	78.85	11.53
<b>Average</b>	<b>85.84</b>	<b>6.41</b>

**Table 8.13:** Recognition and false alarm rate for the six upper face AUs with a decay constant of 0.0001

In Table 8.14 the average recognition rates for the five lower face AUs are given with three different delay constants. The results indicate that a decay constant 0.0001 also works better for the lower face AUs. The individual recognition and false alarm rate for individual lower face AUs are given in Table 8.15.

### 8.5.2 Discussion

Generally RNNs are very difficult to train but once trained they show good generalization capability. During training we encounter the problems like over training of the data sets. This leads to good generalization for the training sets but gives

## 8.6 Recognition of other FACS AUs

	Recognition Rate (%)	False Alarm Rate (%)
Decay constant=0.0001	84.91	6.07
Decay constant=0.001	81.20	9.96
Decay Constant=0.01	No convergence	-

**Table 8.14:** Average recognition and false alarm rates for the five lower face AUs for three different decay constants

AUs	Recognition Rate(%)	False Alarm Rate(%)
AU15	78.13	8.89
AU17	74.64	14.23
AU20	87.9 0	0.00
AU25	92.8	4.2
AU27	91.1	3.04
<b>Average</b>	<b>84.91</b>	<b>6.07</b>

**Table 8.15:** Recognition and false alarm rate for the five lower face AUs with a decay constant of 0.0001

bad performance on the test set. The other common problem is with the variance in the weights of the network. Few network weights tend to increase enormously with respect to others and this leads to inappropriate classification. In order to reduce the effects of over training and variance in weights, we have implemented weight decay. In our experiments weight decay in general improved the overall performance of our FACS AU recognition model.

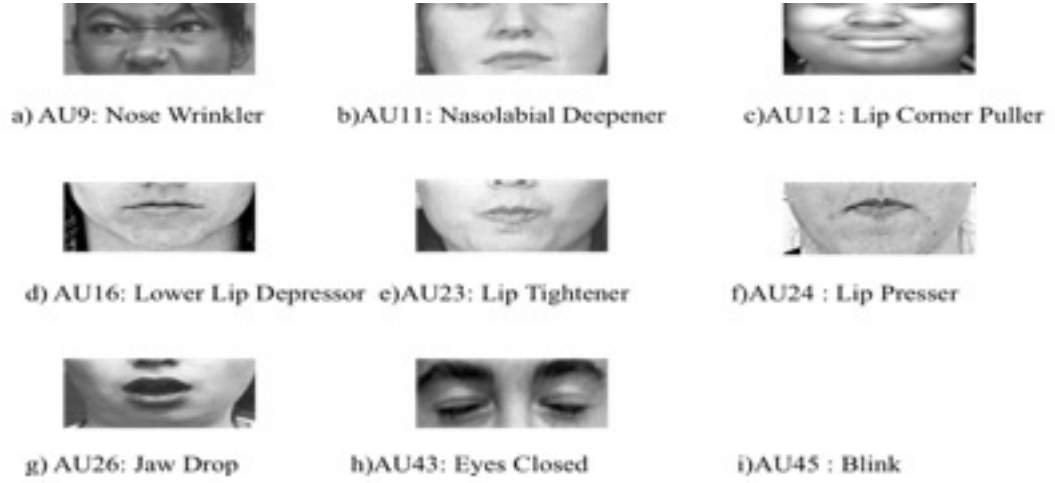
## 8.6 Recognition of other FACS AUs

We have so far restricted ourselves to a set of 11 FACS AUs for classification. The selection of these specific AUs was based on their popularity and the availability of a minimum number of samples which will enable the recognition model to learn the rules present in the input data. However, to completely evaluate our approach we classify some more AUs that are depicted in the Cohn-Kanade database. For some AUs in this set the total number of samples available for training is very

low ( $=20$ ). Insufficient number of samples may have a detrimental effect on the models performance.

### 8.6.1 AUs and Their Description

In this section, we discuss some of the other FACS AUs that were classified using our FACS AU recognition model. The AUs and their description are given in Fig. 8.11.



**Figure 8.11:** Other AUs in the FACS system and their description (Note: Blink cannot be depicted using a single frame.)

### 8.6.2 Experimentation and Results

In this section, we use the same baseline recognition model with weight decay described in Section 8.5. Almost all the AUs that are classified in this section belong to lower face region. Owing to this, we avoided the use of any of the feature selection techniques mentioned in Section 8.6 as none of them were able to improve the performance when used for the classification of lower face AUs. The overall configuration remains the same. The individual recognition and false alarm rates are given for the set of FACS AUs shown in Fig. 8.11 are given in Table 8.16.

---

## 8.7 Single Static Images vs Image Sequences

AUs	Recognition Rate(%)	False Alarm Rate(%)
AU9	62.94	7.45
AU11	58.23	9.20
AU12	69.00	15.20
AU16	64.69	0.00
AU23	77.16	8.52
AU24	75.00	9.54
AU26	62.88	8.51
AU43	80.28	6.21
AU45	82.89	7.48

**Table 8.16:** Classification performance for other FACS AUs

### 8.6.3 Discussion

In this section, we classified some of the other FACS AUs that were depicted in the Cohn-Kanade database with the help of more than 20 samples per AU from different subjects. The eye AUs 43 and 45 present in the upper face region gave the best performance, followed by the mouth AUs and the AUs surrounding the nose region. The poor performance of nose related AUs such as AU 9 and AU 11 can be attributed to the information lost when cropping the entire face region into upper and lower face. During this segmentation a part of information related to nose is lost when considering the lower face region for AU classification. One solution would be to crop the face such that we retain the entire nose and the upper lip region which is effected by these AUs.

The experiments carried out in this section have validated the use of RNNs for FACS AU classification model. It also added emphasis on the availability of sufficient number of samples for training and testing the classification model.

## 8.7 Single Static Images vs Image Sequences

The focus of this entire chapter was on the use of sequence of images with time variant component attached to them. Our main thrust was to justify the use of image sequences and the use of RNNs as a classifier for FACS AU recognition. In

this section we shed some light on the advantages and disadvantages of the type of the data we have used.

### 8.7.1 Advantages of using Image Sequences

- A sequence of images provide more information towards learning the AU depicted
- For real world applications it becomes important to have an understanding of an AU and its formation rather than the end product
- Unlike in static images where only an AU at its peak intensity is provided, a sequence of images will help the model to learn AUs even at low intensities
- Use of images sequences in conjunction with classifiers such as RNNs provide better understanding of the AUs

### 8.7.2 Disadvantages of using Image Sequences

- One major disadvantage in using images sequences is the dimensionality of data for each sample
- Putting together an image sequence that can exactly depict the formation of an AU is much complicated than selecting a single static image

## 8.8 Conclusions

The experiments in this chapter conclude that RNNs can be used for FACS AU recognition and that the recognition model is comparable and in cases better than the use of SVMs using single static images. The baseline recognition for the six upper face and five lower face AUs obtained using RNNs is 83.51% and 81.98% respectively. The study of different set of scales on the generalization performance concluded that the set of higher three frequencies perform better than the other sets. However, when compared to the use of all the frequencies there was an overall reduction in the performance. Local Gabor filters which include all the

frequency scales and yet reduce the dimensionality of the recognition model by over 80% were studied. The best performance was given by LG3 (5x8) variant of local Gabor filters which increased the performance by over 1% for the upper face AUs and a similar performance was obtained for the lower face AUs when compared to using all the Gabor coefficients. The use of PCA, however, failed to increase the generalization performance. Weight decay, an optimization technique helped in improving the overall performance. With a decay constant of 0.0001, a performance of 85.84% for the six upper face AUs and 84.91% for the five lower face AUs was obtained. Some of the other FACS AUs were also classified to generalize our recognition model.

## 8.9 Summary

In this chapter we discussed the use of RNNs as a classifier for FACS AU recognition. Using Gabor filters to extract the features from a sequence of frames, we trained the network with the resulting Gabor coefficients. Our results were comparable and even better to that of using SVMs with single static images. In order to reduce the dimensionality of the data which explodes by 40 times with the application of Gabor filters with five frequencies and eight orientations, we used a well-known technique of selecting a few frequency scales such as frequency scale selection, local Gabor filters and PCA. Local Gabor filters gave some promising results but was depend on the face region. We then studied the use of weight decay for optimizing the recognition model. Use of weight decay improved the overall performance of the recognition model. Towards generalizing our FACS AU recognition model classified a set of other FACS AUs depicted in the database.



## Chapter 9

# FACS AU Recognition Using Multiple Output RNNs

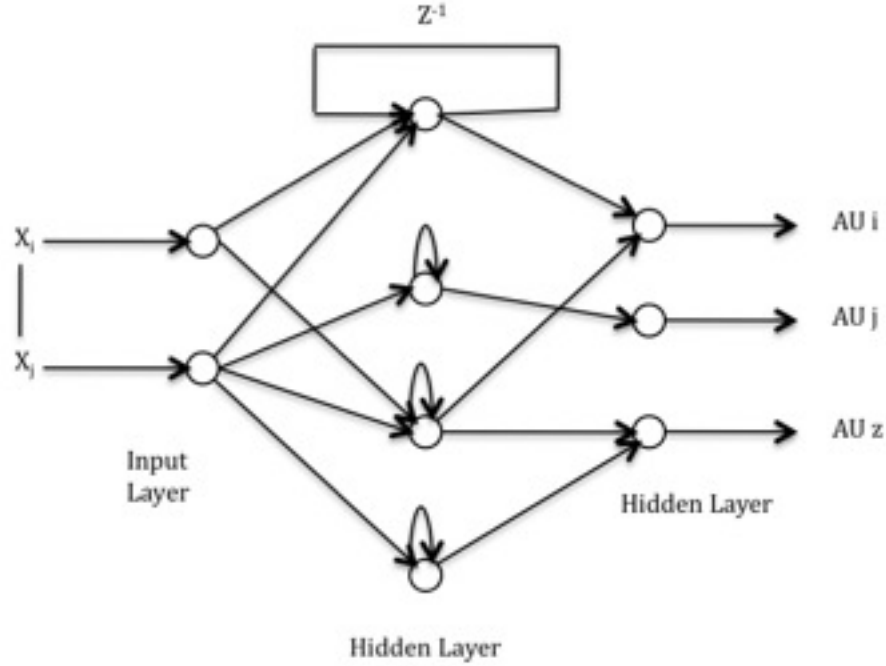
The use of single output RNN for recognition of one single FACS AU was successful. The criterion was to classify a specific AU irrespective of the presence of other AUs in the sequence of frames. In real life situations, however, it would be advantageous to classify a set of AUs by a single RNN. This is motivated from the fact that most of the FACS AUs are mimicked in groups in real life situations which presents a scenario of high AU correlations. High correlations between AUs will be an added advantage when classifying their presence together. In this chapter, we experimented with the classification process where more than one AU is classified by a multiple output RNN. Each output unit will be trained to detect the presence/absence of one AU.

### 9.1 Multiple Output RNNs

Multiple output units will facilitate the recognition of a set of FACS AUs by a single network. Unlike the use of separate networks in our previous chapters, use of a single network with multiple output units will be able to gain the knowledge about the inter AU correlations that may exist in the data towards their classification. The presence of one AU may be an indication of the presence of some other AU. This knowledge is absent when we use a single network for classifica-

tion of each AU. Fig. 9.1 shows the structure of a simple RNN with multiple output units.

We used one RNN each for one combination of AUs from upper face and lower face regions. The number of output units depend on the number of AUs present in the combination. Before we review the use of a multi output RNN it is very important to know the AU combinations that exists the most in the database, their additive/non-additive nature and the degree of AU correlations that exists in the database. This information will provide a better understanding of the performance given by a multi output RNN.



**Figure 9.1:** Structure of a Multiple Output RNN

## 9.2 AU Combinations

In this section, we present the different problematic AU combinations given in the FACS system covering the ones we are interested in. The knowledge of these combinations will enable us in understanding the difficulties faced by a classification

Combination of AUs in the upper face	Combination of AUs in the lower face
1+2	15+17
1+4	20+27
1+2+4	20+25
1+4+6	15+17
1+2+4+6	
4+6	
1+4+5	
1+4+5+7	
2+4+5	
2+4+5+7	
4+5	
4+5+7	
1+4+7	
4+7	
5+7	
6+7	

**Table 9.1:** Problematic AU Combinations present in Cohn-Kanade database

model during their classification. There also exist combinations which makes it difficult to predict the presence of one or more subtle AUs that are influenced by more dominant ones. The different AU combinations given in the FACS manual and that occur in the Cohn-Kanade database are given in Table 9.1.

### 9.2.1 Additive and Non-Additive AUs

The way in which AUs affect one another's appearance will have a drastic effect on their classification. The condition which can result in this situation is the presence of non-additive AUs. Additive AUs do not have an impact on the appearance change of a set of AUs as the appearance of each AU in the combination is identical to that of its appearance when it occurs alone. In additive AU combinations the evidence of each AU remains clearly recognizable because they have combined

without distorting or changing the appearance of each separate AU. In case of non-additive AUs, the combined appearance change by a combination of AUs is in some way different from the separate appearances caused by the individual AUs. Here, the appearance change is not simply the sum of the appearance changes due to different AUs but a distinctively new appearance change. However, when classifying AUs as a set this change may be beneficial towards successful validation of the presence of other AUs. In our view, a multiple output RNN will be able to gain information about the non-additive characteristics. The same is not possible with a single output RNN as the additional information regarding the presence of other AUs in the combination is not provided.

