

https://dspace.org/

A novel facial action intensity detection system

Bingol, Deniz

2015-01-30
http://hdl.handle.net/10539/16832

Downloaded from a DSpace repository. https://dspace.org/

University of the Witwatersrand

Masters Thesis

A Novel Facial Action Intensity Detection System

Author:
Deniz Bingöl

Supervisor:
Dr. Hima B. Vadapalli
Co-Supervisor:
Prof. Turgay Çelik

A thesis submitted in fulfilment of the requirements for the degree of Master of Science

in the

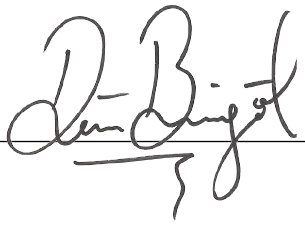
School of Computer Science

October 2014

Declaration of Authorship

I, Deniz Bingöl, declare that this thesis titled, 'A Novel Facial Action Intensity Detection System' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: 

Date:

"If we knew what it was we were doing, it would not be called research, would it?"

Albert Einstein

UNIVERSITY OF THE WITWATERSRAND

Abstract

Faculty of Science
School of Computer Science

Master of Science

A Novel Facial Action Intensity Detection System

by Deniz BINGÖL

Despite the fact that there has been quite a lot of research done in the field of facial expression recognition, not much development has occurred in detecting the intensity of facial actions. In facial expression recognition, the intensity of facial actions is an important and crucial aspect, since it would provide more information about the facial expression of an individual, such as the level of emotion in a face. Furthermore, having an automated system that can detect the intensity of facial actions in an individual's face can lead up to a lot of potential applications from lie detection to smart classrooms. The provided approach includes robust methods for face and facial feature extraction, and multiple machine learning methods for facial action intensity detection.

Acknowledgements

I would like to thank my family and my friends for their never ending support.

I would like to thank my thesis supervisor Dr. Hima B. Vadapalli, and co-supervisor Prof. Turgay Çelik for their supervision which lead to my success in my degree.

I would like to thank the former head of school of the School of Computer Science, Prof. Christian W. Omlin for giving me the opportunity to study at the University of the Witwatersrand, and for his financial and mental support.

Last but not least, I would like to thank his noodly appendage, the Flying Spaghetti Monster for making it possible for me to finalize this thesis ...

Publications

- **Title**: Facial Action Unit Intensity Estimation Using Rotation Invariant Features and Regression Analysis.
  Authors: Deniz Bingöl, Turgay Çelik, Christian W. Omlin, Hima B. Vadapalli
  Accepted for IEEE International Conference on Image Processing 2014 [1].

- **Title**: Automatic Pain Intensity Detection Using Computer Vision and Machine Learning
  Authors: Deniz Bingöl, Turgay Çelik, Hima B. Vadapalli.
  Submitted for review to IEEE Transactions on Pattern Recognition.

Contents

Declaration of Authorship i

Abstract iii

Acknowledgements iv

Publications v

Contents vi

List of Figures ix

List of Tables xi

Abbreviations xii

Physical Constants xiv

Symbols xv

1 Introduction 1
- 1.1 Motivation . . . 1
- 1.2 Research Questions and Hypothesis . . . 2
- 1.3 Technical Objectives . . . 2
- 1.4 Research Methodology . . . 2
- 1.5 Accomplishments . . . 3
- 1.6 Thesis Overview . . . 4

2 Automated Facial Expression Recognition 5
- 2.1 Introduction . . . 5
- 2.2 What is Automated Facial Expression Recognition? . . . 5
  - 2.2.1 General definition . . . 5
  - 2.2.2 Different coding systems . . . 6
  - 2.2.3 Computational methods . . . 6
- 2.3 Facial Action Coding System . . . 6

2.3.1 What is Facial Action Coding System . . . 6
2.3.2 Action Units and Intensities . . . 8
2.3.3 Action Unit Combinations . . . 9
2.4 Importance and Potential Areas . . . 10
2.5 Limitations and Challenges . . . 11
2.5.1 Facial Figure Challenge and Generalization . . . 11
2.5.2 Speed of the System . . . 11
2.5.3 Limitation of Available Data . . . 12
2.6 Summary . . . 12

3 Computational Background **13**
3.1 Introduction . . . 13
3.2 Digital Image Processing . . . 13
3.2.1 Introduction . . . 13
3.2.2 Image Filtering and Convolution . . . 14
3.2.3 Histogram . . . 15
3.2.4 Sampling . . . 16
3.3 Computer Vision . . . 17
3.3.1 Introduction . . . 17
3.3.2 Face Detection . . . 17
3.3.3 Feature Extraction . . . 17
3.3.4 Object Tracking . . . 18
3.4 Supervised Machine Learning . . . 18
3.4.1 What is Machine learning, and Supervised Machine Learning? . . . 18
3.4.2 Support Vector Machines . . . 19
3.4.3 AdaBoost . . . 19
3.4.4 Genetic Algorithms . . . 21
3.4.5 Gradient Descent . . . 21
3.4.6 Regression Analysis . . . 21
3.4.7 Neural Networks . . . 22
3.5 Summary . . . 23

4 Literature Survey **24**
4.1 Introduction . . . 24
4.2 Databases and Data Retrieval . . . 25
4.2.1 The Japanese Female Facial Expression Database . . . 25
4.2.2 Cohn Kanade Database . . . 25
4.2.3 BU-3DFE . . . 26
4.2.4 Bosphorus Database . . . 27
4.2.5 UNBC McMasters Painful Dataset . . . 27
4.2.6 DISFA . . . 28
4.3 Face Detection and Extraction . . . 28
4.4 Geometry Based FER Methods . . . 30
4.4.1 Active Appearance Models . . . 30
4.4.2 3D Analysis . . . 31
4.5 Appearance Based FER Methods . . . 33
4.5.1 Gabor Filters . . . 33

4.5.2 Haar-Like Features . . . 35
4.5.3 Local Binary Patterns . . . 36
4.6 Comparison Between Geometry and Appearance Based FER . . . 37
4.6.1 Recognition Success Rate . . . 38
4.6.2 Speed . . . 38
4.7 Hybrid Systems . . . 38
4.8 FACS AU Intensity Detection . . . 39
4.9 Pain Analysis by Facial Expression Recognition . . . 40
4.9.1 Pain Detection . . . 41
4.9.2 Pain vs Fake Pain . . . 42
4.9.3 Pain in Neonates . . . 43
4.9.4 Pain Intensity . . . 43
4.10 Summary . . . 45

5 Methodology **46**
5.1 Introduction . . . 46
5.2 Feature Extraction . . . 46
5.2.1 Facial Grid and Normalization . . . 46
5.2.2 Angular Features . . . 48
5.2.3 Vectorial Features . . . 49
5.3 Feature Selection . . . 50
5.4 Classification . . . 51
5.4.1 Logistic Regression . . . 51
5.4.2 Feed Forward Neural Networks . . . 52
5.5 Summary . . . 55

6 Experiments, Results and Discussion **56**
6.1 Introduction . . . 56
6.2 FACS AU Intensity Detection . . . 56
6.2.1 Experimental Setup . . . 57
6.2.2 Experiment Results . . . 58
6.2.3 Experiment Discussion . . . 60
6.3 Pain Intensity Detection . . . 62
6.3.1 Experimental Setup . . . 63
6.3.2 Experiment Results . . . 64
6.3.3 Experiment Discussion . . . 65
6.4 Summary . . . 68

7 Thesis Summary and Conclusion **69**
7.1 Summary . . . 69
7.2 Research Questions and Answers . . . 69
7.3 Shortcoming and Challenges . . . 70
7.4 Future Work . . . 71
7.5 Conclusion . . . 71

List of Figures

2.1 Facial areas defined in Table 2.1 [2] . . . 8
2.2 Muscular anatomy under face skin and the corresponding AUs to that region [2]. . . . 10
2.3 Visual representation of FACS AU intensity levels [2]. . . . 10

3.1 Example filters for image processing. . . . 14
3.2 Original and convolved images with blurring and edge detection filters. . . . 15
3.3 A wild gray-scale image and its histogram. . . . 15
3.4 An example of a linear classification border trained by SVM 20
3.5 Linear and Logistic Regression Analysis comparison. . . . 22
3.6 Basic structure of Neural Networks. . . . 22

4.1 Images from JAFFE Database [3]. . . . 25
4.2 First and last images of sequence 1 of candidate 52 in CK+ [4]. . . . 26
4.3 Example 2D and 3D images from BU-3DFE [5]. . . . 26
4.4 Occluded 3D images from Bosphorus Database [6]. . . . 27
4.5 Some sequence of images from the UNBC-McMaster Database [7] with their PSPI pain scale labels. . . . 28
4.6 Some sections of images and their AU intensity coding from DISFA [8] . . . 29
4.7 Two wild images that Tzimiropoulos et al. have experimented on [9]. . . . 32
4.8 Curves extracted by Maalej et al. [10]. . . . 33
4.9 Real and Imaginary parts of Gabor filters created with 5 frequencies and 8 orientations. . . . 34
4.10 Gabor responses on images [11]. . . . 34
4.11 Some examples of filters for extracting Haar-Like features. . . . 35
4.12 Extracting Haar-Like features from human face [12]. . . . 36
4.13 Local binary pattern representation of an example binary string. . . . 37
4.14 Facial analysis produced by Shan et al. [13]. . . . 37
4.15 Texture component analysis by [14]. . . . 39
4.16 Extracted features using fiducial points and image filters by Jeni et al. [15]. 41
4.17 S-PTS, S-APP and C-APP used by Lucey et al. [16]. . . . 42

5.1 Facial feature points given in DISFA database [8]. . . . 47
5.2 AAM grids before and after normalization. . . . 48
5.3 Example P_C, P_L, P_R points, and vectors for feature extraction [8]. . . . 49
5.4 Selected Feature Points for Feature Extraction. . . . 50
5.5 Vectorial features extracted from the upper grid of the face. . . . 51
5.6 The structure of the 1 hidden layer, multi-class neural network 53

6.1 Visual representations of the confusion matrices given in Table 6.5. 60
6.2 Visual representations of the confusion matrices given in Table 6.9. 65
6.3 Some sequence of images from the UNBC-McMaster Database [7] with their PSPI pain scale labels. 66
6.4 The results of Lucey et al. [16] by means of area under the ROC curve. . 67

List of Tables

2.1 Face areas and definitions [2]. 7
2.2 FACS AUs selected for this research [17]. 9
2.3 FACS AU intensity level descriptions [2]. 9
2.4 Seven emotions and their FACS AU combinations [2]. 10

4.1 The number of AU events (EV) and total number of frames for each intensity level given in DISFA [8]. 29

6.1 The number of AU events (EV) and total number of frames for each intensity level given in DISFA [8]. 58
6.2 Number of present and absent frames for each AU in DISFA [8]. 58
6.3 Experiment 1: Resultant recognition rates for binary classification of AUs. 59
6.4 Experiment 1: Resultant recognition rates for intensity classification of AUs. 59
6.5 Confusion matrices for all AUs. 59
6.6 Results received by Mavadati et al. [8]. 61
6.7 The average 6-level intensity detection results from this study and Mavadati et al. [8]. 61
6.8 Experiment methods for pain intensity detection. 63
6.9 Confusion matrix results of six different feature extraction methods given in Table 6.8. 64
6.10 Results of binary classification. 65
6.11 Results of four level pain intensity detection. 65
6.12 Results of three level pain intensity detection. 65
6.13 Pain classification results from Bousmalis et al. [18]. Pain 1: No pain vs high pain. Pain 2: Detection of low, mediocre and high levels of pain. . . 67

Abbreviations

Abbreviation	Meaning
AAM	**A**ctive **A**ppearance **M**odels
ANN	**A**rtificial **N**neural **N**networks
NN	**N**eural **N**etworks
FFNN	**F**eed **F**orward **N**eural **N**etworks
SVM	**S**upport **V**ector **M**achines
FACS	**F**actial **A**ction **C**oding **S**ystem
FER	**F**acial **E**xpression **R**ecognition
RGB	**R**ed **G**reen **B**lue
HSV	**H**ue **S**aturation **V**alue
PSPI	**P**rkachin **S**olomon **P**ain **I**ntensity
CK	**C**ohn **K**anade Facial Expression Database
CK+	**C**ohn **K**anade Extended
JAFFE	**JA**panese **F**emale **F**acial **E**xpression Database
BU-3DFE	**B**inghamton **U**niversity **3D** **F**acial **E**xpression Database
DISFA	**D**enver **I**ntensity of **S**pontaneous **F**acial **A**ction Database
LBP	**L**ocal **B**inary **P**atterns
LBPH	**L**ocal **B**inary **P**attern **H**istogram
HOG	**H**istogram **O**f **G**radient
LPQ	**L**ocal **P**hase **Q**uantisation
PCA	**P**rincipal **C**omponent **A**nalysis
ICA	**I**ndependent **C**omponent **A**nalysis
LTP	**L**ocal **T**ernary **P**atterns
EQP	**E**longated **Q**uinary **P**atterns
RVR	**R**elevant **V**ector **R**egression

CORF	**C**onditional **O**rdinal **R**andom **F**ields
HCRF	**H**idden **C**onditional **R**andom **F**ields
iHCRF	**i**nfinite **H**idden **C**onditional **R**andom **F**ields
S-PTS	**S**imilarity-Normalized **P**oin**TS**
S-APP	**S**imilarity-Normalized **APP**earance
C-APP	**C**anonical-Normalized **APP**earance

Physical Constants

Description	Symbol		Value
Ratio of a circle's circumference to its diameter	π	=	3.14159265359...
Base of the natural logarithm	e	=	2.71828182846...