### 9.2.2 Inter AU Correlation in Cohn-Kanade Database

The study of inter AU correlation becomes important as the database under consideration (Cohn-Kanade database) contains high degree of AU correlations. High AU correlation makes it possible to predict an AU in one face region based on the presence of another AU elsewhere in the face. This is an advantage when the entire face is considered for the classification purposes (118). Even though the face was segmented into two regions there is still scope for taking advantage of correlation information that exists between AUs from the same face region. The advantage of AU correlation is based on the fact that; Suppose  $AU_i$  were more difficult to classify than  $AU_j$  and the classifier has known that  $AU_i$  was perfectly correlated with  $AU_j$ , then the classifier when trying to classify  $AU_i$  will instead classify  $AU_j$  and then output the same result for  $AU_i$ .

We hypothesized that a multi output RNN will be able to gain this knowledge when training the different samples from the database and applying the same while testing. The matrix of AU-correlations second our hypothesis which is given over the data set we used for our experiments. We dealt with upper and lower face AUs separately and so will provided the inter AU correlations for upper and lower face AUs in two separate tables. Tables 9.2 and Table 9.3 provide the same. Here we considered the correlation between  $AU_i$  and  $AU_j$  to be high if  $|\rho_{ij}| \geq 0.50$ . The corresponding entries are shown in bold to depict this high correlation coefficient.

	AU1	AU2	AU4	AU5	AU6	AU7
AU1	<b>1.00</b>					
AU2	<b>0.62</b>	<b>1.00</b>				
AU4	0.24	-0.20	<b>1.00</b>			
AU5	<b>0.54</b>	<b>0.75</b>	-0.15	<b>1.00</b>		
AU6	-0.10	-0.17	0.43	-0.11	<b>1.00</b>	
AU7	-0.09	-0.27	<b>0.68</b>	-0.18	<b>0.52</b>	<b>1.00</b>

**Table 9.2:** AU correlation in our data subset for the upper face AUs

	AU15	AU17	AU20	AU25	AU27
AU15	<b>1.00</b>				
AU17	<b>0.53</b>	<b>1.00</b>			
AU20	-0.12	-0.15	<b>1.00</b>		
AU25	-0.16	-0.27	<b>0.52</b>	<b>1.00</b>	
AU27	-0.11	-0.25	-0.16	<b>0.63</b>	<b>1.00</b>

**Table 9.3:** AU correlation in our data subset for the lower face AUs

### 9.2.3 AU Combinations

This section is dedicated to answer a very crucial question of whether we can use one RNN for the classification of all the upper/lower face AUs with the type of data we have on hand. If not, what AUs can we classify using one single network and prove our hypothesis regarding the effect of AU correlation on the recognition models performance.

The original frame sequence that was formed by collecting a set of frames depicts an AU of interest from its absence to its presence. These sequences will have other AUs acting that may not have been depicted in a correct manner. For instance, while  $AU_i$  is getting depicted with the help of five frames it may so happen that  $AU_j$  will start from frame 1, starts to increase in intensity till frame 3 and stays at that intensity till the last frame. This depiction of  $AU_j$  might not be enough for its correct classification. This scenario may work if the classification model is also able to specify the intensity of the AU under consideration. The other possible case would be where the presence of  $AU_j$  is only evident from the

3<sup>rd</sup> or 4<sup>th</sup> frame and never reaches its apex by the last frame. In these situations trying to classify  $AU_j$  will result in poor performance and in some cases non-convergence.

Upper face AU combinations	Lower face AU combinations
1+2	15+17
1+2+5	20+25
4+7	25+27
6+7	

**Table 9.4:** AU combinations considered for classification

Taking this into account, this work would only consider classifying a combination of AUs rather than all the AUs defined in the upper or lower half of the face. Each group will consist of AUs that can be successfully depicted using the same sequence of images. The AU sets that are classified are given in Table 9.4.

## 9.3 Data Collection

The data collection step varies to some degree when formulating an image sequence depicting a combination of AUs. Unlike the frames which depict only one AU, here the set of frames should depict all the AUs in the combination from their absence to their presence. For this, it would be ideal to have a database where the combination of AUs is depicted simultaneously. This would make it easy for a set of frames to depict all the AUs present in the combination. With the data available in Cohn-Kanade database it becomes difficult to depict all the AUs. Keeping this in mind we have only classified a selected group given in Table 9.4. These include the combinations where each individual AU can be depicted along with the others.

## 9.4 Experiments and Results

The preprocessing and the feature extraction steps for these experiments are the same as in Chapter 8. However, unlike the use of a single RNN for the classification of each FACS AU, here we used one multiple output RNN for the classification of a set of AUs present in either the upper half or lower half of the face region. The system was optimized using weight decay. A value of 0.5 or greater is regarded as AU being present. The average recognition rate is the one attained when classifying the AU using all the combinations given in the Table 9.4. For example in case of AU1, the recognition rate is the average obtained by using the classification performance of AU in combinations 1+2 and 1+2+5. The individual recognition and false alarm rates for all the AUs classified in the upper face are given in Table 9.5 and that of lower face are given in Table 9.6.

FACS AUs	Recognition Rate (%)	False Alarm Rate (%)
AU1	90.02	3.21
AU2	95.23	3.92
AU4	82.30	9.52
AU5	91.04	2.25
AU6	80.98	6.12
AU7	79.62	9.78
<b>Average</b>	<b>86.56</b>	<b>5.80</b>

**Table 9.5:** Recognition and False alarm rate for six upper face AUs

FACS AUs	Recognition Rate (%)	False Alarm Rate (%)
AU15	79.42	7.83
AU17	75.10	13.20
AU20	88.54	1.25
AU25	93.31	4.01
AU27	91.90	2.83
<b>Average</b>	<b>85.65</b>	<b>5.83</b>

**Table 9.6:** Recognition and False alarm rate for five lower face AUs

## 9.5 Discussions

In this chapter, we have studied the use of a multiple output RNN and its ability to handle the inter AU relations in terms of different AU combinations that exists, the non-additive nature of AU combinations and the correlation between the different AU in the Cohn-Kanade database. There was nearly 1% increase in the recognition rate when AUs were trained in combinations when compared to the use of individual RNNs. However, the percentage increase is not significant as expected. This can be attribute to the difficulty in depicting all the AUs in the combination consistently in terms of their intensity as different AUs started at different times in the videos. This has affected some samples where one AU was more pronounced than the other in the 2nd or 3rd frame itself and remained more or less at the same level of intensity till the last frame. The second reason was the unavailability of enough samples for training. One solution for improving the performance will be the availability of a huge database where AUs occur simultaneously starting at the same time which in turn will also help in getting more number of samples for classification. Samples where AUs occur singly should also help in providing a better comparative analysis in terms of the use of a single output RNN vs a multiple output RNN.

## 9.6 Summary

This chapter focuses on our hypothesis which suggests that a multiple output RNN will be able to gain important information from the AU correlations that occur in the database and from the presence of the different AU combinations. This information should help a multiple output RNN perform better. To some extend the hypothesis was verified, but needs more data to provide significant results. One solution would be the availability of a database where AUs in a combination occur simultaneously. Also the availability of a database where AUs occur singly will help in a better comparative analysis between a single output vs multiple output RNNs.



## Chapter 10

# Recognition with Difference Images

So far, we have been looking at the use of single static images and image sequences compiled using the original frames from the video clips. Each such frame carries huge amount of information. A simple  $64 \times 64$  pixels image has a vector of 4096 data points to be considered. High dimensional data is a problem while designing classification models where the classifiers performance is affected by the dimensionality. Classifiers such as SVMs do not suffer from such a disadvantage but classifiers such as RNNs are affected by huge amounts of data. One of the steps to reduce the dimensionality of the data is to incorporate feature selection techniques. However, most of the feature selection techniques such the use of PCA, LDA, and Gabor filters etc require extra computations. There are feature selection techniques such as frequency scale selection and local Gabor filters that do not require any extra computations. However, the results in section 8.6 indicated that these methods had little or no affect on the performance of RNN based FACS AU classification model. This was attributed to the data lose during the selection process.

One of the best ways to reduce the dimensionality of the data is to remove pixels that have not been affected by the muscle actions. This can be achieved by implementing difference images. The idea is to find the difference between two consecutive frames in a image sequence. This removes all the pixels that were

not displaced by muscle actions. The advantages of using difference images are discussed in the next section.

## 10.1 Normal vs. Difference Images

### 10.1.1 Feature Reduction

The normal feature reduction achieved by standard feature selection techniques is clearly evident by the reduction in the number of features in the feature vector. However, this is not the case with the use of difference images. The size of the difference image remains the same ( $m * n$ ) when two images of size ( $n * n$ ) are subtracted;  $m$  and  $n$  being the width and height of the image. This leaves the length of the feature vector intact. The resultant difference image also contains a large number of pixels with a value equal to zero depending upon the degree of similarity between the frames used for calculating the same. When a RNN is trained using these difference images with a huge number of pixels with a value equal to zero most of the network inputs are close to zero. These low values makes the input to hidden weight calculations equal to a value zero making their effect nearly insignificant towards the networks output. This acts as a feature reduction as inputs with a value greater than zero contribute towards the weight calculations which in turn affects the networks output.

### 10.1.2 Noise Reduction

Noise in our application includes all the information that is irrelevant and redundant. Noise also includes information which deviates the learning process from accurately analyzing the input data. The input data consists of entire face region or a part of the face. Usually, this data contains more information than required. For example, classifying AU5 which is related to muscle action around eyelids that results in a change of the eye aperture, one hardly needs information regarding the changes that take place in the forehead, eyebrows, cheeks and root of the nose. However, when considering the image sequences as mentioned in Chapter

8, one will have all this extra information present in every frame. Feature selection techniques in many occasions fail to completely remove such irrelevant and redundant data.

One successful way of removing such redundant data is through the use of difference images. Difference images when calculated using two consecutive frames, removes all the pixels that were not displaced due to the muscles actions; retaining only those clusters that have been displaced. This process removes all the information not related to a particular muscle action thus avoiding the unnecessary and redundant information.

## 10.2 Data Preprocessing

In this section, the same preprocessing steps given in Section 8.4.2 are performed. Fig. 10.1(a) presents a sequence of cropped images depicting a lower face AU. Until now, this was the standard data for all the experiments based on the use of RNNs. In this sequence there is the presence of redundant data from frame to frame such as parts of the face that are unaffected by the muscle actions responsible for the AU. One way of successfully reducing this kind of redundant data is by calculating the difference images. Difference images show the cluster of those pixels that have been displaced due to the muscle action. Considering  $f_{-}(n)$  represents frame  $n$  and  $f_{-}(n + 1)$  represents frame  $(n + 1)$  in the image sequence we compute  $f_{-}(n + 1) - f_{-}(n)$ , as our difference image  $df_{-}(n)$ . The difference images calculated using our original image sequence on Fig. 10.1(a) is shown in Fig. 10.1(b). The reason for the use of difference images rather than a feature selection technique for the removal of redundant data is the loss of important feature information by these selection techniques.

## 10.3 Feature Extraction

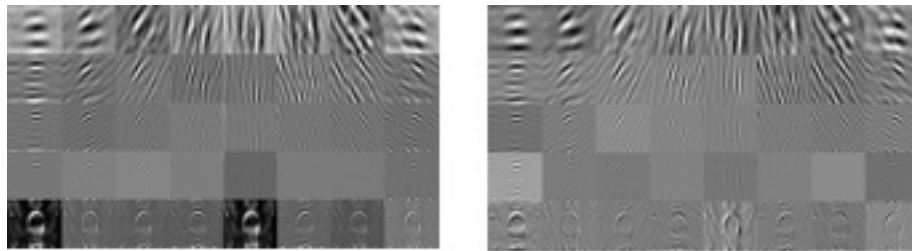
A 5\*8 Gabor jet is used to extract features from the difference images. So each pixel in the difference image will have a set of 40 Gabor coefficients. The Gabor responses obtained by applying a 5\*8 Gabor jet on the last difference image of the sequence depicted in Fig. 10.1(b) is given in Fig. 10.2.



(a) Normal image sequence consisting of 5 frames



(b) Difference image sequence

**Figure 10.1:** Original and the difference image sequences**Figure 10.2:** Filtered 5\*8 coefficients of the last difference image

## 10.4 Classification

A RNN similar to one shown in Fig. 8.4 is used for classification of each FACS AU. All the parameters remain the same except the use of four difference images compared to five normal images in a sequence. The number of hidden units is the same owing to very slight change in the number of input units. A three-fold cross validation is used to obtain the average recognition rate for each AU.

## 10.5 Results and Discussion

An average recognition rate of 88.62% with a false alarm rate of 2.37% is achieved for the six upper face AUs. The individual recognition and false alarm rates are

given in Table 10.1. In case of lower face AUs the recognition rate obtained was 87.28% with a false alarm rate of 4.53%. Individual results are given in Table 10.2.

AUs	Recognition Rate(%)	False Alarm Rate(%)
AU1	87.82	1.91
AU2	92.08	1.92
AU4	90.93	1.81
AU5	90.28	3.03
AU6	90.18	1.5
AU7	80.45	4.06
<b>Average</b>	<b>88.62</b>	<b>2.37</b>

**Table 10.1:** Recognition and False alarm rate for six upper face AUs using difference images and RNN

AUs	Recognition Rate(%)	False Alarm Rate(%)
AU15	82.87	5.56
AU17	78.85	5.13
AU20	86.86	6.14
AU25	94.32	2.54
AU27	93.48	3.26
<b>Average</b>	<b>87.28</b>	<b>4.53</b>

**Table 10.2:** Recognition and False alarm rate for five lower face AUs using difference images and RNN

The original approach of using RNN with normal images in a sequence gave a recognition rate of 85.38% with a false alarm rate of 6.24% upon network optimization. With the use of difference images the performance was improved to a recognition rate of 87.95% and a false alarm rate of 3.45%. The improvement seen in the recognition and false alarm rate is significantly evident taking into account the absence of network optimization with the use of difference images. One of the reasons for this improvement in the performance can be attributed

to the removal of redundant data like parts of the face that are not affected by an AU under consideration. It is also important to note that certain AUs like AU 4,5 and 6 in the upper face were classified much better when compared to the use of normal images. These AUs and the skin deformations due to their muscle actions are limited to a small area on the face (43). This leaves the rest of the face unaffected thus removing a lot of redundant data. The other reason for improved performance would be the presence of concentrated clusters providing more concrete information rather than scattered cluster of pixels all over the image frame. Similar reasoning can be given for AUs 15 and 17 in the lower face region.

## 10.6 Network Optimization

### 10.6.1 Weight Decay

Towards optimizing the network we perform weight decay to the above model based on difference images. The experiments were run for different decay constants (0.01, 0.001 and 0.0001). Surprisingly, there was no improvement in the performance. In fact, the application of weight decay deteriorated the performance as the training progressed. One reason for this would be the presence of significant amounts of NULL data in the difference images. This leads to a system whose weights are relatively very small. On the other hand, weight decay in theory penalizes large weights which in fact reduces the smaller network weights (with a highest value in the network) further degrading the performance. One alternative to this problem is to eliminate all the negligible weights and in turn the nodes that contribute the least towards a networks output. This can be done using weight elimination.

### 10.6.2 Weight Elimination

This approach involves training the recognition model first with 15 hidden nodes (optimal network). Once the training is successful the state neuron with the smallest weight vector is removed reducing the number of hidden nodes to 14

## 10.7 Comparison: Normal vs Difference Images

---

and the network is retrained. The above process is repeated until the network after the recent pruning failed to converge or when the networks reaches the maximum number of epochs. In case the pruned network fails to converge, we use the previous network state as the solution network. The AUs along with their recognition rates along with pruned hidden units (successful convergence indicated by a '\*') is given in Table .10.3.

FACS AUs	Recognition Rate (%)	Hidden units 14 13 12
AU1	88.01	* - -
AU2	91.02	* - -
AU4	90.93	- - -
AU5	90.28	- - -
AU6	90.18	- - -
AU7	80.45	- - -
AU15	81.23	* - -
AU17	78.85	- - -
AU20	86.94	* - -
AU25	92.10	* - -
AU27	93.48	- - -

**Table 10.3:** FACS AU recognition with weight elimination. '\*' indicates that the system converged with the specified RR% and '-' indicates that once pruned to specific AUs the system never converged

It is observed from the results that except in case of couple of AUs the recognition model fails to perform better after the first pruning itself. This presents a scenario where the initial model was optimal in itself and no further optimization is possible.

## 10.7 Comparison: Normal vs Difference Images

From the results, it is clear that the system based on the use of difference images performed better than the one based on the use of normal images. The percentage

## 10.7 Comparison: Normal vs Difference Images

---

increase in the recognition rate was over 3% when compared to the RNN based model using normal images. The reduction in the false alarm rate was statistically significant when compared to baseline results. This was attributed to reduction in non-zero pixels in the feature vector and also the resultant decrease in noise. Change in the performance was much pronounced in case of AUs like 4, 5, 6, 15 and 17. This was attributed to the affect of their respective muscle actions which was limited to a small area on the face region (43). Small cluster of pixels that form a part of difference images depicting these AUs can also be used to second the pronounced improvement observed in the performance.



## Chapter 11

# Conclusions and Directions for Future Work

The research proposed that the use of RNNs would provide a better performance than other classic classifiers, such as traditional neural networks and SVMs, when used with time variant sequences. This was to be applied to recognition of AUs from the FACS database. To our knowledge, no work had been carried out on classification of FACS AUs using RNNs. The germane issues to be investigated were: (i) whether temporal modeling of FACS AUs from video streams could improve the recognition performance compared to the recognition from static images; (ii) whether RNNs were better modeling tools compared to SVMs; (iii) how the cross-correlation between AUs impact recognition performance and whether this cross-correlation could be exploited for FACS AUs recognition; and (iv) whether the use of difference images was a viable alternative to other, more elaborate feature extraction and reduction techniques.

In order to address the first question, the investigation examined the motivation for the use of RNNs. Advantages of using RNNs lie with their ability to learn from previous time steps, which plays a crucial role in learning temporal variant data. The input data in this case, is a sequence of images depicting a typical AU. For feature extraction from image sequences, Gabor filters were applied to face images. The resultant Gabor coefficients were then used to train the RNN. Individual RNNs were used for the recognition of each of 11 predefined FACS AUs. This data set was contains both upper and lower face AUs. As was reported in

---

previous chapters, experimental results obtained proved the ability of RNNs to model time variant FACS AUs from video sequences.

Consequently, the next study was to investigate whether the modeling performance achieved by RNNs was indeed better than could be achieved using classical means, such as SVMs. To provide this comparison, RNNs were applied to learn from image sequences, while SVMs learn from single static images. The use of Gabor filters as feature extractors were studied, with individual SVM classifier being used for each of the 11 FACS AUs. As was reported in previous chapters, the increase in the performance using RNNs was more than 2% when compared to the use of SVMs with a complimentary drop in the false alarm rate. This indicates that the use of image sequences does provide a better recognition of an AU by the classification model. However, the observed improvement may not be statistically significant, which however, could be attributed to the low number of samples (50 - 100) available to the classification model.

In a following investigation, focus was shifted towards a common concern with the use of Gabor filters, namely, the large dimensionality of coefficients generated. Towards reducing this dimensionality frequency scale selection and local Gabor filters were applied. In frequency scale selection, it was shown that higher frequencies performed better than middle and lower frequency sets. However, this was not better than the use of global Gabor coefficients. A reduction in the recognition rate indicated all the frequency scales to have important information. Hence, in order to preserve all the frequency scales, and yet reduce the dimensionality of the data, local Gabor filters were applied. Depending on the selection of the frequency scales three variants of local Gabor filters were suggested, which were named LG1, LG2 and LG3. As was reported in previous chapters, experimental results showed LG3 variants to perform better than LG1 and LG2 for both the upper and lower face regions. The use of LG3 variant also increased the performance by over 1% for the six upper face AUs. However for the five lower face AUs, LG3 variants performance was lowered by 1% when compared to the use all the Gabor coefficients. These results suggest that local Gabor filters reduces the dimensionality, but their effect on the performance varies from face region to face region.

---

Investigation was also conducted on the classical feature selection technique, PCA. Principle component analysis failed to improve the performance even with the use of histogram normalization. This could be attributed to an incompatibility between PCA and RNNs for this particular task. Towards the optimization of the RNNs itself, focus was on the use of network optimization techniques. Weight decay, which has been widely used for optimizing RNNs, was studied. The use of weight decay was shown to increase the overall performance by nearly 3%. Studies on generalization of RNN based FACS AU recognition model using other FACS AUs depicted in the database also showed the approach to work well for most of the FACS AUs.

Studies on cross-correlation between AUs and its impact on recognition performance were motivated by the presence of the high AU correlations that exists in the Cohn-Kanade database. It was suggested that such cross-correlation could be exploited to advantage in FACS AU recognition. Studies on AU correlations experimented with a single multiple output RNN which was trained and tested for the recognition of a given set of AUs. It was hypothesized that one single network would be able to learn better from the relative information that exists in the correlated data. Experimental results indicated that there was some improvement (of 1%) in the performance by the use of the multiple output RNN, when compared to the use of a single output RNN. The percentage increase, however, was not shown to be significant. The unavailability of a large enough data samples depicting combination of AUs with high cross-correlation values can be seen as the main reason for this.

The final focus of this research was the use of difference images and their viability as alternative to more elaborate feature extraction and reduction techniques. Calculation of difference images on their own will not result in the reduction of the feature vector, as the size of the difference image remains the same as that of the original image. However, the number of non-zero features is much less when compared to the original images. The presence of huge number of zero valued pixels acts as a noise reduction and feature reduction step. Studies on the performance of the RNN-based classifier using these difference images indicated that the overall performance increased by over 2.5% with a steep reduction in the false alarm rate (3.45%) when compared to the original RNN-based approach.

---

The reduction in the false alarm rate was, in particular, statistically significant for most of the FACS AUs that were classified using this approach. The use of network optimization techniques, such as weight decay and weight elimination was also studied. It was, thus, concluded that the use of difference images is a viable approach for feature reduction without added computational capability.

This research work has made some important contributions to the field of FACS AU recognition. Firstly, to our knowledge, the use of RNNs for FACS AU recognition has not been studied previously. The research thus has paved the way for other important questions in the field of FACS AU recognition and the use of RNNs for this particular task. This work included little out-of-plane head motions in the data. However, it would be important to know if RNNs can provide an added advantage over other classifiers with out-of-plane head motions, where an AU is occluded for a few frames and then reappears. It is suggested that the ability of RNNs to learn from previous time steps should be able to help recognize the AUs even when occluded. The use of Gabor filters over the entire face region or even part of a face increases the dimensionality of the model by several orders. On the other hand, RNNs are difficult to train when the dimensionality of the data is huge. Other classifiers such as SVMs do not face this problem. It therefore would be interesting to investigate whether the use of other feature techniques such as geometric features help towards improving the performance. Similar steps for improving the networks performance, such as the use of genetic algorithms can in this regard also be tested.

## Appendix A

### Fast Fourier Transforms

A Fast Fourier transform(FFT) is an efficient algorithm to defined compute the discrete Fourier transform(DFT) and its inverse. A DFT decomposes a sequence of values into components of different frequencies. An FFT is a way to compute the same result more quickly. Computing a DFT of  $N$  points takes  $O(N^2)$  arithmetical operations, while FFT can compute the same result in on  $O(N \log N)$  operations. Let  $x_0, x_1, \dots, x_{N-1}$  be the complex numbers. The DFT is defined by the formula

$$x_n e^{-i2\pi kn/Nkn} \quad (11.1)$$

where  $k = 0, 1, \dots, N - 1$ .

For 2D data like images, we need a 2D Fourier transform that is computed by transforming each column of the image as set of  $N$  complex numbers, and then transforming each row of the result as a set of  $N$  complex numbers. In other words, expressing the two-dimensional Fourier transform in terms of a series of  $2N$  one-dimensional transforms decreases the number of required computations. With an image with  $M$  rows and  $M$  columns, the 2D FFT takes  $O(M^2 \log M^2)$  operations. This is given as  $O(N \log N)$  where  $N = M * M$ . For the same  $M*M$  image, the DFT takes  $O(N^2)$  operations. This shows that there is significant improvement by the use of FFT over DFT, in particular for large images. The Fourier Transform produces a complex number valued output image which is displayed with two images, either with

the real and imaginary part or with magnitude and phase. In image processing, often only the magnitude of the Fourier transform is used, as it contains most of the information of the geometric structure of the spatial domain image. However, when the need arises to re-transform the Fourier image into its spatial domain after some processing in the frequency domain, one should preserve both the magnitude and phase of the Fourier image.

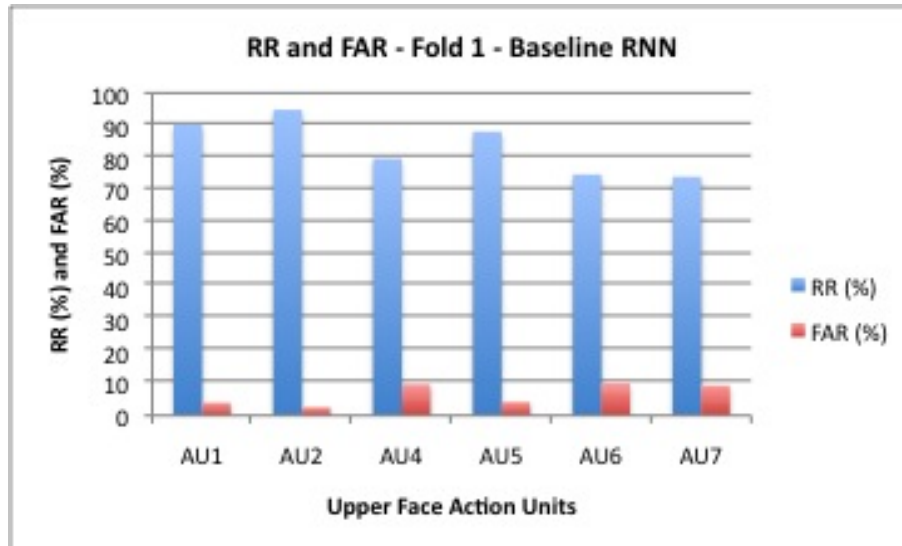
## **Appendix B**

### **Individual Results for Each AU**

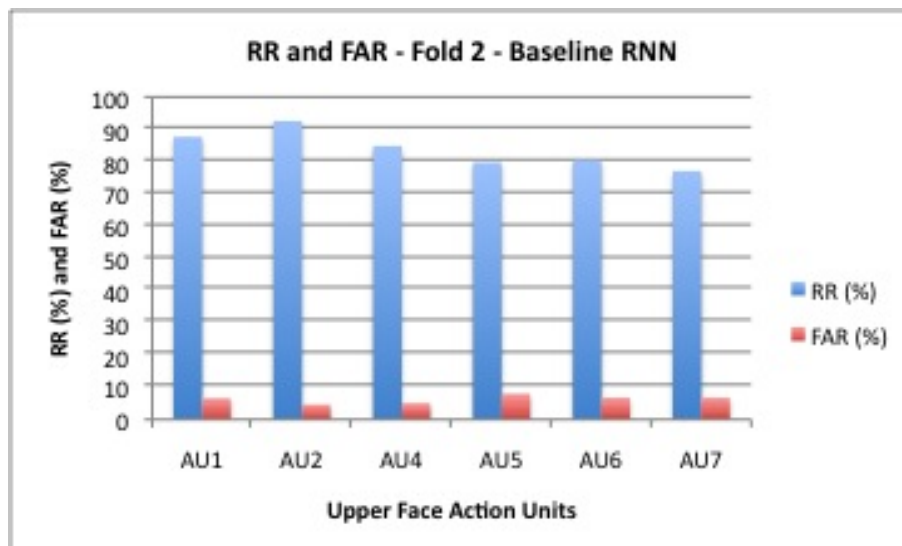
Appendix B presents the three fold cross training and testing graphs for each AU. Each graph includes the recognition rate (RR %) and false alarm rate (FAR %) for each fold obtained for all the AUs in a particular face region i.e. either upper or lower face region. In each fold, the individual classifier is trained at least for 5 times to reach at conclusion in terms of recognition and false alarm rates.