Symbols

Symbol	Description	Measurement
$P_{x,y}^{(i)}$	Cartesian coordinates of ith facial landmark point	pixels
d	Distance	pixels
β	Angle	π radians
ψ	Grid regularization constant	pixels
$E_{x,y}^{L}$	Cartesian coordinates of the left eye	pixels
$E_{x,y}^{R}$	Cartesian coordinates of the right eye	pixels
V_i	ith vector	pixels
$\mathbf{x}$	Feature vector	pixels or π radians
$Q_{D\times P}^{(i)}$	i^{t}h population in a genetic algorithm system	binary matrix
$\Gamma()$	Fitness function in a genetic algorithm system	numeric value ($\mathbb{R}$)
$\boldsymbol{\Theta}$	Total hypothesis function parameters	numeric value ($\mathbb{R}$)
θ	Specific hypothesis function parameter	numeric value ($\mathbb{R}$)
$H_{\boldsymbol{\Theta}}()$	Hypothesis function	returns $r \in (\mathbb{R})$
γ	Difference between expected and actual classification	numeric value ($\mathbb{R}$)
α	Learning rate	numeric value ($\mathbb{R}$)
κ	Regularization parameter	numeric value ($\mathbb{R}$)
$g()$	Neural network activation function	returns $r \in (\mathbb{R})$
$u_i^{(n)}$	Activation value of the nth neuron in ith layer of an ANN	numeric value ($\mathbb{R}$)
$\mathbf{F}$	Cost function of a hypothesis function	returns $r \in (\mathbb{R})$
∂	Partial derivative of a function	no type
$f_{x,y}$	Pixel value in an image at the location x, y	numeric value ($\mathbb{R}$)
$M()$	Histogram function	returns $r \in (\mathbb{R})$

Dedicated to my mom (Merih Kozbay Bingöl) and dad (Turabi Bingöl). All my achievements are simply extensions of their love and support!

Chapter 1

Introduction

1.1 Motivation

Most of the work done in facial expression recognition (FER) has focused mainly on detecting the presence of facial actions without taking the intensity factor into consideration. Even though automated FER has been the subject of many studies, thorough research has thus far not been done to study the intensity of facial actions.

The study of facial action intensity is a very promising research topic because of its relevance to a variety of future and current research areas. Its current applications include the duplication of human facial actions on a 3D human face model, estimation of the level of expressed emotion, real-time human-to-human interaction analysis, and improvement of human-computer interactions.

In this study, two aspect of automated FER are taken into account. The initial one is to determine the intensities of facial action coding system (FACS) action units (AU) [19] [17]. The second aspect is detecting the intensity of pain through facial actions using Prkachin and Solomon pain scale (PSPI) [20] [21].

In FER, many varieties of different machine learning methods have been used. These include support vector machines (SVMs), neural networks (NNs) and adaptive boosting. Most of the research done so far has focused mainly on detecting the presence of either FACS AUs or pain. Databases containing the FACS AU and PSPI intensities of images were not publicly available, therefore it was not possible to work with supervised machine learning methods. Yet Recently some datasets that are coded for FACS AU and/or PSPI intensity were made publicly available. Therefore it is now possible to work on facial action intensity detection and/or estimation using supervised machine

learning methods. So far, facial action intensity detection been addressed by only a few researchers, therefore this area remains unexploited.

1.2 Research Questions and Hypothesis

The main motivation of this research is to determine a facial action intensity (both FACS AU and PSPI level) detection method that is both efficient and robust, so the research is focused around the following three questions:

1. How can the facial feature points and facial grid be used effectively to analyze the intensities of facial actions?
2. What type of geometric relations between facial feature points and facial grid would be more efficient and successful for facial action intensity detection?
3. Is logistic regression analysis technique or feed-forward neural networks (FFNN) more effective for classifying facial actions and intensities?

It is hypothesized that an effective facial action intensity detection system can be created by extracting geometric facial features and classifying by supervised machine learning methods.

1.3 Technical Objectives

The following technical objectives need to be addressed to answer the research questions.

1. Extracting geometrical features from images efficiently.
2. Training of a logistic regression classifier for facial action intensity detection from images.
3. Training of a FFNN classifier for facial action intensity detection from images.
4. Comparison of different feature extraction and classification methods.

1.4 Research Methodology

For this research the following methodology was applied.

1. Data acquisition: The publicly available DISFA [8], and UNBC McMasters Painful Data [7] databases were acquired

2. Face detection and facial feature point extraction: The frames in the used databases were already labeled with the 66 facial feature points. The said points were used for feature extraction.

3. Normalization of facial grid: The facial feature points were rotated and re-scaled to fit $[-1, 1]$, and both vectorial and angular features were extracted.

4. Geometric feature extraction: The normalized feature points were used to extract two sets of features.

   (a) Angular features: Based on the angles between vectors derived from geometric locations of facial feature points.

   (b) Vectorial features: Based on the displacement vector of feature points from the neutral grid.

5. Classification of extracted features: Logistic regression and FFNN were implemented for classification.

This methodology is explained in detail in Chapter 5.

1.5 Accomplishments

The following accomplishments were achieved while this study was conducted.

1. Extraction of geometric features from images efficiently.

2. Designing and implementing of a logistic regression classifier for facial action intensity detection.

3. Designing and implementing of a a FFNN classifier for facial action intensity detection.

4. The conclusion that geometric features and supervised machine learning methods are efficient and robust for facial action intensity estimation was established.

5. The conclusion that FFNN is a more preferable classification method compared to logistic regression when classifying facial actions was established.

These accomplishments are explained in detail in Chapter 6.

1.6 Thesis Overview

The structure of this thesis is as follows: Chapter 2 gives a detailed introduction and discussion on FER which includes the general overview of FER, FACS, the importance of FER, and the limitations and challenges of FER.

In Chapter 3, the fundamental computational background of FER is discussed. The main discussion topics are digital image processing (DIP), computer vision, and machine learning. Chapter 4 consists of the literature survey. The chapter informs about different FER systems with machine learning and feature extraction methods as well as their comparison and combination. In Chapter 5, the methodology of the thesis is explained. Chapter 6 consists of experiments and results as well as an objective discussion regarding the results. The final chapter (Chapter 7) consists of the summary and conclusion of the thesis.

Chapter 2

Automated Facial Expression Recognition

2.1 Introduction

In this chapter, the general overview of FER is explained. Initially, brief information regarding the definition of FER is given. Then, one of the most fundamental systems used for FER which is FACS is explained. Afterwards, the importance of FER and potential areas regarding FER are discussed. Finally, the limitations and challenges regarding FER are shown.

2.2 What is Automated Facial Expression Recognition?

2.2.1 General definition

It is a widely known fact that vast majority of what one says does not come from his/her words. Analyzing the body language of others is almost instinctual to humans. Humans are extremely capable of seeing if one is sad, angry, happy etc. However for computers, the analysis of human body language is not as simple as it is for humans. Automated FER is defined as the analysis and classification of facial muscle movements. The facial movements are coded by a certain standard and via computational methods, these facial actions are classified.

2.2.2 Different coding systems

Multiple coding methods have been suggested so far to analyze the facial actions of people. Maximally Discriminative Facial Movement Coding System (MAX) [22], a system for Identifying Affect Expressions by Holistic Judgment [23], FACS [19] [17], and emotional facial action coding system (EMFACS) [24] are amongst these codings. Even though multiple facial action coding methods have been suggested so far, the commonly preferred one is FACS. Therefore, to make this study relevant and comparable to other studies, it is used in the proposed system as well. Facial action coding system is explained in detail in Section 2.3.

2.2.3 Computational methods

The discussed system is an automated system, and therefore multiple areas of computational sciences needs to be applied so as to achieve such system. One can divide computational methods for automated FER into three groups; Computer vision, digital image processing and machine learning. By definition, computer vision is the analysis of multidimensional data (such as images) to produce numerical or symbolic information. Since the research area is strictly based on coded facial actions, computer vision is fundamental for any researcher willing to exploit automated FER. The other aspect of automated FER is DIP. Digital image processing is a sub-field of signal processing that works with two dimensional signals (digital images). This computational field is very fundamental for automated FER because it is quite crucial for image normalization and feature extraction. The last part of automated FER is machine learning. Since the extracted features from digital images contain such fuzzy data, a static algorithm is impossible to devise. Therefore a statistical modeling method is crucial. All the computational methods introduced here are discussed in detail in Chapter 3.

2.3 Facial Action Coding System

2.3.1 What is Facial Action Coding System

Facial Action Coding System is a facial action analysis system that is commonly accepted amongst researchers. The system was first developed for psychological purposes rather than automated FER. It was proposed by Paul Ekman and Wallace V. Friesen in 1978 [19]. The fundamental principal of the system is to divide facial muscle areas into 11 parts (Table 2.1, Figure 2.1), and muscle movement in those parts into action units. The initial system had 44 AUs ranging from 1 to 46 (numbers 3 and 40 were not used) [19].

In 2002, a revised version of FACS was developed with additional 12 AUs involving head and eye movements [17]. Facial action coding system is one of the most commonly used method for automated FER for it is detailed, clear and robust.

Facial Area	Definition
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 peoples' 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 2.1: Face areas and definitions [2].

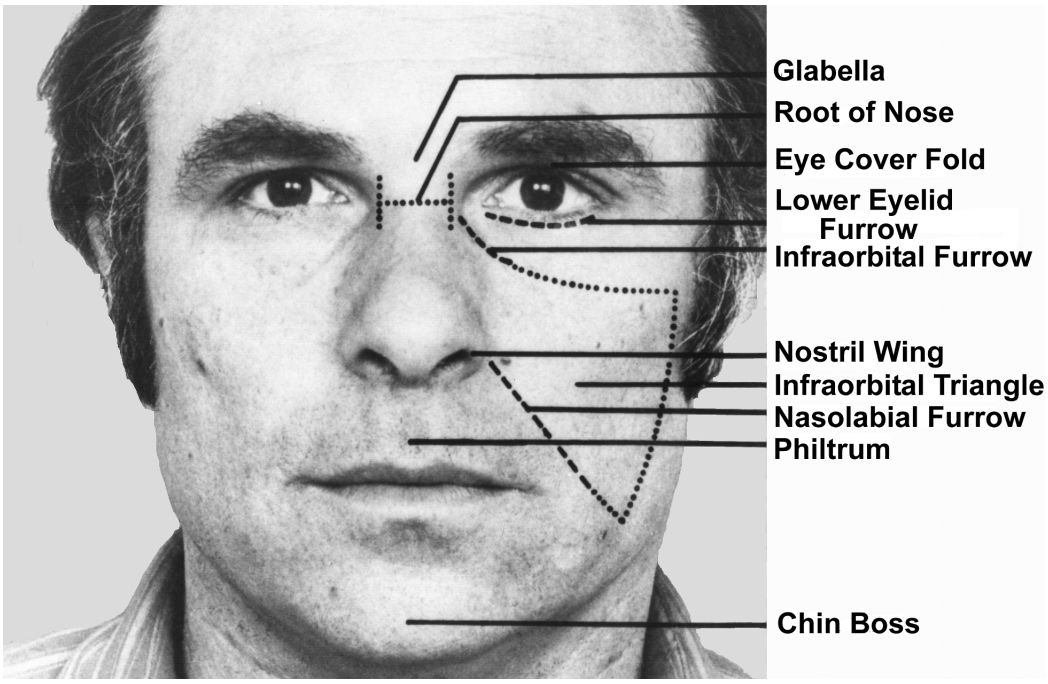


FIGURE 2.1: Facial areas defined in Table 2.1 [2].

2.3.2 Action Units and Intensities

Action units are numerical codes for the movement of face muscles. The system divides AUs into three sets; upper and lower face, and miscellaneous (head-eye movement) AUs. Action units 1-7 are accepted as upper face, 8-46 are as lower face AUs. The remaining 12 are miscellaneous AUs [2]. Table 2.2 show some of the AUs selected for this research. Figure 2.2 shows the muscular anatomy under a partition of a face and AUs related to said partition.