#### **Baseline Recognition Using RNNs**

Figures from 11.1 to 11.6 show the results obtained using a RNN classifier towards recognizing the upper and lower face AUs. The results depict the baseline recognition and false alarm rate, with no feature reduction, feature selection or network optimization techniques.

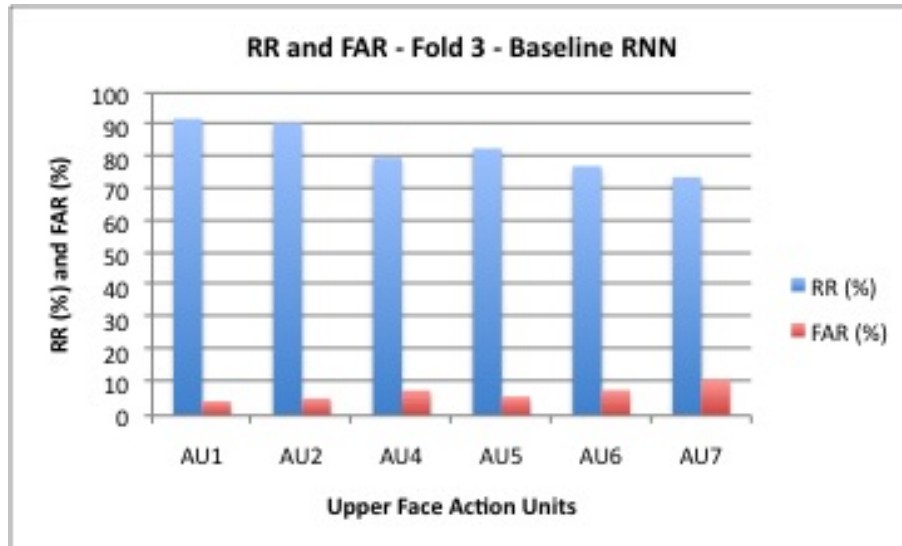


**Figure 11.1:** Baseline recognition (RR %) and false alarm rate (FAR %) for all the upper face AUs obtained during fold 1 using RNN classifier

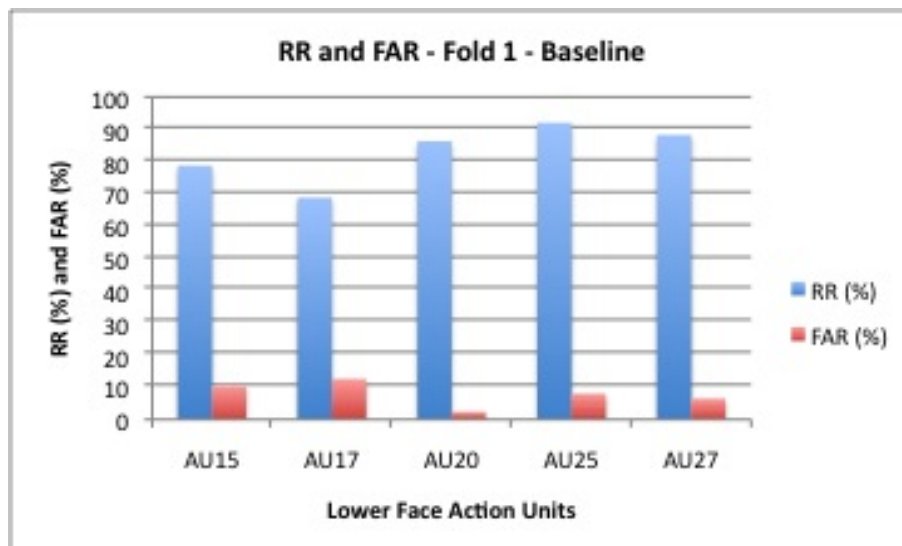


**Figure 11.2:** Baseline recognition (RR %) and false alarm rate (FAR %) for all the upper face AUs obtained during fold 2 using RNN classifier

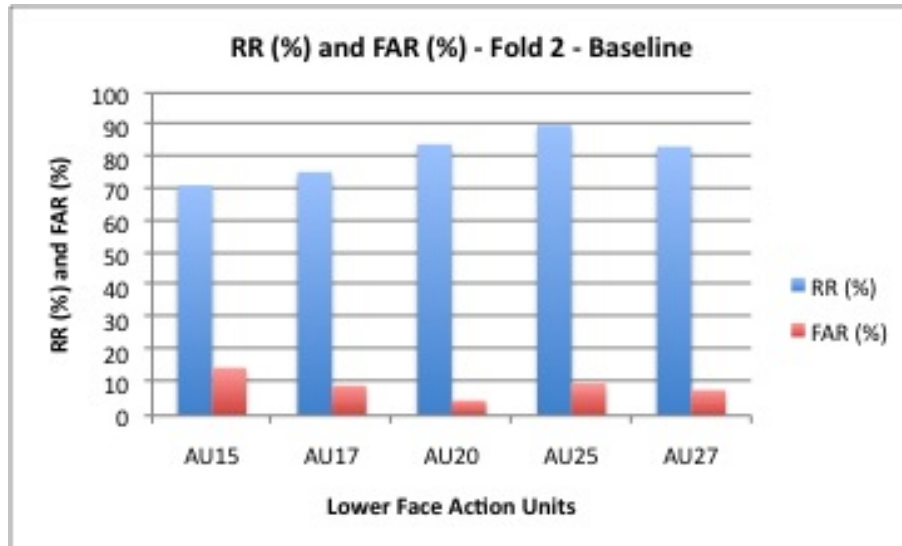




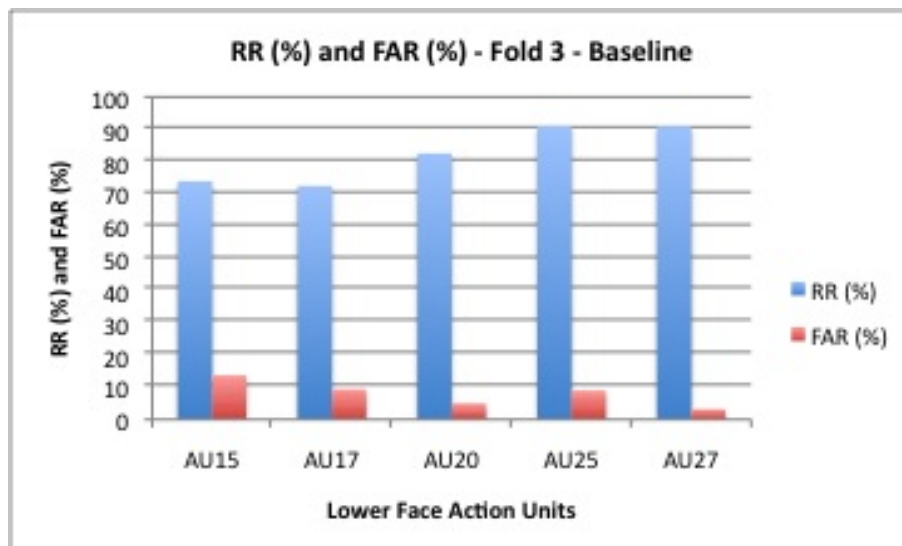
**Figure 11.3:** Baseline recognition (RR %) and false alarm rate (FAR %) for all the upper face AUs obtained during fold 3 using RNN classifier



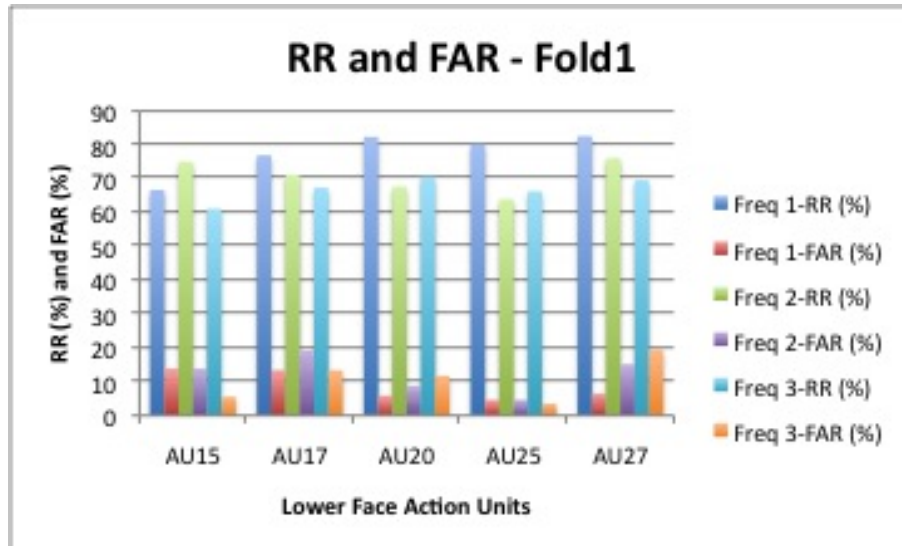
**Figure 11.4:** Baseline recognition (RR %) and false alarm rate (FAR %) for all the lower face AUs obtained during fold 1 using RNN classifier



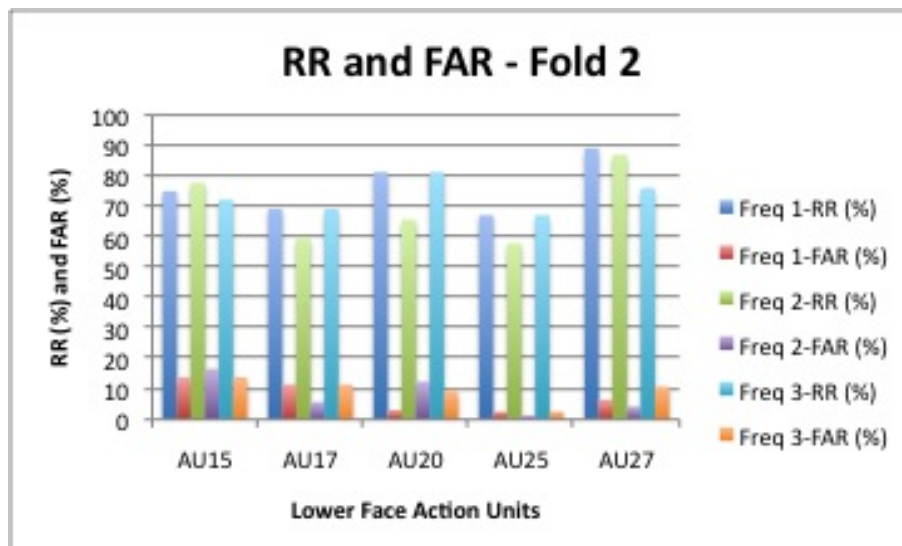
**Figure 11.5:** Baseline recognition (RR %) and false alarm rate (FAR %) for all the lower face AUs obtained during fold 2 using RNN classifier



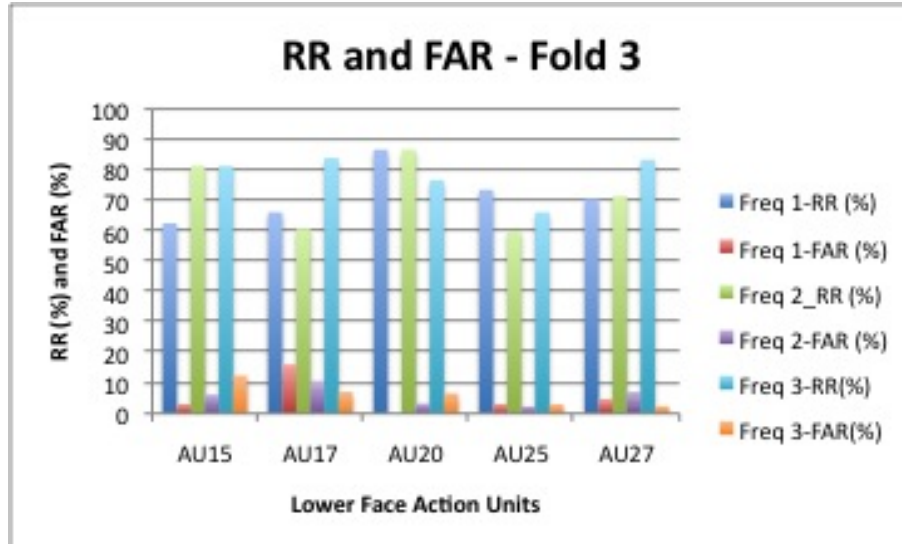
**Figure 11.6:** Baseline recognition (RR %) and false alarm rate (FAR %) for all the lower face AUs obtained during fold 3 using RNN classifier



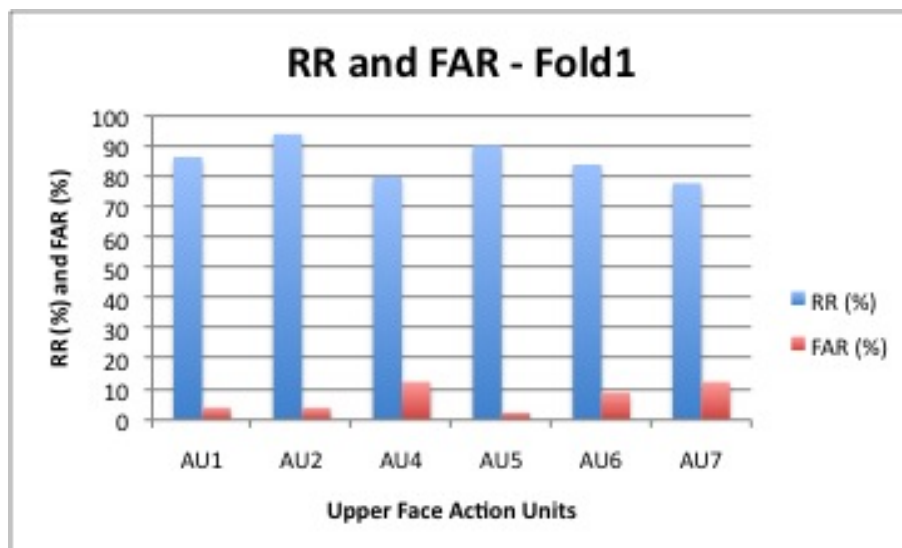
**Figure 11.7:** Recognition (RR %) and false alarm rate (FAR %) for all the upper face AUs obtained during fold 1 using frequency scale selection and RNN classifier



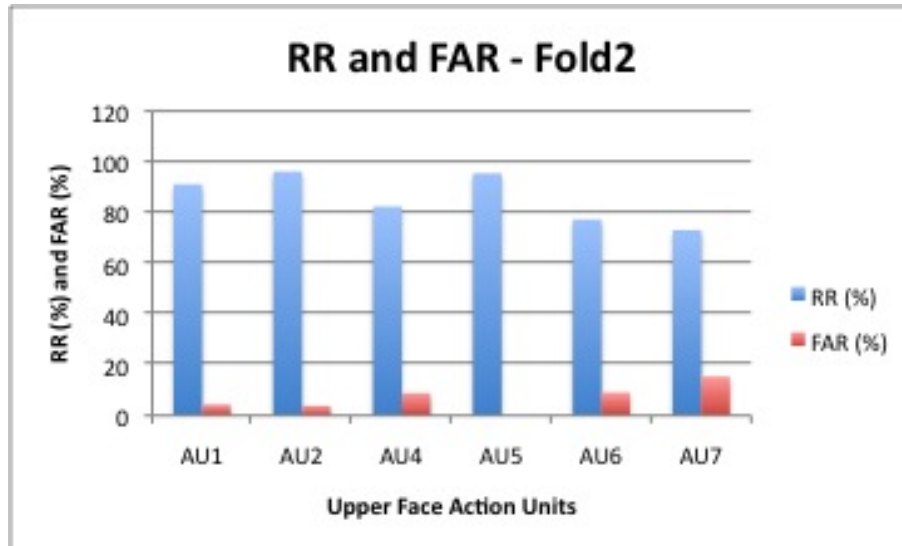
**Figure 11.8:** Recognition (RR %) and false alarm rate (FAR %) for all the upper face AUs obtained during fold 2 using frequency scale selection and RNN classifier



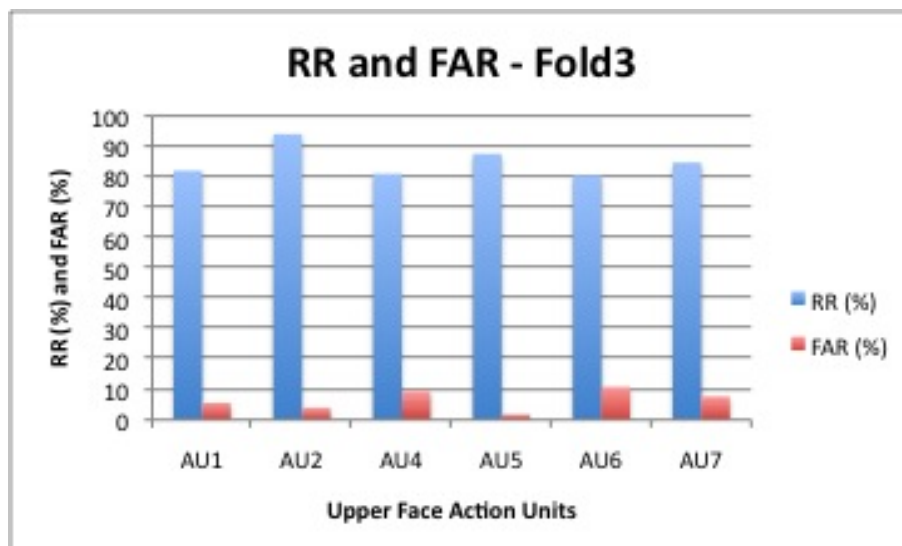
**Figure 11.9:** Recognition (RR %) and false alarm rate (FAR %) for all the upper face AUs obtained during fold 3 using frequency scale selection and RNN classifier



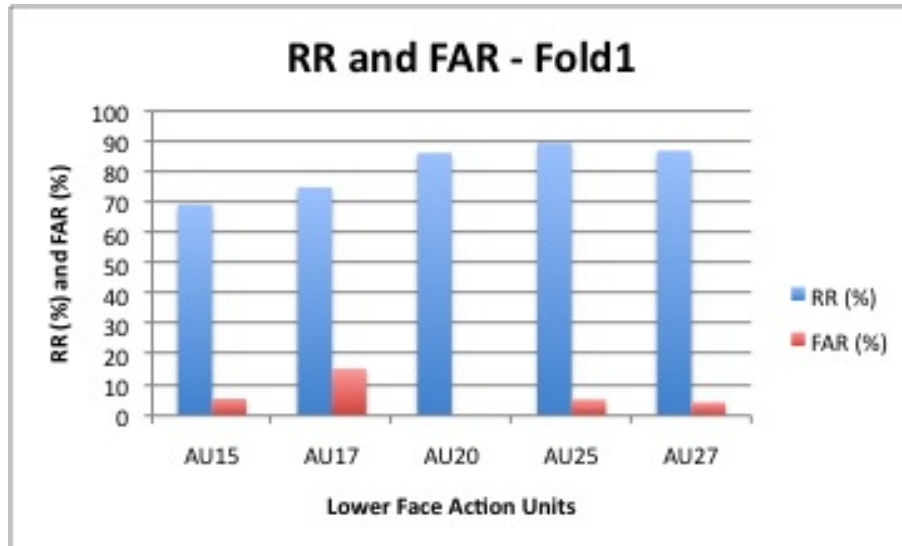
**Figure 11.10:** Recognition (RR %) and false alarm rate (FAR %) for all the upper face AUs obtained during fold 1 using optimized(baseline+weight decay with a decay constant of 0.0001) RNN classifier



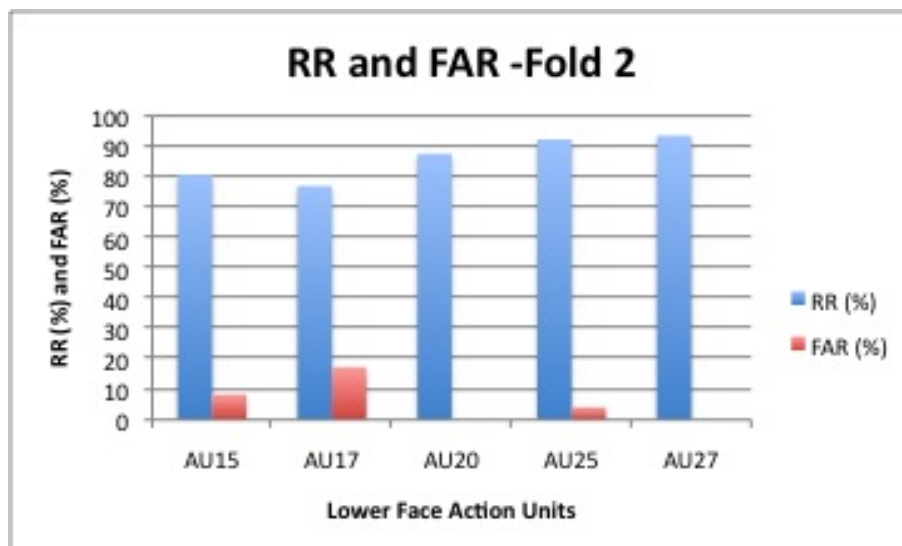
**Figure 11.11:** Recognition (RR %) and false alarm rate (FAR %) for all the upper face AUs obtained during fold 2 using optimized(baseline+weight decay with a decay constant of 0.0001) RNN classifier



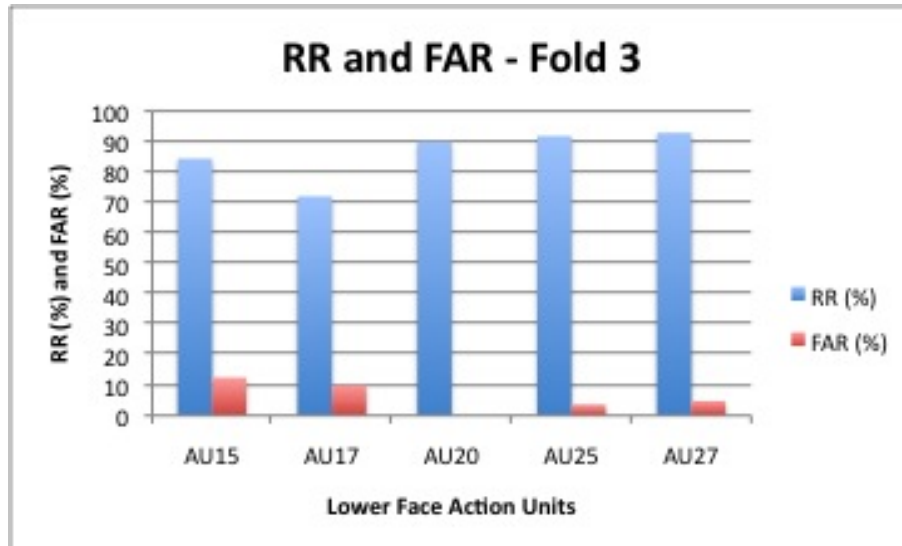
**Figure 11.12:** Recognition (RR %) and false alarm rate (FAR %) for all the upper face AUs obtained during fold 3 using optimized(baseline+weight decay with a decay constant of 0.0001) RNN classifier



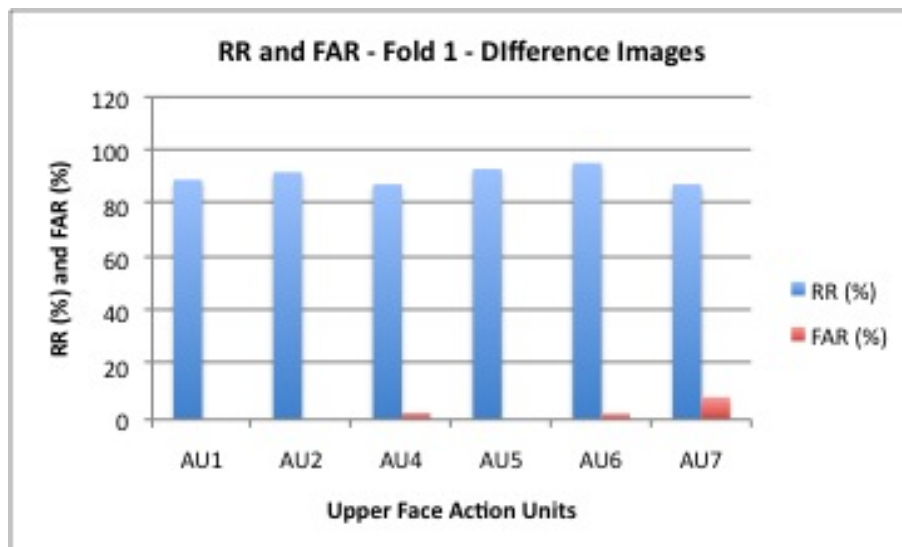
**Figure 11.13:** Recognition (RR %) and false alarm rate (FAR %) for all the lower face AUs obtained during fold 1 using optimized(baseline+weight decay with a decay constant of 0.0001) RNN classifier