Analysis on human face regarding the presence of AUs is fundamental, yet detecting the presence of AUs is naturally not enough to analyze a face in detail. Therefore, AUs are not only defined as present or absent, but are defined to have a level of presence as well which is called intensity. The level definitions of AU intensities are given in Table 2.3, and Figure 2.3.

As it can be seen AU intensities are not linearly distributed. It can not be claimed that level E is five times the intensity of level A. This is quite natural for human body language is not something that can be analyzed linearly, yet it makes any study regarding FACS AU intensity more complicated.

AU	Used Muscle	Description of Muscle Movement
1	Frontalis, pars medialis	Inner corner of eyebrow raised
2	Frontalis, pars lateralis	Outer corner of eyebrow raised
4	Corrugator supercilii, Depressor supercilii	Eyebrows drawn medially and down
5	Levator palpebrae superioris	Eyes widened
6	Orbicularis oculi, pars orbitalis	Cheeks raised; eyes narrowed
9	Levator labii superioris alaeque nasi	Upper lip raised and inverted; superior part of the nasolabial furrow deepened; nostril dilated by the medial slip of the muscle
12	Zygomaticus major	Lip corners pulled up and laterally
15	Depressor anguli oris (a.k.a. Triangularis)	Corner of the mouth pulled downward and inward
17	Mentalis	Skin of chin elevated
20	Risorius w/ platysma	Lip corners pulled laterally
25	Depressor labii inferioris, or relaxation of mentalis, or orbicularis oris	Lips parted
26	Masseter; relaxed temporal and internal pterygoid	Jaw dropped

TABLE 2.2: FACS AUs selected for this research [17].

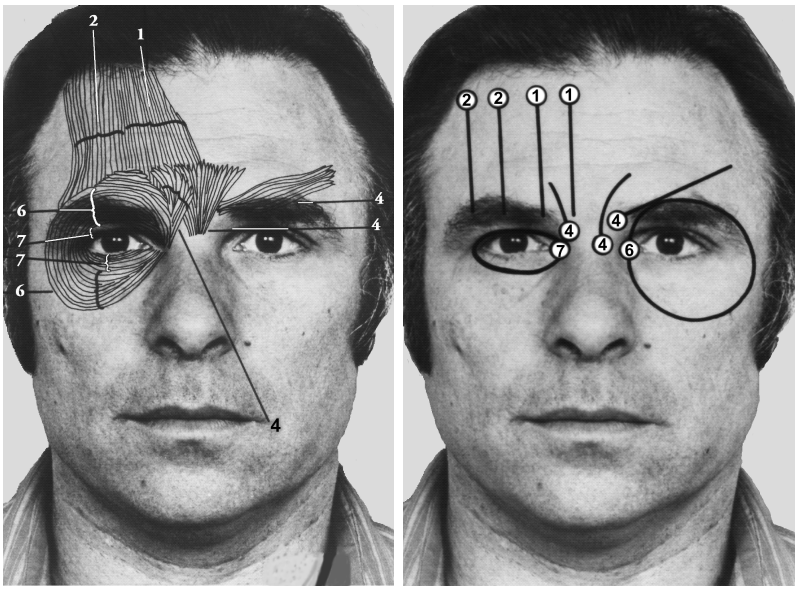
Intensity	Definition
A	Trace
B	Slight
C	Marked or Pronounced
D	Severe or Extreme
E	Maximum

TABLE 2.3: FACS AU intensity level descriptions [2].

2.3.3 Action Unit Combinations

As the main subject of FACS is to analyze human behavior through facial actions, it is natural to expect facial actions to occur in combinations. For example AU26 (jaw dropped) is impossible to occur without AU25 (lips parted). Over the years of analyzing facial actions through FACS, over 7000 AU combinations have been observed [17]. Said combinations are not only defined by having multiple AUs present, but by intensities as well.

Action units' combinations are applicable to many different areas, and one aspect of FACS to human behavior is the analysis of emotions. Emotions are defined my EMFACS [24] which is directly derived from FACS AU combinations [2]. There are seven accepted emotions which are happiness, sadness, surprise, fear, anger, disgust, and contempt. All



(A) Muscular anatomy (B) Relative AU

FIGURE 2.2: Muscular anatomy under a face region and the corresponding AUs to that region [2].

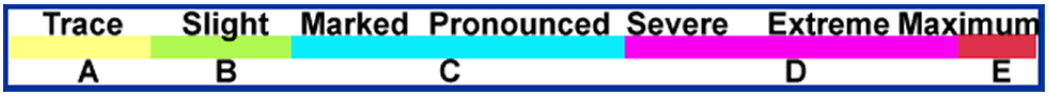


FIGURE 2.3: Visual representation of FACS AU intensity levels [2].

of these emotions are successfully definable by FACS AU combinations. Table 2.4 shows the seven emotions and their definitions by FACS AUs.

Intensity	FACS AU combination
Happiness	6+12
Sadness	1+4+15
Surprise	1+2+5B+26
Fear	1+2+4+5+7+20+26
Anger	4+5+7+23
Disgust	9+15+16
Contempt	R12A+R14A

TABLE 2.4: Seven emotions and their FACS AU combinations [2].

2.4 Importance and Potential Areas

Vast majority of what humans say does not come from their mouths. An automated facial action detection system can potentially lead up to a lot of advancements in a lot of fields. One potential application would be a system that analyzes the user's face as he/she is using an application on his/her computer and reacts accordingly. One possible

example is, a video game that changes its course as the user starts to get bored with it. The potential applications can even lead up to smart classrooms that analyzes students' body languages to give the lecturer a feedback of which students are bored or not.

One potential field would be automated pain detection (which is also done in this study) via patients' faces. The suggested application would be a breakthrough in patient-care system because even now, medical professionals are bound by under-developed pain charts. In 1992, Prkachin associated pain with multiple FACS AUs [20]. The associated AUs are brow lowering (AU4), orbital tightening (AU6 and AU7), levator contraction (AU9 and AU10) and eye closure (AU43). Furthermore, in 2002 Prkachin and Solomon developed the PSPI scale [21]. Prkachin and Solomon pain intensity scale suggests that the summation of pain related AUs' intensities would give the intensity of pain. They suggested PSPI scale by analyzing the facial images of patients with shoulder pain and currently PSPI is a commonly accepted method for pain assessment other than patients' self-diagnosis.

2.5 Limitations and Challenges

2.5.1 Facial Figure Challenge and Generalization

Since the research area is about analyzing peoples' faces, one challenge is the facial figure of an individual. Not everybody has the same skin tone, face size or facial hair. This is one of the main issues with automated FER, but can be overcome by normalization processes such as sampling to make facial area of every subject the same, or illumination normalization methods to normalize skin tone and lighting changes. These methods are quite effective for normalization yet they limit the process by their slow speed. One challenge would be to create a system that could work without normalization or to establish a method that does normalization faster.

2.5.2 Speed of the System

Speed of the system is crucial for any automated FER method for majority of the potential fields revolve around real-time applications. Normalization process is one aspect of speed challenge, but not the only one. After normalization, feature extraction and classification is done on the system which would also take time. Some methods such as Gabor filters, and SVMs have been proven to be successful for accurate recognition, yet they are slow classifiers. Support vector machines and Gabor filters are explained in Section 4.5. One aspect of this research is to create a system that could potentially

be used in real-time. Therefore logistic regression and neural networks (explained in Chapters 3 and 5) were chosen for the classification task because they are remarkably fast classifiers.

2.5.3 Limitation of Available Data

One aspect that limits this research process is the limitation of data to work with. There are some databases that are publicly available but only a few are coded for the FACS AU intensities such as Bosphorus database [6], and DISFA [8]. Since this research also focused on pain intensity estimation, it is worth mentioning that UNBC McMasters database [7] is the only painful image dataset that is publicly available [7]. All the mentioned databases are discussed in Section 4.2 .

Another limitation of available data comes from the ambiguity of FACS AU intensities. The presence and absence of AUs is doable for a certified FACS coder but different FACS coders might code same images with different AU intensities.

2.6 Summary

In this chapter, a general overview on automated FER was given. The definition automated FER, the most commonly used facial action analysis system and limitations and challenges of the field were discussed. Chapter 3 discusses the computational background of automated FER.

Chapter 3

Computational Background

3.1 Introduction

As the suggested system is directly related to multiple computational concepts such as digital image processing, computer vision and machine learning. In this chapter, said computational concepts are briefly explained and discussed.

3.2 Digital Image Processing

3.2.1 Introduction

In FER, DIP is fundamental for it would be impossible to extract any features, analyze, or normalize the data without DIP methods. In this section, some of the DIP methods that are used in FER are explained.

An image might be referred to as a two dimensional function $p = f_{x,y}$ where x and y represent the coordinates in the image and p represent the pixel value in any possible image format such as gray-scale, red-green-blue (RGB), or hue-saturation-value (HSV) [25]. Digital image processing is a type of signal processing field that deals with any image data such as static images, temporal video data, or real time video input. Image processing is used in many real world applications such as image reconstruction or noise reduction of old images, visual medical data analysis, or in FER. Image processing is one of the most critical parts of FER. In FER, convolution with image filters, histogram analysis, and image sampling are quite commonly used methods for different purposes.

3.2.2 Image Filtering and Convolution

Image filtering is one of the fundamental methods for feature extraction in FER. An image processing filter is a function that is described as $p = g(x, y)$ where p denotes the filter value in x and y coordinates [25]. In image processing, there are many filters which are used for noise reduction, edge detection, image reconstruction, blurring and edge detection. These filters are applied on image via two dimensional convolution. In FER, there are three filters that are commonly used which are Gabor filters, Haar-Like features, and Local Binary Patterns (LBP). Gabor filters, Haar-Like features, and local binary patterns are explained in Chapter 4.

Convolution is described as placing a moving image filter described as $p = g(x, y)$ on the image and computing the sum in every location [25].

Continuous Time Convolution:

$$f(x,y) * g(x,y) = \int_{\tau_1=-\infty}^{\infty} \int_{\tau_2=-\infty}^{\infty} f(\tau_1, \tau_2) g(x - \tau_1, y - \tau_2) \mathrm{d}\tau_1 \mathrm{d}\tau_2 \tag{3.1}$$

Discrete Time Convolution:

$$f[x,y] * g[x,y] = \sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} f[n_1, n_2] g[x - n_1, y - n_2] \tag{3.2}$$

The difference between discrete and continuous is that discrete values are distinct and separate values and continuous values can take any value between $-\infty$ and ∞. $f[x, y]$ represent the pixel values of the processed image in discrete time. $f(x, y)$ represent the pixel values of the processed image in continuous time. $g[x, y]$ represent the values in the applied filter in discrete time. $g(x, y)$ represent the values in the applied filter in discrete time.

Example image processing filters, and the results of image filtering with said filters are given in Figure 3.1, and Figure 3.2.

$$\begin{bmatrix} \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \end{bmatrix} \qquad \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

(A) Blurring (B) Edge Detection

FIGURE 3.1: Example filters for image processing.



(A) Original Image (B) Blurred Image (C) Edge Detected Image

FIGURE 3.2: Original and convolved images with blurring and edge detection filters.

3.2.3 Histogram

Histograms are graphic representations of the pixel value distributions in images [25]. The histogram of an image is generated by calculating the amount of pixels in an image that falls into each bin in the histogram. This method is applied to see the color distribution in an image. Histogram analysis is a very robust method for analyzing the given image for noise detection and/or determining if there is a need for image reconstruction. The following equation shows how to compute the histogram of an image, and Figure 3.3 is an example of histogram computation.

$$M(i) = \sum_{x} \sum_{y} f_{x,y} \forall f(x,y) = i \tag{3.3}$$

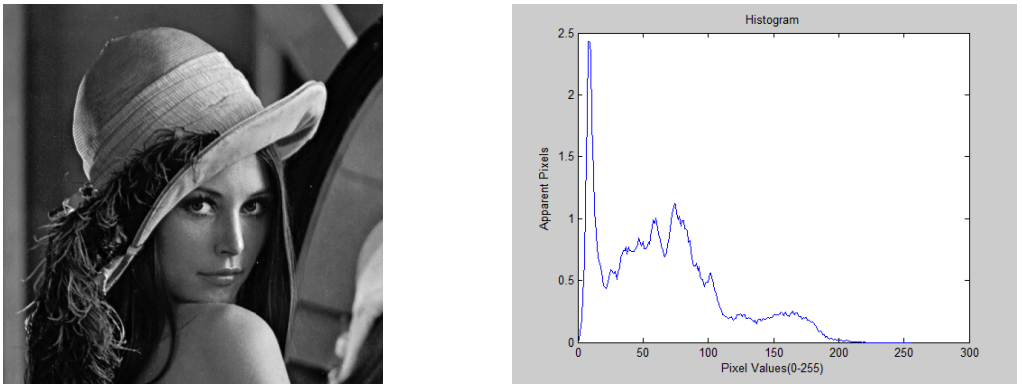


Image Histogram

FIGURE 3.3: A wild gray-scale image and its histogram.

3.2.4 Sampling

In DIP, sampling is the method of changing the dimensions of the image by a ratio and is divided into three groups; sub-sampling, re-sampling, and fractional-sampling [25]. Sub-Sampling in DIP is downscaling an image by the ratio μ by selecting every μ^{th} pixel in the image, re-sampling is up-scaling an image by the ratio μ by repeating every pixel $\mu * \mu$ times, and sampling an image by a fraction $\frac{x}{y}$ is re-sampling the image by x and sub-sampling the result by y. Sampling is mainly used to get the images and/or extracted faces to a standardized size for data normalization purposes. Algorithm 1 and, Algorithm 2 show the re-sampling, and sub-sampling algorithms respectively.

Algorithm 1 Image Re-Sampling Algorithm

Input: I: Input Image, I_{ij}: The pixel value of I in position $[i, j]$, m and n: Total rows and columns in I, μ: Re-Sampling ratio
Output: A: Re-Sampled I by the ratio μ

for $x = 1$, $i = 1$; $i \leq m * \mu$; $i+ = \mu$, $x++$ **do**
  for $y = 1$, $j = 1$; $j \leq n * \mu$; $j+ = \mu$, $y++$ **do**
    for $k = 1$; $k \leq i + \mu - 1$; $k++$ **do**
      for $l = 1$; $k \leq j + \mu - 1$; $l++$ **do**
        $A_{kl} = I_{xy}$
      end for
    end for
  end for
end for
return A

Algorithm 2 Image Sub-Sampling Algorithm

Input: I: Input Image, I_{ij}: The pixel value of I in position $[i, j]$, m and n: Total rows and columns in I, μ: Sub-Sampling ratio
Output: A: Sub-Sampled I by the ratio μ

for $x = 1$, $i = 1$; $i \leq m$; $i+ = \mu$, $x++$ **do**
  for $y = 1$, $j = 1$; $j \leq n$; $j+ = \mu$, $y++$ **do**
    $A_{xy} = I_{ij}$
  end for
end for
return A

3.3 Computer Vision

3.3.1 Introduction

Computer vision is a computational field that is based on analyzing multidimensional data to provide numeric or symbolic information. There are multiple methods in computer vision that applies to automated FER. It is most commonly used for face detection, image normalization and feature extraction. For automated FER purposes, computer vision combines image processing techniques and machine learning methods to detect faces or face regions (such as nose or eyes) and extract features. In this section, some relative parts of computer vision are briefly explained. More detailed information regarding relative computer vision methods are given in Chapters 4 and 5.

3.3.2 Face Detection

There are multiple classification methods for face detection in computer vision but the most efficient one is the Viola-Jones object detection technique [26]. Viola-Jones object detection framework is applicable to detecting any object in a frame, but it is most commonly used for real-time face detection. The method uses Haar-Like features (as mentioned in Section 4.5.2) to derive averaged pixel values of the analyzed portion of the frame and classifies it as either face or not. This method is also applicable to detect other facial figures other than faces such as eyes or mouth location. The most important property of Viola-Jones object detection framework for automated FER is that it is much faster than any other method suggested which makes it applicable to real-time applications. This method is explained throughly in Section 4.3.

3.3.3 Feature Extraction

Another part of computer vision that is directly applicable to automated FER is feature extraction. In any machine learning method, features are mathematical representations of any type of data that are used for classification. One of the most fundamental method in computer vision that is applied in automated FER is active appearance models (AAM). The method is based on using a set of training data labeled with the landmark points, and then matching the derived statistical model of object shape and appearance to a new image. In the case of automated FER, the mentioned shape is a facial grid that consists of 66 landmark points on the face including outer face grid, eye, nose and mouth landmarks. This method has been applied to multiple applications which are explained in Section 4.4.1.

3.3.4 Object Tracking

Another important method in computer vision that is directly applicable to automated FER is object tracking. In many cases, face detection from frame to frame is not computationally inefficient for Viola-Jones object detection framework works very fast, yet the same is not applicable to facial grid and landmarks. It would be very inefficient and slow to detect the AAM grid in every frame of a stream. Because of that, some effective algorithms such as Scale Invariant Feature Transform (SIFT) [27] or Speeded Up Robust Features (SURF) [28] has been implemented. Both SIFT and SURF are feature extraction methods that are quite robust, also they are quite applicable to feature point tracking because of their speed.

SIFT [27] initially detects pixel extremas in the first image using difference of Gaussians of the image. Then uses the extracted extremas to a voting system based on Hough transform voting to detect the most similar points in the second image.

SURF is based on determinant of Hessian blob detectors for point detection via features extracted by Haar wavelets.

3.4 Supervised Machine Learning

3.4.1 What is Machine learning, and Supervised Machine Learning?

In FER, as in many other areas, machine learning is essential since without machine learning, it would be difficult to work with vast amounts of fuzzy data. In FER, a variety of machine learning methods have been used to classify extracted data. Machine learning is divided into three groups: supervised, unsupervised and semi-supervised learning methods.

Supervised machine learning is a type of machine learning where each sample in the dataset is labeled. The classifier uses a training set to learn a set of parameters and tries to classify the testing set successfully using the learned parameters. Unsupervised machine learning methods try to find hidden structures in an unlabeled data. Semi-supervised machine learning uses both labeled and unlabeled data. A commonly conducted practice in semi-supervised machine learning is to use small amount of labeled and large amount of unlabeled data for training, and large amount of labeled and small amount of unlabeled data for testing.

In FER the most commonly used machine learning methods are supervised learning methods such as; SVMs, NNs, adaptive boosting or AdaBoost, and regression analysis.

All supervised machine learning methods use a training data to train a hypothesis function for future predictions $h(\mathbf{\Theta})$. Training data is defined as such:

$$S = \{(X_i, Y_i) | \forall i \in \{1, ..., N\}\} \tag{3.4}$$

Where, S is the training dataset, X_i are the extracted features, Y_i is the classification of the i^{th} member of the training data and N is the number of subjects for training the hypothesis function $h(\mathbf{\Theta})$, where $\mathbf{\Theta} = \{[\mathbf{\Theta}_i] \forall i \in \{1, ..., N\}\}$ for future predictions.

3.4.2 Support Vector Machines

A support vector machine is a non-probabilistic binary linear classifier. The main idea behind SVMs is the creation of distinct borders between partitions of given data, in order to break the data into multiple sections that could be used for classification purposes with the future input [29]. Support vector machines are trained to produce a function that can predict the classification of the future data. Support vector machines are more margin optimization models that can classify non-linear data using hyperplanes, rather than greedy output search systems. They use the dataset S to train the hypothesis function, $h(\theta)$, for future predictions. However, their methodology is a bit different from other methods since SVMs are used to classify data between already known clusters. Support vector machines initially determines the support vectors which are the border elements of a cluster. Then a hyperplane equation is derived using these support vectors. The following equation is solved for the hyperplane parameters.

$$X_s * W - b = \{Y_s | \forall X, W, Y\} \tag{3.5}$$

Where, W is the set of normal vectors that is defined as $W = \{W_i(x, y) | \forall i \in \{1, ..., t\}\}$, X_s is the s^{th} support vector, b is the constant in the hyperplane equation, Y is the solution set for the equation.

Figure 3.4 illustrates the use of an SVM in a classification task.

3.4.3 AdaBoost

Adaptive boosting, or AdaBoost is another supervised machine learning method often used in FER. AdaBoost is an optimization method used for classification purposes [30]. It uses weakly classified data more frequently, in order to better classify such data. It avoids the already successfully classified data, since this would result in a waste of

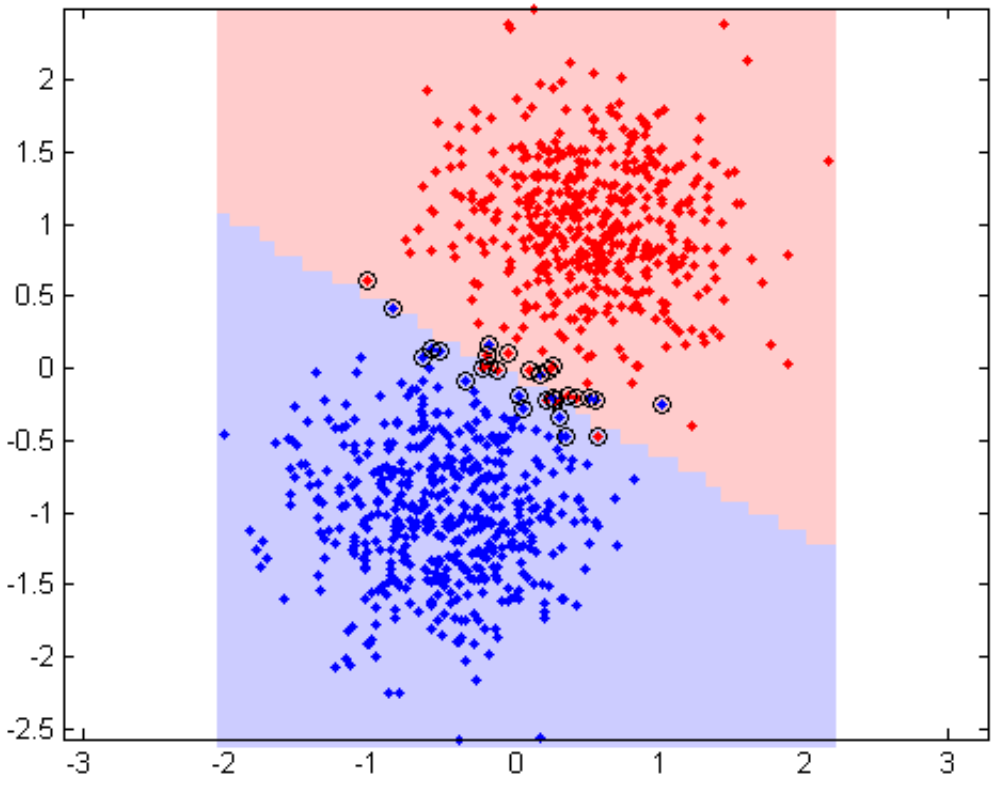


FIGURE 3.4: An example of a linear classification border trained by SVM . Blue and red dots represent different clusters, and circled dots are support vectors.

processing power. In some cases, the AdaBoost algorithm might not achieve a higher success rate. However, compared to most of the other machine learning algorithms, it is much faster at classifying data and less susceptible to over-fitting.

AdaBoost takes the training set S, $L_i \forall i \in \{1, ..., N\}$ (where L_i is the i^{th} weak classifier in N amount of classifiers), error function $E(L, y, x)$ (L is the weak classifier function, y is the expected result and x is the classified subject), and weights $W_i \forall i \in \{1, ..., N\}$. Then in every iteration of the learning process, it finds the weak learner in L that minimizes the weighted sum error ϵ ($\epsilon = \sum_N \{W_i E(L_i)\}$). After the weak classifier is found, an α value is computed ($\alpha = \frac{1}{2} \ln \frac{1-\epsilon}{\epsilon}$). In the last process, all the weights $W_i \forall i \in \{1, ..., N\}$ are updated through the error function ($W_i \leftarrow W_i E(L, y, xW_i\epsilon)$). Finally all the weights are renormalized to make sure $\sum_N W_i = 1$.

There are multiple different versions of AdaBoost such as Real AdaBoost, LogitBoost, Gentle AdaBoost. Real AdaBoost is a probability estimation method that focuses on determining if the analyzed sample is in the positive class [31]. LogitBoost is the application of an established logistic regression on AdaBoost [31]. Gentle AdaBoost is focused on making the AdaBoost algorithm more efficient by taking the error function as mean squared error ($E(L, y, x) = (y - L(x))^2$) and not applying any other coefficient [32].

3.4.4 Genetic Algorithms

Genetic algorithms are derived from the evolutionary process of living organisms [33]. The main idea is to generate a random population, then evaluate the fitness of every subject in the population by a fitness function. The last process is to de-population via death by lack of fitness, and re-population by crossover between parents, mutation of the offspring, and/or immigration. These methods are most commonly used for optimization and search problems. Genetic algorithms are also very effective for feature selection.