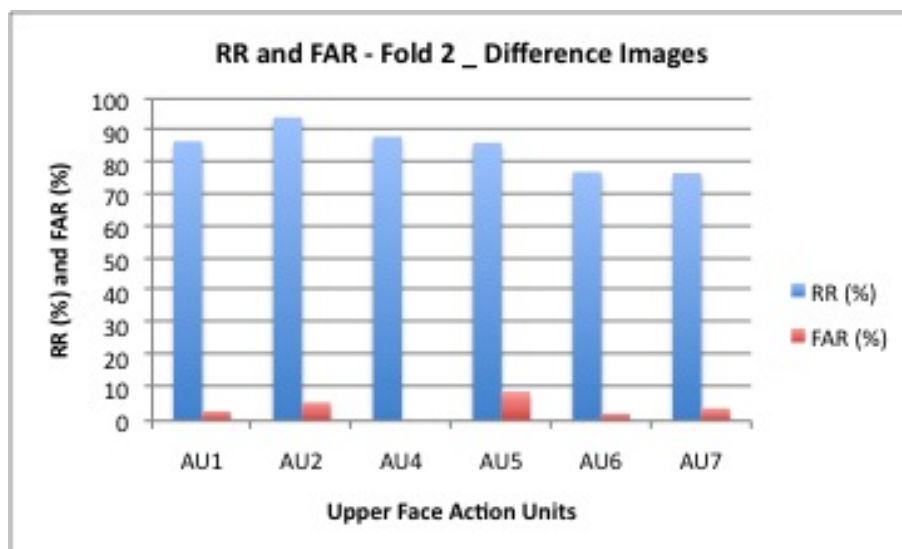
**Figure 11.14:** Recognition (RR %) and false alarm rate (FAR %) for all the lower face AUs obtained during fold 2 using optimized(baseline+weight decay with a decay constant of 0.0001) RNN classifier



**Figure 11.15:** Recognition (RR %) and false alarm rate (FAR %) for all the lower face AUs obtained during fold 3 using optimized(baseline+weight decay with a decay constant of 0.0001) RNN classifier

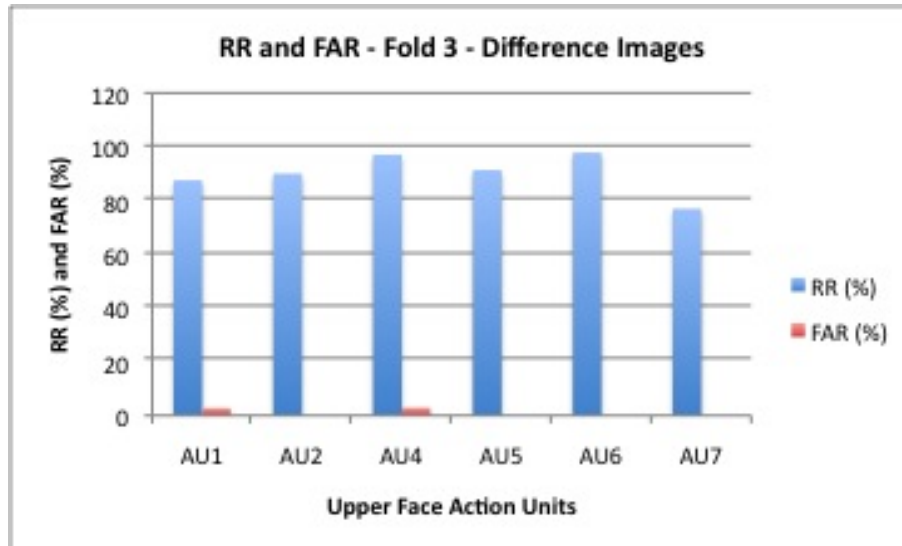


**Figure 11.16:** Recognition (RR %) and false alarm rate (FAR %) for all the upper face AUs obtained during fold 1 using difference images and RNN classifier

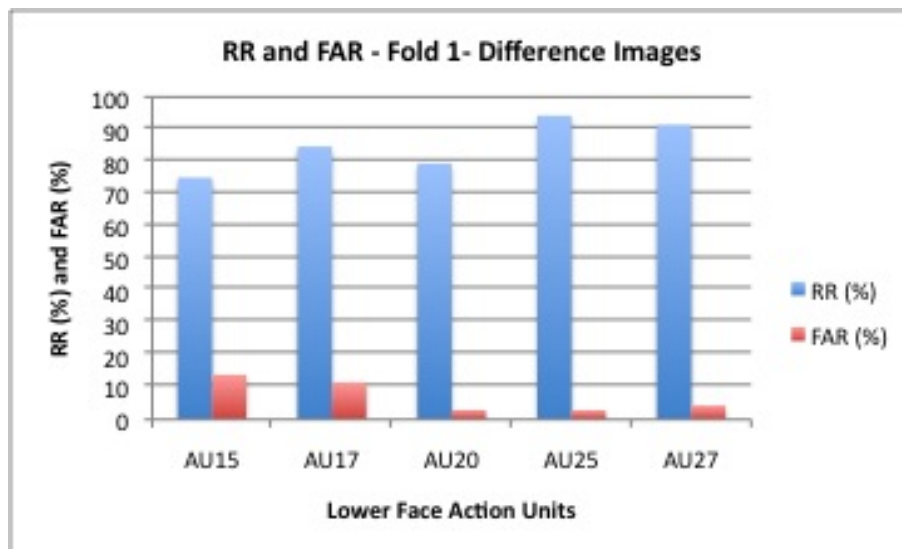


**Figure 11.17:** Recognition (RR %) and false alarm rate (FAR %) for all the upper face AUs obtained during fold 2 using difference images and RNN classifier

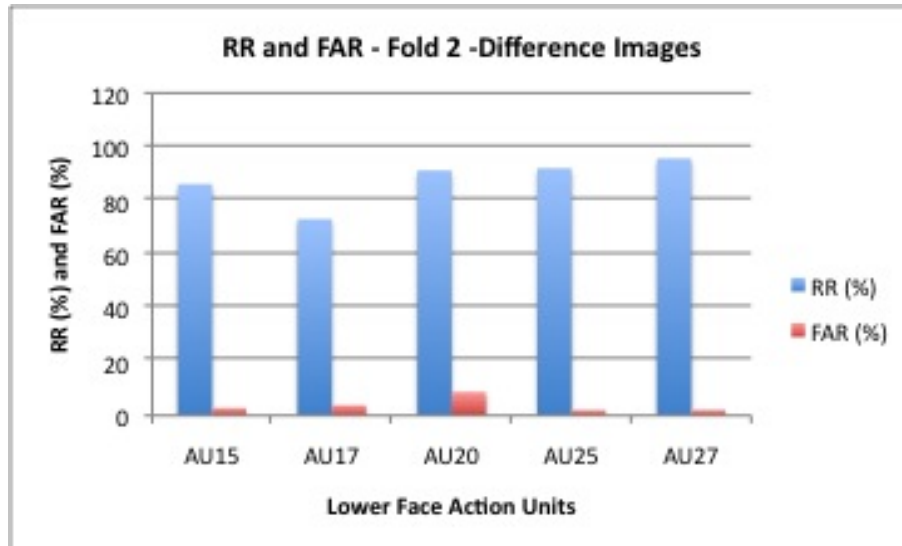




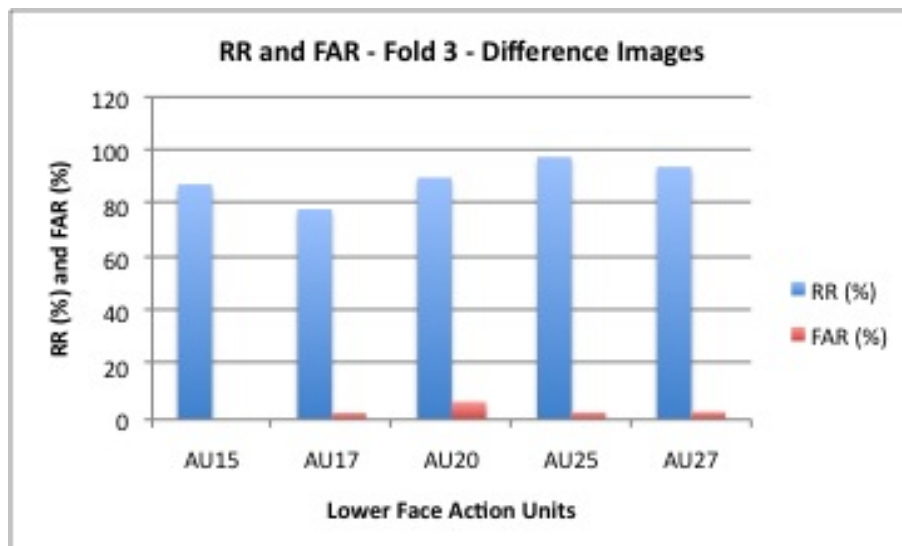
**Figure 11.18:** Recognition (RR %) and false alarm rate (FAR %) for all the upper face AUs obtained during fold 3 using difference images and RNN classifier



**Figure 11.19:** Recognition (RR %) and false alarm rate (FAR %) for all the lower face AUs obtained during fold 1 using difference images and RNN classifier



**Figure 11.20:** Recognition (RR %) and false alarm rate (FAR %) for all the lower face AUs obtained during fold 2 using difference images and RNN classifier



**Figure 11.21:** Recognition (RR %) and false alarm rate (FAR %) for all the lower face AUs obtained during fold 3 using difference images and RNN classifier

# References

- [1] AHLBERG J. **Cnadide-3- An Updtaed Parameterized Face.** *Report No. LiTH-ISY-R-2326, Dept. of Electrical Engineering, Linkoping University, Sweden, 2001.* 37
- [2] AHMAD A.M., ISMAIL S., AND SAMAON D.F. **Recurrent Neural Network with Backpropagation through Time for Speech Recognition.** *In International Symposium on Communications and Information Technologies 2004(ISCIT 2004), Sapporo, Japan, October 26-29, 2004.* 90
- [3] ALEX G., SANTIAGO F., AND JRGEN S. **Bidirectional LSTM Networks for Improved Phoneme Classification and Recognition.** *In Proceedings of the 2005 International Conference on Artificial Neural Networks, Warsaw, Poland, September, 2005.* 81, 103
- [4] ALI G., FEYZULLAH T., KADER E., AND SERDAR C. **Signature Verification Performance of Elman's Recurrent Neural Network.** *Technology, Vol 7, Issue 4, pp. 541-547, 2004.* 90
- [5] AMANDA C., LINTS-MARTINDALE, THOMAS H., BRUCE B., AND STEPHEN J. G. **A Psychophysical Investigation of the Facial Action Coding System as an Index of Pain Variability among Older Adults with and without Alzheimers Disease.** *In Pain Medicine, Vol. 8, Issue 8, pp. 678-689, 2007.* 23
- [6] BARTLETT P.L. **For Valid Generalization, the Size of the Weights is more Important than the Size of the Network.** *In Mozer, M.C., Jordan, M.I., and Petsche, T., (eds) Advances in Neural Information Processing Systems 9, Cambridge, MA: The MIT Press, pp. 134-140, 1997.* 73

## REFERENCES

---

- [7] BARTLETT M. S., HAGER C. J., EKMAN P., AND SEJNOWSKI J. T. **Measuring Facial Expressions by Computer Image Analysis.** *In Psychophysiology, Vol 36(2), pp. 253-263, March, 1999.* 39
- [8] BARTLETT M. S., DONATO G., MOVELLAN R. J., HAGER C. J., EKMAN P., AND SEJNOWSKI J. T. **Image Representations for Facial Expression Coding.** *in Advances in Neural Information Processing Systems, S.A. Solla, T.K. Leen and K.R. Muller, Eds.2000,vol 12,MIT press, 2000.* 40, 42, 45
- [9] BARTLETT M. S., LITTLEWORT G., BRAATHEN B., SEJNOWSKI T. J., AND MOVELLAN J. R. **A Prototype for Automatic Recognition of Spontaneous Facial Actions.** *In Advances in Neural Information Processing Systems,vol 15.MIT Press-S. Becker and K Obermayer(eds.), 2003.* 42, 47, 121
- [10] BARTLETT M., LITTLEWORT G., FASEL I., AND MOVELLAN J. **Real Time Face Detection and Facial Expression Recognition: Development and Applications to Human Computer Interaction.** *In Computer Vision and Pattern Recognition Workshop on Human-Computer Interaction, 2003.* 13, 51, 108
- [11] BARTLETT M. S., LITTLEWORT G., FRANK M., LAINSCSEK C., FASEL I., AND MOVELLAN J. **Machine Learning Methods for Fully Automatic Recognition of Facial Expression and Facial Actions.** *In Proceedings of the IEEE Conference on Systems, Man and Cybernetics, The Hague, Netherlands, 2004.* 3, 65, 112, 113, 131, 132
- [12] BARTLETT M. S., LITTLEWORT G., FRANK M., LAINSCSEK C., FASEL I., AND MOVELLAN J. **Recognizing Facial Expression: Machine Learning and Application to Spontaneous Behavior.** *In IEEE Conference of Computer Vision and Pattern Recognition, 2005.* 10
- [13] BARTLETT M. S., LITTLEWORT G. C., FRANK M. G., LAINSCSEK C., FASEL I. R., AND MOVELLAN J. R. **Automatic Recognition of Facial Actions in Spontaneous Expressions.** *In Journal of Multimedia, 196:22, 2006.* 5, 107, 113
- [14] BASHYAL S. AND VENAYAGAMOORTHY G. K. **Recognition of Facial Expressions using Gabor Wavelets and Learning Vector Quantization.** *In Engineering Applications of Artificial Intelligence, Vol. 21, No. 7, pp. 1056-1064, 2008.* 44

## REFERENCES

---

- [15] BAUM E. B., AND HAUSSLER D. **What Size Nets Get Valid Generalization.** *In Neural Computation, Vol. 1, pp. 151-160, 1989.* 93
- [16] BELHUMEUR P. N., HESPANHA J. P., AND KRIEGMAN. **Eigenfaces vs Fisherfaces: Recognition using Class Specific Linear Projection.** *In IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 19, pp 711-720, 1997.* 40
- [17] BENGIO Y., SIMARD P., AND FRASCONI P. **Learning Long-Term Dependencies with Gradient Descent is Difficult.** *In IEEE Transactions on Neural Networks, Vol. 5, No. 2, pp. 157-166, 1994.* 92
- [18] BLACK M. J., AND YACOOB Y. **Tracking and Recognizing Rigid and Non-Rigid Facial Motions using Local Parametric Models of Image Motion.** *In the Proceedings of Fifth International Conference on Computer Vision, pp 374-381, 1995.* 35
- [19] BLOOM B. S. **The 2 Sigma Problem: The Search for Methods of Group Instruction as Effective as One-to-One Tutoring.** *in Educational Researcher, 1396:4-16, June/July, 1984.* 12
- [20] BOSER B. E., GUYON I. M., AND VAPNIK V. N. **A Training Algorithm for Optimal Marginal Classifiers.** *In D. Haussler Editor, 5th Annual ACM Workshop on COLT, ACM Press, pp. 144-152, 1992.* 73, 78
- [21] BOUREL F., CHIBELUSHI C. C., AND LOW A. A. **Recognition of Facial Expressions in the Presence of Occlusion.** *In Proceedings of the Twelfth British Machine Vision Conference, Vol 1, Manchester, UK, pp.213-222, 2001.* 33
- [22] BOUREL F., CHIBELUSHI C. C., AND LOW A. A. **Robust Facial Expression Recognition using a State Based Model of Spatially Localized Facial Dynamics.** *In Proc. Fifth IEEE International Conference of Automatic Face and Gesture Recognition, pp. 106-111, 2002.* 33
- [23] BRAATHEN B., BARTLETT M. S., LITTLEWORT G., AND MOVELLAN J. R. **First Steps Towards Automatic Recognition of Spontaneous Facial Action Units.** *In Proceedings of ACM International Conference, Vol 15, pp 1-5, 2001.* 36, 37

## REFERENCES

---

- [24] BUTKO N., THEOCHAROUS G., PHILIPOSE M., AND MOVELLAN J. **Automated Facial Affect Analysis for one-on-one Tutoring Applications.** In *IEEE International Conference on Automatic Face and Gesture Recognition*, 2011. 13
- [25] CARIDAKIS G., MALATESTA L., KESSOUS L., AMIR N., RAOUZAIYOU A., AND KARPOUZIS K. **Modeling Naturalistic Affective States via Facial and Vocal Expressions Recognition.** In *Proceedings of the 8th International Conference on Multimodal Interfaces, ACM New York*, 2006. 53, 54, 91
- [26] CHAN WAH NG, SURENDRA RANGANATH. **Real-time Gesture Recognition System and Application.** *Department of Electrical and Computer Engineering, national University of Singapore, December*, 2002. 81
- [27] CHIBELUSHI C. C., AND BOUREL F. **Facial Expression Recognition: A Brief Tutorial Overview.** In *Cvonline: On-Line Compendium of Computer Vision*, 2003. 8
- [28] COHN J., ZLOCHOWER A., LIEN J., WU Y., AND KANADE T. **Automated Face Coding: A Computer-Vision Based Method of Facial Expression Analysis.** In *Seventh European Conference on Facial Expression Measurement and Meaning, Salzburg, Austria, pp. 329-333*, 1997. 52
- [29] COHN J. F., ZLOCHOWER A. J., LIEN J., AND KANADE T. **Automatic Face Analysis by Feature Point Tracking Has High Concurrent Validity with Manual FACS Coding.** In *Psychophysiology, Vol 36, pp 35-43*, 1999. 32
- [30] CORTES C. AND VAPNIK V. **Support-Vector networks.** *Machine Learning, Vol. 20, No. (3), pp. 273-297*, 1995. 79
- [31] COTTRELL G. W. AND METCALFE J. **EMPATH: Face, Gender and Emotion Recognition using Holons.** In *R. P. Lippman, J. Moody, and D. S. Touretzky (Eds.), Advances in Neural Information Processing Systems, Vol. 3, pp. 564-571*, 1991. 51
- [32] COWIE R., DOUGLAS-COWIE E., TSAPATSOULIS N., VOTSIS G., KOLLIAS S., FELLEENZ W., AND TAYLOR J. G. **Emotion Recognition in Human-computer Interaction.** In *IEEE Signal Processing Mag., Vol 18, pp. 32-80*, 2001.
- [33] DANILO P. MANDIC AND JONATHON A. **A Normalized Real Time Recurrent Learning Algorithm.** In *Signal Processing, Vol 80, Issue 9, pp. 1909-1916, Sep*, 2000.

## REFERENCES

---

- [34] DATCU D., AND LEON J.M. ROTHKRANTZ. **Automatic Bi-modal Emotion Recognition System Based on Fusion of Facial Expressions and Emotion Extraction from Speech.** In *IEEE Face and Gesture Conference FG2008*, 2008.
- [35] DAW-TUNG LIN. **Facial Expression Classification Using PCA and Hierarchical Radial Basis Function Network.** In *Journal of Information Science and Engineering*, 22, pp. 1033-1046, 2006.
- [36] DENG HONG-BO, JIN LIAN-WEN, ZHEN LI-XIN AND HUANG JIAN-CHENG. **A New Facial Expression Recognition Methods Based on Local Gabor Filter Bank and PCA plus LDA.** In *International Journal of Information Technology*, vol 11, no. 11, pp. 86-96, 2005.
- [37] DONATO G., BARTLETT M. S., HAGER J. C., EKMAN P., AND SEJNOESKI T. **J. Classifying Facial Actions.** In *IEEE Transactions on Patterns Analysis and Machine Intelligence*, Vol 21, Issue 10, pp. 974-989, October, 1999.
- [38] DOUGLES E., AND JUERGEN S. **A First Look at Music Composition using LSTM Recurrent Neural Networks.** *Technical Report, IDSIA*, 2002.
- [39] ELMAN J. **Finding Structure in Time.** In *Cognitive Science*, Vol 14, pp. 179-211, 1990.
- [40] ELLGRING H. **Nonverbal Expression of Psychological States in Psychiatric Patients.** In *European Archives of Psychiatry and Neurological Sciences*, Vol 236, no. 1, pp. 31-34, 1986.
- [41] ESSA I.A., AND PENTLAND A.P. **Coding, Analysis, Interpretation and Recognition of Facial Expressions.** In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol 19, No.7, pp. 757-763, July, 1997.
- [42] EKMAN P., AND FRIESEN W. V. **Constants Across Cultures in the Face and Emotion.** In *Journal of Personality Social Psychology*, Vol 17, No. 2, pp. 124-129, 1971.
- [43] EKMAN P., AND FRIESEN W. **The Facial Action Coding System: A Technique For the Measurement of Facial Movement.** In *Consulting Psychologists Press, San Francisco, CA*, 1978.
- [44] EKMAN P., AND FRIESEN W. **Rationale and Reliability For EMFACS Coders.** *Unpublished*, 1982.

## REFERENCES

---

- [45] EKMAN P., FRIESEN W., AND HAGER J. C. **Facial Action Coding System (FACS)**. *A Human Face*, Salt Lake City, 2002.
- [46] EKMAN. **Methods for Measuring Facial Action**. In *Handbook of Methods in Nonverbal Behavior Research*, K.R. Scherer and P. Ekman, Eds., pp. 45-90, Cambridge University Press, Cambridge, 1982.
- [47] FADI D., AND FRANCK D. **Facial Expression Recognition in continuous Videos using Linear Discriminant Analysis**. In *the Proceedings of IAPR Conference on Machine Vision Applications*, pp. 277-280, May, 2005.
- [48] FASEL I., DAHL R., HERSHEY J., FORTENBERRY B., SUSSKIND J., AND MOVELLAN J.R.. **Machine Perception Toolbox-MPISearch**. *Machine Perception Laboratory, University of California San Diego*.
- [49] FASEL B., AND LUTTIN J. **Recognition of Asymmetric Facial Action Unit Activities and Intensities**. In *Proceedings of ICPR 2000, Barcelona, Spain*, 2000.
- [50] FASEL B., AND LUETTIN J. **Automatic Facial Expression Analysis: A Survey**. In *Pattern Recognition, Vol 36, No. 1*, pp. 259-275, January, 2003.
- [51] FELIX A. G., NICOL N. S., AND JURGEN S. **Learning Precise Timing with LSTM Recurrent Networks**. in *Nournal of Machine Learning Research, Vol. 3*, pp. 115-143, 2002.
- [52] FRANK M.G. **Assessing Deception: Implications For the Courtroom**. In *The Judicial Review, Vol. 2*, pp. 315-326, 1996.
- [53] FREUND Y., AND SCHAPIRE R. E. **A Short Introduction to Boosting**. In *Journal of Japanese Society For Artificial Intelligence, Vol. 14, No. 5*, pp. 771-780, September, 1999.
- [54] GERS F. A., SCHMIDHUBER J., AND CUMMINS F. **Learning to Forget: Continual Prediction with LSTM**. In *Neural Computation, Vol. 12, No. 10*, pp. 2451-2471, 2000.
- [55] GILES C. L., MILLER C. B., CHEN D., CHEN H. H., SUN G. Z., AND LEE Y. C. **Learning and Extracting Finite State Automata with Second-order Recurrent Neural Networks**. In *Neural Computation, Vol. 4, No. 3*, pp. 393-405, 1992.



## REFERENCES

---

- [56] GILES C. L. AND OMLIN C. W. **Pruning Recurrent Neural Networks for Improved Generalization Performance.** In *IEEE Transactions on Neural Networks*, Vol. 5, No. 5, pp. 848-851, September, 1994.
- [57] GRAVES A., MAYER C., WIMMER M., SCHMIDHUBER J., AND RADIG B. **Facial Expression Recognition With Recurrent Neural Networks** In *Proceedings of the International Workshop on Cognition for Technical Systems, Munich, Germany, October, 2008.*
- [58] HAI TAO, CHEN H., AND HUANG T. **Analysis and Compression of Facial Animation Parameters Set (FAPs).** In *IEEE First Workshop on Multimedia Signal Processing, Princeton, USA, pp. 245-250, June, 1997.*
- [59] HAYKIN S. **Neural Networks: A Comprehensive Foundation.** Prentice Hall, New Jersey, second ed. ISBN: 0-13-273350-1, 1999.
- [60] HELLER M., AND HAYNAL V. **Depression and Suicide Faces.** In *Ekman, P. and Rosenberg, E.L. (eds.): What the face reveals. Oxford, England : Oxford University Press, pp 339-407, 1997.*
- [61] HINTON G. E. **Learning translation invariant recognition in a massively parallel network.** In *PMLE: Parallel Architectures and Languages Europe, in Lecture Notes in Computer Science. Berlin: Springer-Verlag, pp. 1-13, 1987.*
- [62] IZARD C.E. **The Maximally Discriminative Facial Movement Coding System.** *Academic Computing Services and University Media services, University of Delaware, Newark, Delaware, Revised Edition, 1983.*
- [63] IZARD C.E., DOUGHERTY L.M., AND HEMBREE E.A. **A System For Identifying Affect Expressions by Holistic Judgments.** *Unpublished Manuscript, University of Delaware, 1983.*
- [64] JOSE V. R., BELEN M., JUAN J. P., AND ANGEL S. **Comparing Feature Point Tracking with Dense Flow Tracking For Facial Expression Recognition.** *Bioinspired Applications in Artificial and Natural Computation, Lecture Notes in Computer Science, Springer Berlin/Heidelberg, Vol. 5602, pp.264-273, 2009.*
- [65] JORDAN M.I. **Attractor Dynamics and Parallelism in a Connectionist Sequential Machine.** In *Proc. of the Ninth Annual Conference of the Cognitive Science Society, Lawrence Erlbaum, pp. 531-546, 1986.*

## REFERENCES

---

- [66] KANADE T., COHN J., AND TIAN Y. **Comprehensive Database for Facial Expression Analysis.** *In Proceedings of 4th IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 46-53, 2000.
- [67] KIM J. **Bimodal Emotion Recognition using Speech and Physiological Changes,** *In Robust Speech Recognition and Understanding*, Book edited by; Michael Grimm and Kristian Kroschel, Vienna, Austria, 2007.
- [68] KOBAYASHI H., AND HARA F. **Dynamic Recognition of Basic Facial Expressions by Discrete- time Recurrent Neural Network.** *In Proceedings of the International Joint Conference on Neural Networks*, pp. 155-158, 1993.
- [69] KONDRATENKO V.V., AND KUPERIN YU. A. **Using Recurrent Neural Networks To Forecasting of Forex.** *In Disordered Systems and Neural Networks*, April, 2003.
- [70] KOTSIA IRENE AND PITAS IOANNIS. **Facial Expression Recognition in Image Sequences using Geometric Deformation Features and Support Vector Machines.** *In IEEE Transactions on Image Processing*, Vol. 16, No. 1, pp. 172-187, 2007.
- [71] KROGH A., AND HERTZ J. A. **A Simple Weight Decay Can Improve Generalization.** *In Advances in Neural Information Processing Systems 4*, J. E Moody, S J Hanson and R P Lippmann eds. Morgan Kauffmann Publishers, San Mateo CA, pp-950-957, 1995.
- [72] KROON R. S. **Support Vector Machines, Generalization Bounds and Transduction.** *Master's thesis, Department of Computer Science, University of Stellenbosch*, 2003.
- [73] KROON S. AND OMLIN C. W. **Getting to Grips with Support Vector Machines: Theory.** *In South African Statistical Journal*, Vol. 28, No. 2, pp. 93-114, 2004.
- [74] LADES M., VORBRUGGEN J., BUHMANN J., LANGE J., VON DER MALSBURG C. AND WURTZ R. KONEN. **Distortion Invariant Object Recognition in the Dynamic Link Architecture** *In IEEE Transactions on Computers*, Vol.42, No.3, pp. 300- 311, 1993.