3.4.5 Gradient Descent

Gradient descent is an optimization algorithm to find the global minima in any given search space [34]. It is a very commonly used method to minimize the cost function $\boldsymbol{J}(\boldsymbol{\Theta})$ for any $H_{\boldsymbol{\Theta}}$ in any statistical modeling function. The main idea in gradient descent is to derive the most successful path of $\theta \in \boldsymbol{\Theta}$ values, and directing the θ update rule to the detected direction. Therefore, the equation for gradient descent is:

$$\theta_k \leftarrow \theta_k - \alpha \frac{\partial}{\partial \theta_k} \boldsymbol{J}(\boldsymbol{\Theta}) \tag{3.6}$$

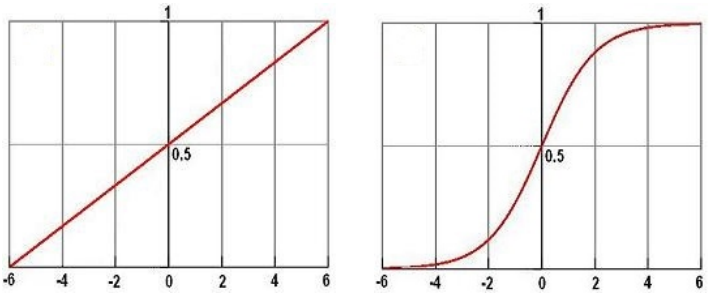
Where α stands for learning rate.

Gradient descent is one of the most commonly used $\boldsymbol{\Theta}$ update methods for regression analysis and neural networks.

3.4.6 Regression Analysis

Regression analysis is a statistical process for estimating relations between different parts of data. There are different types of Regression Analysis such as the linear approach, the logistic approach, and the ordinary least squares approach. Regression analysis trains a hypothesis function $h(\boldsymbol{\Theta})$ that predicts the probability of Y given X; that is, $h(\boldsymbol{\Theta}, X) = P(Y|X)$. Linear regression is the most basic regression analysis approach. It defines $h(\boldsymbol{\Theta}, X) = \boldsymbol{\Theta}^T X * \epsilon$ [35]. Logistic regression uses the training data S, then trains either a convex or concave estimation function of the form $h(\theta, X) = \frac{1}{1+e^{(\boldsymbol{\Theta}_0 + \boldsymbol{\Theta}_{i+1} * X_i)}}$. It then compares the result to Y_i, and uses the success of the estimation to update the parameters of the estimation function [36].

The linear regression and logistic regression analysis curves are shown in Figure 3.5.



(A) Linear Regression (B) Logistic Regression

FIGURE 3.5: Linear and Logistic Regression Analysis comparison.

3.4.7 Neural Networks

Neural networks is another supervised machine learning technique that is used in FER. A typical neural network consists of three parts; an input layer, one or more hidden layers, and an output layer. Neural networks are different from the commonly-used computation methods for the processing and memory components are not separated. In this regard, they resemble an actual biological neural network, which is where neural networks get their name from [37]. Neural networks compute the output of a system by obtaining data from the input layer, processing it in the hidden layer(s) and returning the output in the output layer. Neural networks gets the training data S, and $U_i \forall i \in \{1, ..., n\}$ where U_i is the i^th layer in the network and maps the activation of every node in a layer to the next layer until the output layer is reached $X_j \rightarrow U_0 \rightarrow U_1 \rightarrow ... \rightarrow U_i \rightarrow U_{i+1} \rightarrow ... \rightarrow U_n \rightarrow Y_j$ where $U_i = h_i(\mathbf{\Theta}) \forall i \in \{1, ..., n\}$and finally checks the result with the expected result, then updates θ until an error margin is reached. The basic structure of a neural network is shown in Figure 3.6.

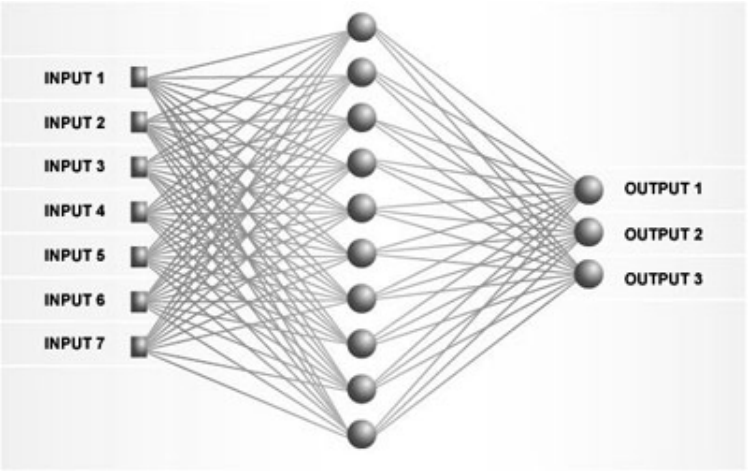


FIGURE 3.6: Basic structure of Neural Networks.

3.5 Summary

In this section, the fundamental computational background for automated FER is discussed. Image processing techniques, computer vision and its applications to automated FER, and finally supervised machine learning techniques are discussed. In the next chapter (Chapter 4), the literature survey related to automated FER is discussed.

Chapter 4

Literature Survey

4.1 Introduction

Facial expressions and their meanings have always been a matter of question to humans ever since humans have evolved into civilized beings but it was Charles Darwin [38] who was the first one to research the expressions and emotions of both human beings and animals alike. Since then, there has been a lot of research done on recognizing facial expressions and how to interpret them. Facial expressions have been analyzed and researched by psychologists, computer scientists, engineers and many any other researchers from different fields who are interested in the mystery of expression of emotions in human beings. In this chapter, some of the studies conducted so far are explained and discussed.

Majority of studies in FER revolve around two types of approaches. Geometry based and appearance based approaches. Geometry based methods mainly focus on detecting the fiducial points in the face, or analyzing the 3D representation of the face as facial features, and using the geometric relations between said features for expression recognition. The appearance based approach mainly focuses on applying digital image processing filters on images, and analyzing the texture information, or pixel values on relative face regions, to recognize expressions.

Majority of the studies conducted so far on automated FER follow this pattern:

1. Data acquisition.
2. Face detection and extraction.
3. Feature extraction.
4. Classification of extracted data.

4.2 Databases and Data Retrieval

In the field of FER, vast majority of the work done so far has been done with machine learning methods. For any machine learning method to be applicable to any situation, data for both training and testing is necessary. In this section, publicly available datasets for automated FER are presented.

4.2.1 The Japanese Female Facial Expression Database

The Japanese Female Facial Expression Database (JAFFE) is a database that is publicly available for research purposes that was published by Lyons et al in 1997 [3]. The dataset consists of ten subjects posing for three or four frontal facial picture per each subject that makes a total sum of 219 images. There have been a couple of studies conducted using this database. Figure 4.1 shows some of the images taken from JAFFE database.



FIGURE 4.1: Images from JAFFE Database [3].

4.2.2 Cohn Kanade Database

The Cohn Kanade Facial Image Database (CK) is a database that was released by Cohn et al. in the year 2000 [39]. This database consists of 2105 images taken from 186 adults by cameras that records 30 frames per second. The subjects of this database are between the ages of 18-50 and the database consists of 69% females, and 31% males. The statistical distribution of ethnicities in CK are as follows: 81% Euro-Americans, 13% Afro-Americans and 6% other ethnicities. The images in the database are coded for the presence of AUs, fiducial points of the face, and the emotion shown in the image but the AU intensities in every image is not coded. The only thing known in the database related to FACS AU intensity is that initial image in any image sequence has 0 AU intensity and the last image either has E intensity or if the intensity is lower than E, the final intensity is coded. Therefore the dataset has sequential coding with appex property. Even though CK was a very expansive database, it was argued that expression coding

was inconsistent and the dataset was cluttered with non-posed visual data. Therefore Lucey et al. [4] released the extended version of CK which is referred to as CK+ that has 22% more sequences and 27% more candidates. CK+ also had non-posed smiles as well as more consistent and correct emotion coding. Another extension to CK was made by RPI ISL research group [40] who re-coded CK database frame by frame by the scale AU existent, non-existent and existent with low intensity which is called CK enhanced dataset. Figure 4.2 shows the first and last images of one video sequence from CK+.



(A) First image (B) Last image

FIGURE 4.2: First and last images of sequence 1 of candidate 52 in CK+ [4].

4.2.3 BU-3DFE

The very first attempt to generate a publicly available 3D FER database was made by Yin et al. [5] which is Binghamton University 3D Facial Expression Database (BU-3DFE). BU-3DFE dataset is a 3D face image database that consists of 100 subjects which are 60% female and 40% male with varying ethnic backgrounds. The images in the database are displaying six basic emotions (Sadness, Anger, Contempt, Disgust, Surprise, Fear) in four intensity levels. The database also has AU 12 (Lip corners pulled up) and AU14 (Lip corners tightened) coded for their FACS AU intensities. This is one of the most commonly used database for the newly developing 3D FER studies for its diversity, 3D image detail and AU intensity coding. Example 2D and 3D images fro BU-3DFE are given in Figure 4.3.

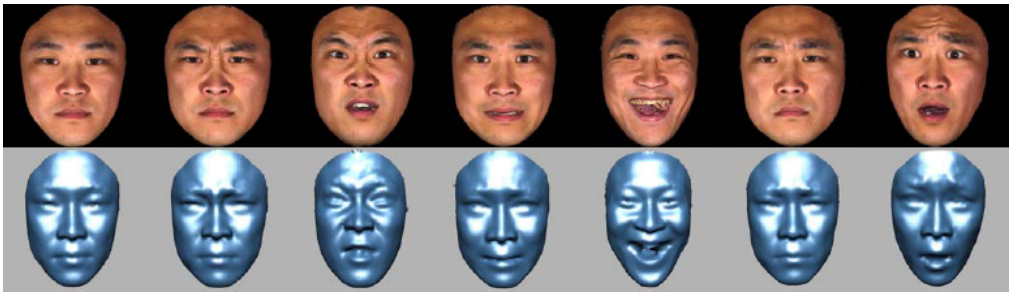


FIGURE 4.3: Example 2D and 3D images from BU-3DFE [5].

4.2.4 Bosphorus Database

Bosphorus database is another 3D FER database that was developed by Savran et al. [6] for 3D FER research purposes in 2008. This database was developed using images from 105 adult candidates some of whom are professional actors/actresses. Professional actors/actresses were selected as subjects to generate a more realistic expression database. The images in this database (both 2D and 3D) are coded for 20 lower face AUs, 5 upper face AUS, and 3 AU combinations and their intensities. The database is coded for six basic emotions and 24 fiducial facial feature points as well. This database also contains 3D images that has occlusions such as hands or glasses. In Figure 4.4, some occluded images from the Bosphorus database are shown.

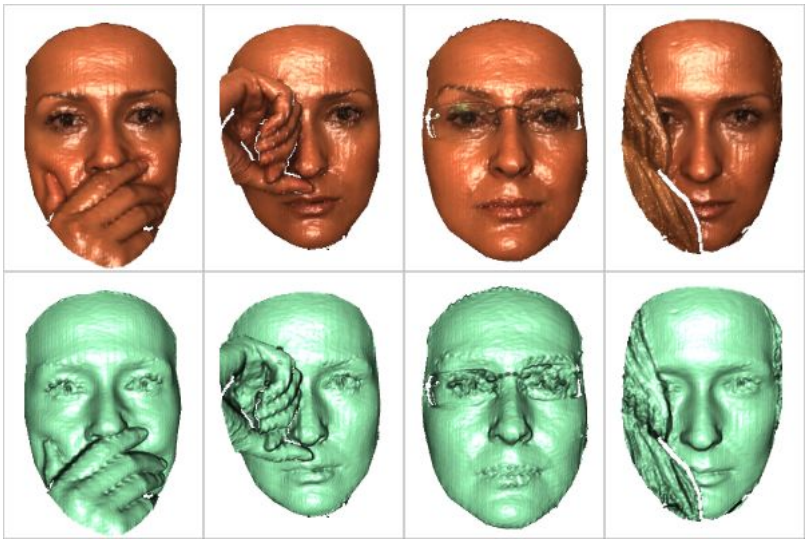


FIGURE 4.4: Occluded 3D images from Bosphorus Database [6].

4.2.5 UNBC McMasters Painful Dataset

UNBC McMasters database is another dataset that is used for automated FER [7]. This database consists of images from video streams taken from 25 subject with shoulder injuries. The database has 48,398 frames from 200 video streams with spontaneous facial expressions. The database is coded with PSPI [21], self-report pain intensity and observer determined pain intensity. As mentioned in the Section 2.4, PSPI scale is directly related to intensities of AU4, AU6, AU7, AU9, AU10 and AU43. Therefore, each frame is also coded for the intensities relative AUs as well. The painful images were taken while the subjects were activating their injured and unharmed arms to make sure the pain expressions were genuine. Figure 4.5 show some of the sequential frames in the dataset.



FIGURE 4.5: Some sequence of images from the UNBC-McMaster Database [7] with their PSPI pain scale labels.

4.2.6 DISFA

Denver Intensity of Spontaneous Facial Action Database (DISFA) [8] is the newest publicly available database in the field of FER. This database consists of video frames taken from 27 adults (12 female and 15 male) with varying ethnicities while watching stimulating videos, therefore this database is a non-posed facial action database. Every candidate has at least 4845 coded frames. One of the most important attribute of DISFA is that the database has 12 AUs coded for their intensities (AUs 1,2,4,5,6,9,12,15,17,20,25,26) by two certified FACS coders and therefore crucial for researchers who are interested in AU intensity detection. Some sections of images with their AU coding from DISFA is given in Figure 4.6. Also the distribution of AUs and intensities in the whole dataset are given in Table 4.1.

4.3 Face Detection and Extraction

For FER, one of the most important parts of data analysis is face detection and extraction. Many studies regarding face detection has been conducted so far and remarkable progress has been made. In this section some of the applied face detection methods are discussed.

In 2001, Viola and Jones devised a method for image detection using Haar-Like Features [26]. Their method consisted of convolving Haar-Like-Features on the image and using AdaBoost for classifying the convolution results. Their research showed that convolving Haar-Like features with images is both quick and an efficient way for face detection.

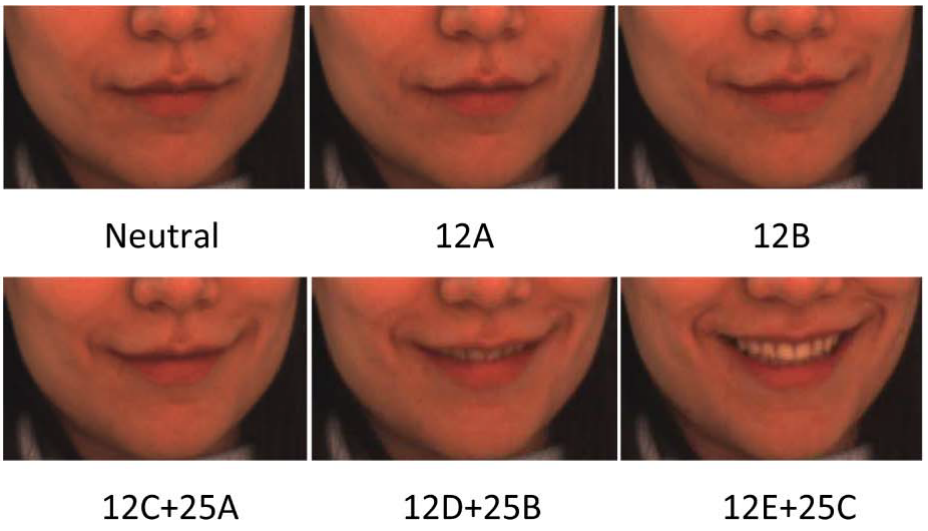


FIGURE 4.6: Some sections of images and their AU intensity coding from DISFA [8].

	AU Info		# Frames for each AU Intensity					
	AU#	#EV	0	1	2	3	4	5
Upper-Face	1	161	122036	2272	1749	2809	1393	555
	2	111	123450	1720	934	3505	836	369
	4	238	106220	4661	7636	6586	4328	1383
	5	100	128085	1579	719	293	104	34
	6	170	111330	9157	5986	3599	601	141
Lower-Face	9	73	123682	1659	2035	3045	316	77
	12	250	100020	13943	6869	7233	2577	172
	15	97	122952	5180	1618	1017	47	0
	17	271	107884	6342	4184	2281	112	11
	20	99	126282	1591	1608	1305	28	0
	25	296	84762	9805	13935	15693	5580	1039
	26	321	105838	13443	7473	3529	314	217

TABLE 4.1: The number of AU events (EV) and total number of frames for each intensity level given in DISFA [8].

An improvement was made by Lienhart and Maydt in 2002 [41] which included rotated Haar-Like Features as well for tilted face detection as well as for the reduction of false positives. Yet the improvement made by Mita et al. [42] was the most significant. They used joint Haar-Like features and an extended AdaBoost method for face detection. Method proposed by Mita et al. reduced the error rate by 37% compared to previous work and it is significantly faster than the previously published methods. In 2009, Geetha et al. [43] proposed a method for real time face tracking. Geetha et al. [43] used the motion information between two consecutive frames for facial contour extraction and used the eye size and location for face size estimation. Most recently, another research was conducted by Zhao et al. [44] that focused on facial landmark detection. Zhao et al. [44] also suggested a robust face detection algorithm that uses traditional

Haar-Like features and Non-Adjacent Haar-Like features. Therefore the system both looks up global and local texture informations. Their system is proven to outperform any other system so far when detecting faces, but using two distinct types of Haar-Like features makes the system very slow. Haar-Like features are explained in more detail in Section 4.5.2.

4.4 Geometry Based FER Methods

One of the two distinct type of automated FER methods is geometry based FER. Geometry based models are mainly focused on either fiducial point detection or estimation using AAMs, or 3D model fitting and/or face reconstruction methods. In this section, studies by both fiducial point related and 3D related geometry based FER will be explained.

4.4.1 Active Appearance Models

In FER, one of the main approaches is AAMs, which focuses on detection of fiducial points in human faces and uses the geometrical relations between those points for facial feature extraction and expression recognition. In this section, recent studies involving both AAM features for FER and improvements made on AAM fitting are discussed.

One research was done by Cristinacce and Cootes [45] that used AAMs for generating feature templates for fiducial point detection rather than detecting actual fiducial points. This approach was taken in order to create an object class that would detect the feature points in a new instance, but AAMs were not their main usage for facial feature point detection. It was Saragih and Gocke in 2007 [46] that did a comprehensive research to update AAM geometric update functions. Due to the use of simple update models, previous researches on AAM fitting were bottlenecked. In order to tackle that bottleneck, Saragih and Gocke used AAMs with a non-linear update model for point location updates and showed that non-linear update models are better choices for model fitting approaches. Another research was also done by Saragih and Gocke in 2009 [47] that also focused on linear update models' inaccuracy in AAMs. They did not try to replace the linear fitting model, but they simulated the update of facial grid multiple times for linear fitting model optimization. Their results show that linear fitting models are capable of learning the optimal fitting patterns by simulations with more iterations, and are also more than capable of providing superior performance. An improved research on fiducial point detection and tracking for FER was conducted by Tsalakanidou and Malassiotis in 2010 [48] for real time fiducial point related FER. They used active

appearance models for tracking and extracting facial data from the real time input. Their methodology consisted of local detectors for different parts of the face. They did not only use the observable edges in human faces for this, but also used the observable depth in images using a 3D spectrum camera for fiducial point detection as well. They applied some static rules for FACS AU detection and tested their system on real time candidates as well as a large database that they assembled themselves. Tsalakanidou and Malassiotis [48] also showed that their proposed system is much more resistant to lighting features and facial deformations such as wrinkling. In 2010, Valstar et al. [49] conducted an experiment to increase the efficiency of AAMs using SVMs and Markov Models. Their research was based on reducing the search space for fiducial point detection so as to reduce the time it would take the system to detect facial feature points. Their research opened the doors for real-time facial feature detection and AAM based FER due to its speed. Another improvement on fiducial point detection was done by Zhu and Ramanan [50]. Zhu and Ramanan proposed a system that does not only work with training and testing data from a specific database, but also can work with random images that were taken from real-life situations. Their system consisted of a tree network with multiple selection partitions. Their experiments showed that their method is more than capable of capturing global aesthetic deformations and much easier to optimize. One of the latest published research that included fiducial point detection was done by Tzimiropoulos et al. [9] in 2013. Tzimiropoulos et al. suggested an improved version of AAMs that are called Active Orientation Models (AOM). Active orientation models used different statistical models than AAMs, was accompanied by more accurate facial parameter estimation methods, and was more prone to generalization of different face parameters. They not only tested their results on database inputs, but also on random images from the internet as well and received remarkable facial action point detection results. They also gave the link to their source code for testing and research purposes. Figure 4.7 shows two wild images that Tzimiropoulos et al. have experimented on.

4.4.2 3D Analysis

In FER, some research have been done on 3D facial expression analysis using extracted features from human faces as well, and some of the research is done on 3D human face shape analysis.

In 2008, Tang and Huang [51] proposed a method for automatic facial point detection in 3D images. They applied statistical models for discriminating between possible facial action points in reconstructed 3D human faces. They also used facial action points for AdaBoost training and then tested their data on BU-3DFE database and received an average of 95.1% recognition rate in detecting six basic emotions. A similar approach

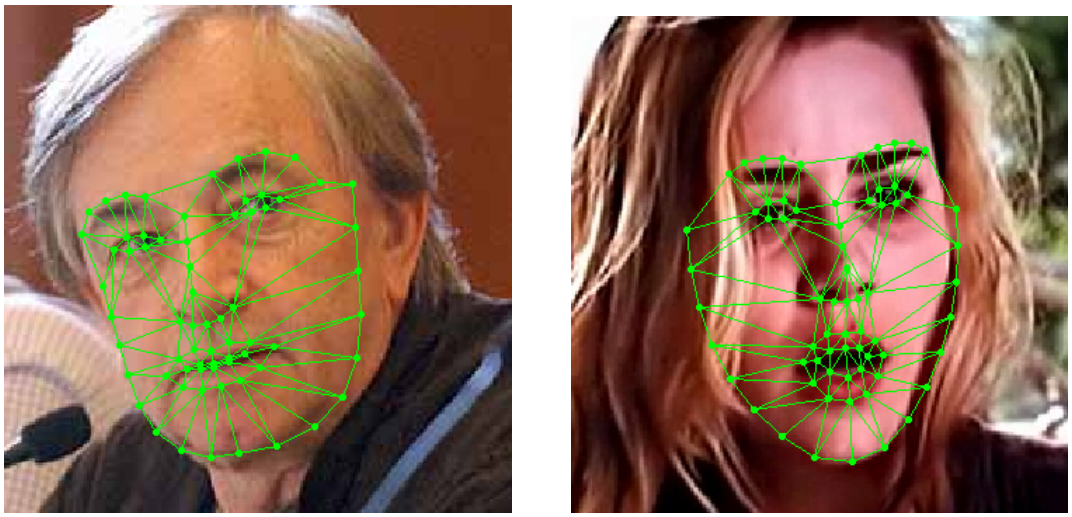


FIGURE 4.7: Two wild images that Tzimiropoulos et al. have experimented on [9].

was done by Maalej et al. [10] on 3D model analysis for FER. They conducted a system in which certain facial areas were extracted from 3D human face models and shape analysis was conducted on the extracted areas for the analysis of 3D deformations. Figure 4.8 shows the extracted patches from human faces. They used SVMs and AdaBoost for analyzing optimal facial feature deformation so as to classify between six basic emotions using BU-3DFE database. Their research showed that facial area patching on 3D models is not only possible for FER, but quite successful as well. In 2011, Maalej et al. [52] published a further research on their previous approach that also included non-frontal views and multiple patches colliding with each other in a human face. Their final method was also tested on BU-3DFE database and achieved 98.81% recognition success when detecting emotions.

A different type of approach was made by Mpiperis et al. [53]. They used facial grids in the face to reconstruct the facial image in a 3D environment and used the BU-3DFE database for both training and testing. Mpiperis et al. [53] showed that face reconstruction is also a viable research area for 3D FER. A similar research was done by Li et al. [54] that also used sparse features in the face for face reconstruction in 3D meshes. Li et al. [54] also showed that when using features that are more relevant for expression recognition, FER can be done without collecting exhaustive data from each face. In 2010, Segundo et al. [55] conducted another research that focused on facial feature point detection via 3D FER. They used the extracted depth, edge information in human faces, and the properties of basic curvature models for 3D face reconstruction. Segundo et al. [55] received remarkable results when detecting 3D facial feature points as they received above 99% recognition success with their study.

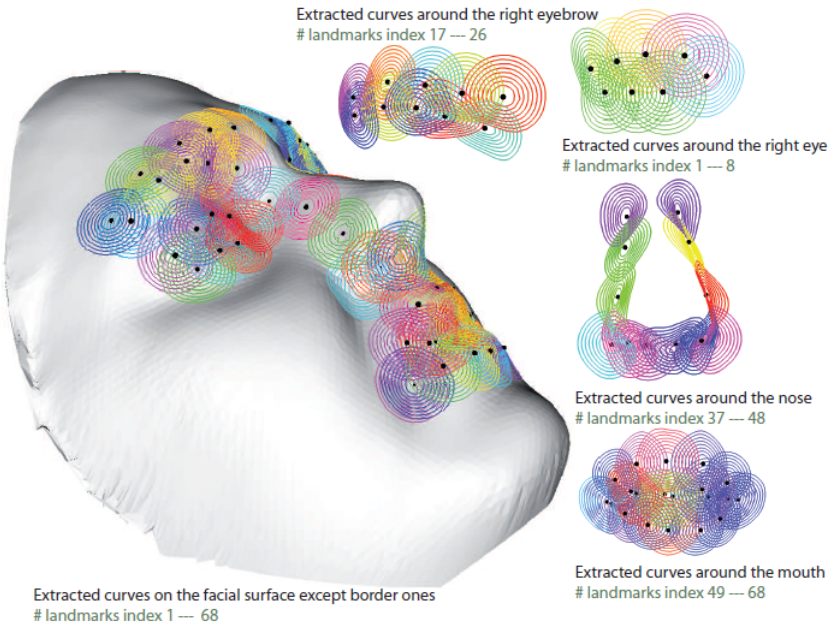


FIGURE 4.8: Curves extracted by Maalej et al. [10].