## REFERENCES

---

- [75] LEON J.M. ROTHKRANTZ AND DAAN NOLLEN. **Automatic Speech Recognition using Recurrent Neural Networks.** *In Neural Network World, Vol.10, No.3, pp. 445-453, July, 2000.*
- [76] LIEN J. **Automatic Recognition of Facial Expressions using Hidden Markov Models and Estimation of Expression Intensity.** *Ph.D. Dissertation, Carnegie Mellon University, Pittsburg, PA, 1998.*
- [77] LIEN J. J., KANADE T., COHN J., AND LI C. **Automated Facial Expressions Based on FACS Action Units.** *In Third IEEE International Conference on Automatic Face and Gesture Recognition, pp. 390-395, April, 1998.*
- [78] LIN S. H., HSIEH P. F., AND WU C. H. **Facial Phoneme Extraction for Taiwanese Sign Language Recognition.** *In Affective Computing and Intelligence Interaction, Lecture Notes in Computer Science, Vol 3784/2005, pp. 187-194, 2005.*
- [79] LISA A. P., BRIDGET M. W., AND SARAH J. V. **Classifying Chimpanzee Facial Expressions Using Muscle Action** *In Emotion, Vol 7, Issue 1, pp. 172-181, February, 2007.*
- [80] LITTLEWORT-FORD G., BARTLETT M. S., AND MOVELLAN J. R. **Are Your Eyes Smiling? Detecting Genuine Smiles with Support Vector Machine and Gabor Wavelets.** *In Proceedings of the 8th Annual Joint Symposium on Neural Computation, 2001.*
- [81] LITTLEWORT G., BARTLETT M. S., FASEL I., SUSSKIND J., AND MOVELLAN J. **Dynamics of Facial Expression Extracted Automatically from Video.** *In IEEE Conference on Computer Vision and Pattern Recognition, Workshop on Face Processing in Video, Vol. 5, pp. 80, June, 2004.*
- [82] LYONS M. J., BUDYNEK J., PLANTE A., AND AKAMATSU S. **Classifying Facial Attributes using a 2D Gabor Wavelet Representation and Discriminant Analysis.** *In Proceedings of the 4th International Conference on Automatic Face and Gesture Recognition, Grenoble France, IEEE Computer Society, pp. 202-207, 2000.*
- [83] MARCUS L., ALEX G., SANTIAGO F., HORST B., AND JRGEN S. **A Novel Approach to On-line Handwriting Recognition Based on Bidirectional**

- long Short-Term Memory Networks.** *In Proceedings of the 9th International Conference on Document Analysis and Recognition, ICDAR 2007, Curitiba, Brazil, September, 2007.*
- [84] MAYER H., NAGY I., KNOLL A., SCHIRMBECK E. U., AND BAUERNSCHMITT R. **The EndoPAR system for minimally invasive surgery.** *In International Conference on Intelligent Robots and Systems (IROS), Sendai, Japan, 2004.*
- [85] MAYER H., GOMEZ F., WIERSTRA D., NAGY I., KNOLL A., AND SCHMIDHUBER J. **A System for Robotic Heart Surgery that Learns to Tie Knots Using Recurrent Neural Networks.** *In Advanced Robotics, Vol. 22, No. 13-14, pp. 1521-1537, 2008.*
- [86] MICHAEL N., METAXAS D. N., AND NEIDLE C. **Spatial and Temporal Pyramids for Grammatical Expression Recognition of American Sign Language.** *In Proceedings of the 11th International ACM SIGACCESS Conference on Computers and Accessibility, ASSETS'09, pp. 75-82, 2009.*
- [87] MOVELLAN J. R. **Tutorial on Gabor Filters,** 2008.
- [88] MULLER S., WALLHOFF F., HULSKEN F., AND RIGOLL G. **Facial Expression Recognition Using Pseudo 3-D Hidden Markov Models.** *In 16th Int. Conference on Pattern Recognition (ICPR), Vol. 2, pp. 32-35, 2002.*
- [89] NARENDRA K.S., AND PARTHASARATHY K. **Identification and Control of Dynamical Systems using Neural Networks.** *In IEEE Transactions on Neural Networks, Vol 1, No. 1, pp. 4-27, 1990.*
- [90] OLIVER N., PENTLAND A., AND BERARD F. **LAFTER: A Real-time Lips and Face Tracker with Facial Expression Recognition.** *In Pattern Recognition, Vol. 33, No. 8, pp. 1369-1382, 2000.*
- [91] OSTER H. **Baby FACS: Facial Action Coding System for Infants and Young Children.** *Unpublished Monograph and Coding Manual, New York University, 2003.*
- [92] OTSUKA T., AND OHYA J. **Spotting Segments Displaying Facial Expression from Image Sequences using HMM.** *In IEEE Proceedings of the Second International Conference on Automatic Face and Gesture Recognition (FG98), Nara, Japan, 1998, pp. 442-447, 1998.*

## REFERENCES

---

- [93] PARKE F.I., AND WATERS K. **Computer Facial Animation**. Wellesley, MA: A.K. Peters, 1996.
- [94] PANTIC M., ROTHKRANTZ L. J. M. **Automatic Analysis of Facial Expressions: The State of the Art**. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol 22, No. 12, pp. 1424-1445, 2000.
- [95] PETAR S. A., AND AGGELOS K. K. **Automatic Facial Expression Recognition Using Facial Animation Parameters and Multistream HMMs**. In *IEEE Transactions on Information Forensics and Security*, Vol 1, No. 1, pp. 3-11, March, 2006.
- [96] PETER TINO, CHRISTIAN SCHITTENKOPF, AND GEORG DORFFNER. **Financial Volatility Trading using Recurrent Neural Networks**. In *IEEE Transactions on Neural Networks*, Vol 12, No. 4, pp. 865-874, July, 2001.
- [97] PHILIPP M., AND RANA E. K. **Real Time Facial Expression Recognition in Video Using Support Vector Machines**. In *Proceedings of the 5th International Conference on Multimodal Interfaces, ICMI03*, pp. 258-264, 2003.
- [98] PYUN H., KIM Y., CHAE W., KANG H., AND SHIN S. **Facial Animation and Hair: An Example Approach for Facial Expression Cloning**. In *ACM SIGGRAPH Eurographics Symposium on Computer Animation SCA 03, California*, 2003.
- [99] QUEN- ZONG WU, I-CHANG JOU AND SUH-YIN LEE. **On-line Signature Verification using Cepstrum and Neural Networks**. In *IEEE Transactions on Systems, Man and Cybernetics*, Vol 27, No. 1, pp. 148-153, 1997.
- [100] RABINER L. R. **A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition**. In *Proceedings of the IEEE*, Vol. 77, No. 2, February, 1989.
- [101] ROSENBERG E. **Introduction: The Study of Spontaneous Facial Expressions in Psychology**. In *Ekman, P. and Rosenberg, E. (Eds.), (1997). What the face reveals: Basic and applied studies of spontaneous expressions using the Facial Action Coding System (FACS)*. (pp. 3-18). Oxford University Press: New York, 1997.

## REFERENCES

---

- [102] ROSENBLUM M., YACOOB Y., AND DAVIS L. **Human Expression Recognition from Motion using a Radial Basis Function Network Architecture.** *In IEEE Trans. Neural Networks, Vol 7, No. 5, pp. 1121-1138, 1996.*
- [103] RUSSELL J.A. **Is There Universal Recognition of Emotion from Facial Expression? A Review of the Cross-cultural Studies.** *In Psychological Bulletin, Vol 115, No. 1, pp. 102-141, January, 1994.*
- [104] SAKO H., AND SMITH A. V. W. **Real Time Facial Expression Recognition Based on Features Position and Dimensions.** *In Proceedings of the 13th International Conference on Pattern Recognition(ICPR), Vol 3, pp. 643-648, 1996.*
- [105] SCHMIDHUBER J., AND HOCHREITER S. **Guessing can Outperform Many Long Time Lag Algorithms.** *Technical Report IDSIA, IDSIA, 1996.*
- [106] SEPP H. **The Vanishing Gradient Problem During Recurrent Neural Nets and Problem Solutions.** *In International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, Vol. 6, No. 2, april, 1998.*
- [107] SEYEDARABI H., AGHAGOLZADEH A., AND KHANMOHAMMADI S. **Recognition of Six Basic Expressions by Feature-points Tracking using RBF Neural Network and Fuzzy Interface System.** *In IEEE Conference on Multimedia and Expo, ICME, Vol 2, pp. 1219-1222, 2004.*
- [108] SMITH E., BARTLETT M.S., AND MOVELLAN J.R. **Computer Recognition of Facial Actions: A Study of Co-articulation Effects.** *In Proceedings of the 8th Annual Joint Symposium on Neural Computation, 2001.*
- [109] THEEKAPUN C., TOKIA S., AND HASE H. **Facial Expression Recognition from a Partial Face Image by Using Displacement Vector.** *In the Proceedings of 5th International Conference on Electrical/Electronics, Computer, telecommunications and Information Technology, ECTI-CON, pp.441-444, 2008.*
- [110] TIAN Y., KANADE T., AND COHN J. F. **Eye State Action Unit Detection by Gabor Wavelets.** *In Proceedings of the Third International Conference on Advances in Multimodal Interfaces, ICMI'00, pp. 143-150, 2000.*
- [111] TIAN Y., KANADE T., AND COHN J. F. **Recognizing Action Units for Facial Expression Analysis.** *In IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 23, No. 2, pp. 97-115, February, 2001.*

- [112] TIAN Y, KANADE T AND COHN J F.. **Evaluation of Gabor Wavelet Based Facial Action Unit Recognition in Image Sequences of Increasing Complexity.** *In Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition (FGR'02)*, pp. 229-234, May, 2002.
- [113] TONY ROBINSON. **The Application of Recurrent Nets to Phone Probability Estimation.** *In IEEE Transactions on Neural Networks*, Vol 5, No. 2, pp. 298-305, 1994.
- [114] VIOLA P., AND JONES M. J. **Robust Real-Time Face Detection.** *In International Journal of Computer Vision*, Vol. 57, No. 2, pp. 137-154, 2004.
- [115] VOGLER C., AND GOLDENSTEIN S. **Analysis of Facial Expressions in American Sign Language.** *In Proceedings of the 3rd Intl. Conference on Universal Access in Human-Computer Interaction (UAHCI)*, Las Vegas, 2005.
- [116] WERBOS P. **Backpropagation: Past and Future.** *In Proceedings of the IEEE International Conference on Neural Networks*, IEEE Press, pp. 343-353, 1988.
- [117] WERBOS J. PAUL. **Back Propagation Through Time: What it Does and How to do it.** *In Proceedings of the IEEE*, Vol 78, No. 10, pp. 1550-1560, October, 1990.
- [118] WHITEHILL J. **Automatic Real-Time Facial Expression Recognition for Signed Language Translation.** *M.Sc. thesis, University of the Western Cape (South Africa)*, 2006.
- [119] WHITEHILL J., AND OMLIN C. W. **Local versus Global Segmentation for Facial Expression Recognition.** *In Proceedings of 7th International Conference on Automatic Face and Gesture Recognition*, pp. 357-362, April, 2006.
- [120] WHITEHILL J., BARTLETT M., AND MOVELLAN J. **Automatic Facial Expression Recognition for Intelligent Tutoring Systems.** *In 2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, CVPRW'08*, pp. 1-6, June, 2008.
- [121] WILLIAMS R.J., AND ZISPER D. **A Learning Algorithm for Continually Running Fully Recurrent Neural Networks.** *In Neural Computation*, Vol 1, No. 2, pp. 270-280, March, 1989.

## REFERENCES

---

- [122] YACOOB Y., AND DAVIS L. S. **Recognizing Human Facial Expressions from Long Sequences using Optical Flow.** *In IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 18, No. 6, pp. 636-642, June, 1996.*
- [123] YUN C., DENG Z., AND HISCOCK M. **Can Local Avatars Satisfy a Global Audience? A Case Study of High-fidelity 3D Facial Avatar Animation in Subject Identification and Emotion Perception by US and International Groups.** *In CIE Computers in Entertainment, ACM New York, Vol 7, No.2, June, 2009.*
- [124] ZHANG Z. **Feature Based Facial Expression Recognition: Sensitivity Analysis and Experiments With a Multi Layer Perception.** *Technical Report 3354, INRIA Sophia Antopolis, 1998.*
- [125] ZHANG Z., LYONS M., SCHUSTER M., AND AKAMATSU S. **Comparison Between Geometry Based and Gabor Wavelets Based Facial Expression Recognition Using Multi Layer Perception.** *In Proceedings of Third IEEE International Conference on Automatic Face and Gesture Recognition, Nara Japan, IEEE Computer Society, pp. 454-459, 1998.*