4.5 Appearance Based FER Methods

The other commonly used approach for FER is appearance based FER. In appearance based FER, the main focus is approaching images and/or video frames as 2D signals, and applying signal filters on those images, then finally observing the pixel values for facial expression recognition purposes. Many researchers studied facial expressions using appearance based models. The more relative and more recent studies involving appearance based FER are discussed in this section.

4.5.1 Gabor Filters

Gabor filters are image processing filters that are used for edge detection. The main application of Gabor filters in FER, is to convolve Gabor filter on either the whole image, or on facial feature areas, get the combined response from the image which is called Gabor jets, and use this response for expression analysis.

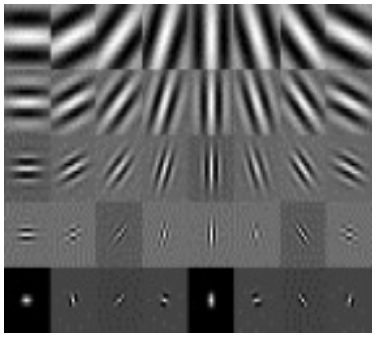
Gabor filters are represented as such:

$$\psi(x, y, \omega, \theta) = \frac{1}{2\pi\sigma^2} e^{\frac{-(x'^2+y'^2)}{2\sigma^2}} [e^{i\omega x'} - e^{\frac{-\omega^2\sigma^2}{2}}] \tag{4.1}$$

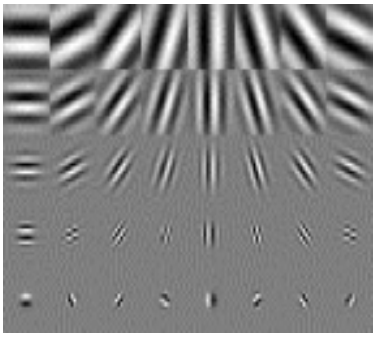
$$x' = x\cos\theta + y\sin\theta \tag{4.2}$$

$$y' = -sin\theta + y\cos\theta \tag{4.3}$$

In this equation, x and y represents the coordinates of the pixel value in spatial domain, ω represents the radial center frequency, θ represents the orientation of the Gabor filter, and the σ is the standard deviation of the round Gaussian function along the x-y axis. Imaginary and real parts of Gabor wavelets with 8 different orientations, and 5 different frequencies are given in Figure 4.9. The Gabor response of two images with different Gabor wavelets are given in Figure 4.10.



Real Parts of Gabor Filters



Imaginary Parts of Gabor Filters

FIGURE 4.9: Real and Imaginary parts of Gabor filters created with 5 frequencies and 8 orientations.

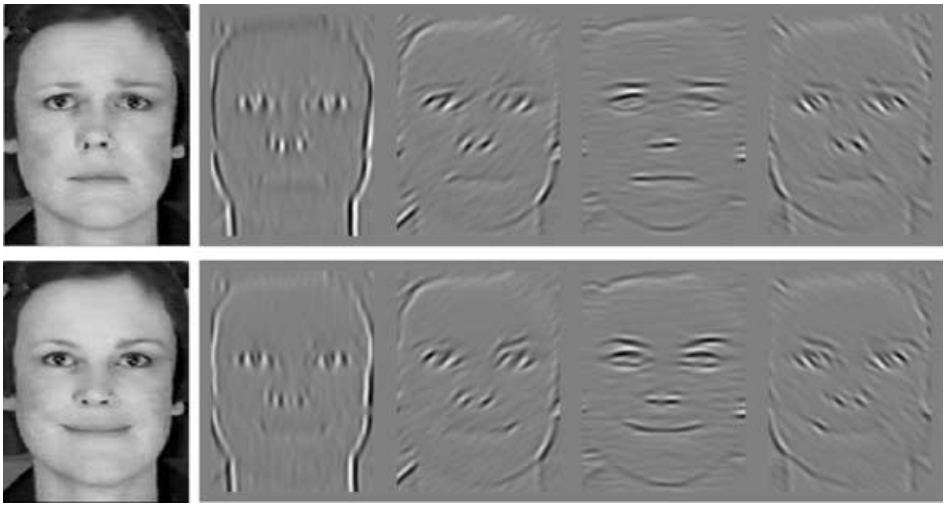


FIGURE 4.10: Gabor responses on images [11].

In 2007, Tong et al. [40] conducted a study for real time FACS AU detection. They did not just try to analyze the existence of AUs or combination of AUs, but they developed a system that could analyze the semantic and temporal relations between them. For this research, they used Gabor wavelets for facial feature extractions and used AdaBoost for classification purposes. A similar research was also done in 2008 by Kotsia et al. [56]. Their research was based on recognizing facial expressions from occluded images and to do so, they used Gabor filters and SVMs. Their research showed that occlusion of one side of the face does not affect the recognition rate for in the majority of the times, left and right sides of the face are symmetrical. Yet they also showed that effective FER

is not realistic when there is occlusion around upper or lower regions of the face. In 2010, Wu et al. [57] conducted an experiment on motion based Gabor filters for FER. They created a spatial Gabor filter bank and a spatio-temporal Gabor motion energy filter for comparison of efficiency and success rate. They used a linear SVM method for classification. Their results show that even though both methods are successful, Gabor motion energy filters are more effective. One of the most recent studies made by Gabor filters is the one conducted by Gu et al. [58]. They used Gabor filters for pre-processing and feature extraction and these extracted features were fed to local classifiers so as to generate a global classifier for FER. They also extended their research to detect facial expressions in occluded images and created global classifiers. The cross database tests they made between CK and JAFFE shows that their classifier is successful regardless of the database.

4.5.2 Haar-Like Features

One commonly used image processing filter in FER is Haar-like features. Haar-Like features are extracted by convolving rectangular shaped black and white filters that are either perpendicular or tilted with the analyzed image. The main idea is to block some rectangular areas in the filtered area in the image and taking the convolution of the area, which results in the average of the area that the filter is being applied to without the blocked partition. Haar-Like features are most commonly used in object recognition in image processing, and in FER, Haar-Like features are mainly used for face detection and/or facial feature extraction. Regular and rotated filters for extracting Haar-Like features are given in Figure 4.11.

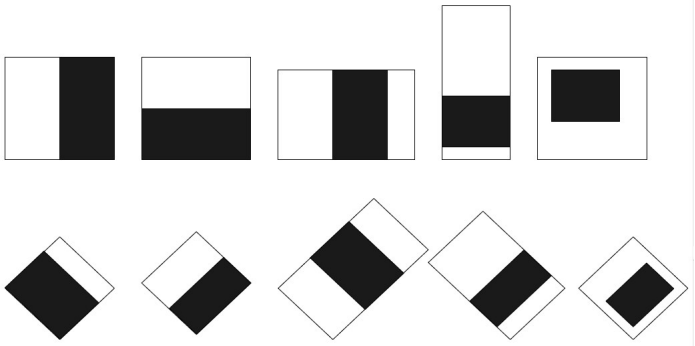


FIGURE 4.11: Some examples of filters for extracting Haar-Like features.

Researchers have started using Haar-Like features for FER quite recently and one of the initial ones to use this method for FER was Wang et al. [59]. They used Haar-Like features for feature extraction and AdaBoost for classification. Their study have shown that Haar-Like features and AdaBoost was much faster than SVM based FER when

tested on JAFFE. Another research similar to this was done by Whitehill and Omlin [12] that also used Haar-Like features for facial feature extraction. They used AdaBoost for selecting the optimal filter for Haar-Like feature extraction. They compared their results with Gabor wavelet and SVM based FER and also shown that Haar-Like features are much faster than Gabor filters. The process of feature extraction using Haar-Like features used by Whitehill and Omlin is represented in Figure 4.12. Another research that was done by Yang et al. [60] also used Haar-Like features for facial feature extraction but they used dynamic filters for feature extraction instead of static. Their method suggested using binary coding for Haar-Like feature updates and using these binary strings for weak classifiers to AdaBoost. They showed the superior efficiency of Haar-Like features for FER. In 2008, Yang et al. furthered their research [61] to also include binary classification for extracted features. They used Haar-Like features for feature extraction and used dynamic binary patterns for coding and classification of the extracted features. Their results show that their more recent work is more successful for feature extraction and classification for FER.



FIGURE 4.12: Extracting Haar-Like features from human face [12].

4.5.3 Local Binary Patterns

Local binary patterns, are binary representations of numbers that are circularly placed on a filter and then convolved with images. Local binary patterns is a different type of filter that is used for FER as well. The basics of this type of image processing filter is having either zero or one for every element in the filter and convolving the filter with the image, then taking the local histograms from partitions of the image for feature extraction. These filters are similar to Haar-Like features except that the zeros and ones do not have to be in rectangular shape. LBPs are most commonly preferred for their extraordinary tolerance for lighting conditions. This type of filtering approach is generally used with histogram evaluation of subsections in an image for FER. Representation of local binary patterns is given in Figure 4.13

In 2007, Zhao and Pietikainen [62] used rotation invariant local binary patterns for facial expression recognition as well as dynamic texture classification. They used the further

$$(216)_{10} = (11011000)_2 \rightarrow \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

FIGURE 4.13: Local binary pattern representation of an example binary string.

neighbors of convolved pixels for feature extraction and applied their suggested classification model for both dynamic texture and facial expression recognition. Zhao and Pietikainen [62] tested their system on multiple databases including CK. They showed that LBPs are more effective for situations where dynamic lighting changes occur compared to other appearance based methods. In 2008, a different approach was made by Gritti et al. [63]. Griti et al. used LBPs for FER from images which had face registration errors. They used multiple feature extraction methods and shown that overlapping LBPs are more efficient for FER by showing 92.2% emotion detection with CK. Another research was done by Shan et al. [13] that furthered the LBP approach for FER. Their study included low resolution images and comparison of different FER methods for evaluation. Their facial analysis representation is shown in Figure 4.14. Their results show that a boosted LBP approach with SVMs show the best performance and recognition rate. A more recent research was conducted by Jiang et al. [64] to exploit real time action unit detection by LBPs. Their research included a comparison between LBPs and local phase quantisation (LPQ) for real time FACS AU detection. Their results show that LPQs slightly outperform LBPs in AU detection, yet LBPs are much cheaper at computational expenses.

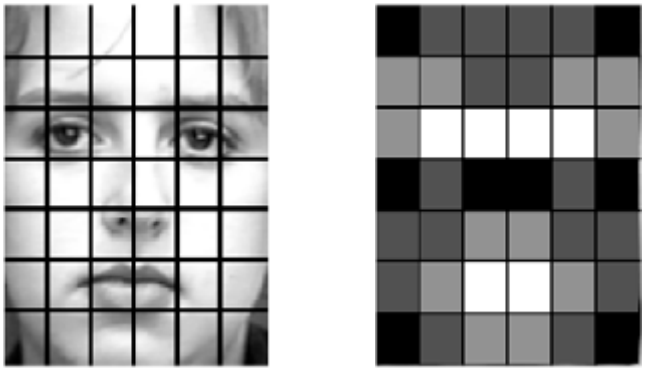


FIGURE 4.14: Facial analysis produced by Shan et al. [13].

4.6 Comparison Between Geometry and Appearance Based FER

In any given research field, there are bound to be multiple approaches. In the previous sections, most commonly used approaches are discussed , but there are advantages

and disadvantages of every proposed method. In this section, different positives and drawbacks of multiple methods in recognition success rate and speed are discussed.

4.6.1 Recognition Success Rate

The most important thing in any field is success rate and so far, many researchers have argued that their method is superior to others in success rates. In geometry based FER, Tzimiropoulos et al in 2012 [9] shows the most promising fiducial point detection, but there is not much of a comparison between their proposed system and other approaches for AAMs are not used for recognition of facial expressions, but to only detect fiducial points. As far as FER success rate goes, 3D analysis is the one that can be compared to appearance based models. Study conducted by Maalej et al. [52] showed an average of 98.81% success rate when detecting emotions which has never been achieved by appearance based models. Yet, the work of Wu et al. [57] came very close with 97.81% when using Gabor motion energy filters for feature extraction. Also, results by Gu et al. [58] were quite promising when compared to geometry based FER. Nevertheless it remains that 3D models slightly outperform appearance based models as far as recognition rate is discussed such like the work done by Segundo et al. [55].

4.6.2 Speed

Another part that needs to be discussed is the computational speed of the given approaches. In the previous sections, it was made clear that the work done with LPBs [13, 64] and Haar-Like features [12, 60] are very fast and effective when applied with AdaBoost. Yet in AAMs, the initial facial feature point detection is the time consuming part [50]. Yet, it is done once at the initialization. Also the work done by Valstar et al. [49] to reduce the grid search area of AAMs showed that the speed of AAMs can be increased. It is the Gabor wavelet and 3D approaches that take the most time because Gabor wavelets are used with 40 different wavelets [57, 58] and these filters have two sections which are the real and imaginary parts which take computational time, and 3D analysis simply have way too much data to work on [52, 54, 55]. In conclusion, AAMs, Haar-Like features and LBPs are superior to other methods as far as speed analysis goes.

4.7 Hybrid Systems

Like in any field, combination of approaches have been applied in FER as well to take advantage of different approaches. In this section, some of the applied hybrid systems

are explained.

Quite recently, Maaten and Hendriks [65] conducted a research that uses both AAMs and Gabor wavelets for automatic AU detection. They applied AAM grids on given images and applied multiple texture models on them for AU detection and received high success rates with CK+. Another similar research was done by Chen et al. [14] that also used geometric fiducial point detection and texture differences. Yet Chen et al. [14] used the extracted features for emotion detection rather than AU detection. Chen et al. [14] also tested their system on CK+ and received an average of 95.64% success rate when detecting emotions. Texture component analysis applied by Chen et al. [14] is represented in Figure 4.15.

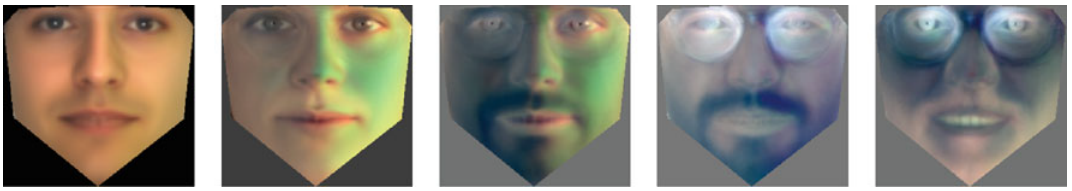


FIGURE 4.15: Texture component analysis by [14].

4.8 FACS AU Intensity Detection

In the field of FER, not much research has been conducted on AU intensity detection, so AU intensity detection is still an important area to be exploited. Some of the research that has been done so far for FACS AU intensity detection are discussed in this section.

One of the very first attempts to detect the FACS AU intensity was done by Fasel and Luettin [66] who applied principal component analysis (PCA) and independent component analysis (ICA) so as to analyze images and estimate FACS AU intensity. They showed the PCA is more effective than ICA, for ICA is quite sensitive to noise. In 2006, Bartlett et al. [67] revisited FACS AU intensity estimation with real-time application. They used Gabor wavelets for feature extraction, AdaBoost for feature selection and dimensionality reduction. Both SVMs and regression analysis were used for classification. They received a mean rate of 93% recognition with 20 AUs for the detection of AU. From these 20 AUs, Bartlett et al. [67] selected six AUs (AU1, AU2, AU4, AU5, AU10 and AU20) for AU intensity detection. Their method received 83% recognition success for FACS AU intensity detection with within-subject testing (training and testing images were from the same subject) and 53% recognition success with between-subject testing (training and testing images were from different subjects). Another aspect of FACS AU intensity detection was researched by Mahoor et al. [68] in 2009. Mahoor et al. conducted a research on detection and intensity estimation of AUs in non-posed

mother-infant relations. They used AAMs for facial representation, and applied spectral regression for dimensionality reduction. They trained a SVM based system for automatic recognition of intensities for AU6 and AU12 (combination of AU6 AU12 is happy). In 2012, Savran et al. revisited FACS AU intensity detection [69] with a new aspect of 3D analysis. They applied the probability estimation property of logistic regression for FACS AU intensity detection in 2D images. They also showed that for 3D images, AdaBoost shows superior efficiency for feature selection. Their regression based method showed better performance to previously suggested SVM based methods. They also showed that it is both possible and feasible to generate an intensity estimation system that uses facial grid and/or facial fiducial point detection. In conclusion, they showed that using the combination of 3D and 2D images gives better results compared to either one individually. Quite recently, another research for FACS AU intensity detection was conducted by Jeni et al. [15]. Their system consists of four parts which are; fiducial point detection by constrained local models (a different type of AAM), local patch removal using fiducial points, application of non-negative matrix factorization, and finally, training the SVM using the extracted features. Their feature extraction method is demonstrated in Figure 4.16 They coded some of the images in CK+ between 0 to B level intensities of AUs AU1, AU2, AU4, AU5, AU6, AU7 and AU9. They used the aforementioned coded on data with the apex property of the extended CK and BU-3DFE dataset for training and testing. They received an average of 0.894 when the results were analyzed as Pearson's Correlation Coefficient. Another very recent study was conducted by Mavadati et al. [8] when they were releasing DISFA dataset. Using their own dataset, they conducted multiple experiments to analyze FACS AU intensity detection. They used multiple different feature extraction methods with SVM classifiers and compared the results. Feature extraction methods used by Mavadati et al. [8] were Local Binary Pattern Histogram (LBPH), Histogram of Gradient (HOG) and Gabor wavelets. They received 0.69, 0.70 and 0.77 ICC with LBPH, HOG and Gabor wavelets respectively. They also received 81.54%, 79.14% and 85.71% recognition rates with LBPH, HOG and Gabor wavelets respectively. Their study showed that amongst the three feature extraction methods applied so far, Gabor wavelets give the most successful results with 85.71% recognition rate and 0.77 ICC.

4.9 Pain Analysis by Facial Expression Recognition

Pain is another fundamental state of consciousness humans express via facial actions. In the recent years, research fields related to analyzing pain in humans has made good progress. In the quest to analyze and understand pain, different methods including

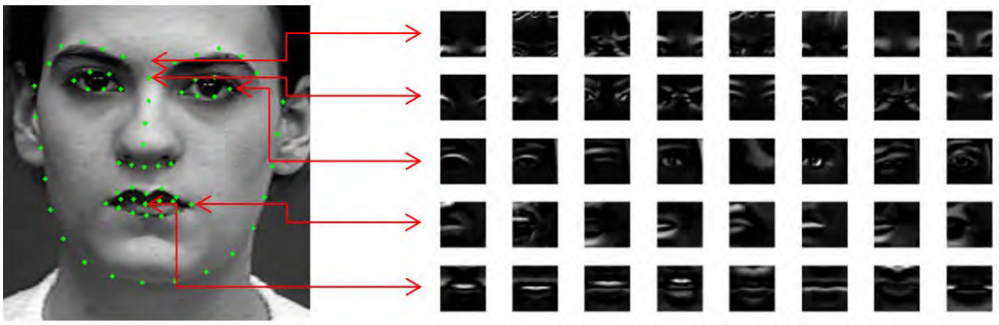


FIGURE 4.16: Extracted features using fiducial points and image filters by Jeni et al. [15].

automated facial action analysis has been applied to detect the existence of pain, differentiating between real and fake pain, and estimating the intensity of pain. Also, manual analysis of faces has been done so as to understand pain in human beings.

4.9.1 Pain Detection

One of the important aspect of pain in automated FER is detection of pain. In 2011, Lucey et al. [16] conducted a research on pain detection in adults using FACS AUs. Their research included AAM fitting for feature extraction, and SVM for feature classification. For feature extraction, they used S-PTS, S-APP, and C-APP. S-PTS refers to facial feature points where all the rigid geometric variations (translation, rotation and scale) are removed [16]. S-APP refers to facial image where all the rigid geometric variations (translation, rotation and scale) are removed [16]. C-APP refers to canonical-normalized appearance which is obtained by piecewise affine warp on each triangle patch appearance in the source image so that it aligns with the base face shape [16]. S-PTS, S-APP and C-APP are show in Figure 4.17. For classification, the distance between decision hyperplanes (for each AU detector) and the tested subject is taken. These distances were then used as inputs for logistic regression training. They used the UNBC-McMaster Database [7] for training and testing. Their results show that the recognition performance for both AUs and pain is the most successful when S-PTS + S-APP + C-APP is used for features. In 2012, another research group lead by Lucey [70] conducted additional experiments with UNBC-McMaster Database [7] to further the previous research on automated pain analysis in [16]. In this research, they also used AAMs for feature extraction, but for this one, they extracted 3D information from 2D AAM using non-rigid structure-from-motion. For this research, Lucey et al. used binary detection of AUs to detect if pain is existent or not in a human face. They used SVMs for binary classification of both AUs, and pain. Their research was also based on frame-level and sequence-level pain detection. Frame-level analysis refers to analyzing individual frames,

and sequence-level refers to analyzing the sequence of consecutive frames from the same session. They showed that both frame-level and sequence level pain detection is successful, but the speed of the system is faster with sequence based analysis. Chen et al. [71] extended the previously suggested methods in pain detection using transformation based classification. They addressed the problem of pain being subjective, and the fact that there is not much data to be trained on. They compared traditional AdaBoost classification with new-generation AdaBoost classification that they suggested. Their classification method was based on transforming learned parameters from one subject to the learning process of another to boost the performance and speed of the system. They used LBPs for feature extraction. Their study was based on detecting the presence of pain and they received 0.895 in terms of area under the ROC curve.

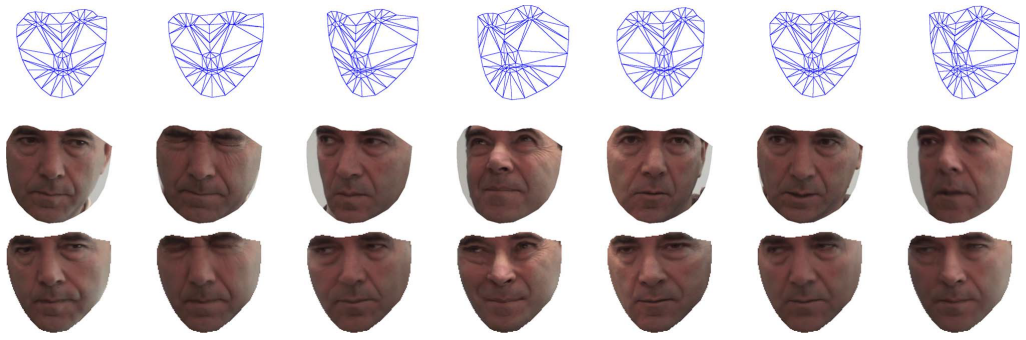


FIGURE 4.17: S-PTS, S-APP and C-APP used by Lucey et al.. First row shows S-PTS, second row shows S-APP and third row shows C-APP [16].

4.9.2 Pain vs Fake Pain

Analysis of pain has been a subject for medical, and psychological researchers for years and another important aspect of expression of pain in humans is differentiation of pain and fake-pain.. The physical assessment of pain has been puzzling until Hill and Craig [72] analyzed the pain expressions of adults to detect how facial actions are related to pain. Using FACS AUs, they were able to associate facial actions with pain. They used 30 lower-back pain subjects (23 male, 17 female) from the age group 19-27 to measure the relativity of AUs to genuine, faked, and masked pain. Their research showed that AUs are directly related to pain in adults. They also showed that in faces with fake pain, AUs tend to reach the peak intensity faster, and last longer at peek intensity compared to genuine pain. Furthermore, they concluded that faces with faked pain and real pain differ in activated AUs as well. Given that AUs are directly related to pain, Larochette et al. [73] extended the research of Hill and Craig [72] to the extent of children. They used cold pressor equipment for assessing pain in 50 children (25 male, 25 female). They also followed the FACS standard for facial action analysis and shown that children show more facial actions during pain and they are more prone to and successful at hiding

their pain. However, both researches show that facial actions are directly related to pain. One of the very first automated pain assessment system involving real vs fake-pain was proposed by Littleworth et al. [74]. They analyzed 26 subjects under three conditions; neutral, genuine, and faked pain. They used cold-pressor for genuine pain. Approximately 5000 frames were manually coded for FACS AUs. They extracted facial features by Gabor wavelets, and trained an SVM classifier for the detection of 20 AUs. The results of the said SVM classifier were used for real-fake pain detection. Some of the frames they used were analyzed by naive observers for real and fake pain, and the results of said naive observers were compared with the results of Littleworth et al. [74]. While naive observers were able to reach an average of 49% recognition rate for real-fake pain, their system received 88%.

4.9.3 Pain in Neonates

Pain is an important aspect in patient care for adults, and children, but pain assessment of neonates or infants is even more puzzling for they are physically incapable of expressing pain verbally. Gholami et al. [75] did a research on assessing the pain in neonates using relevant vector machines and AAMs in 2010. A relevant vector machine is a bayesian extension of the regular SVM. They used the COPE database [76] for training and testing, which includes 26 caucasian newborns with pain and no-pain labels. They also labeled every frame manually with a pain intensity level between 0-100. They showed that their method out-performs both non-expert and expert humans. In 2010, Nanni et al. [77] conducted a set of experiments to observe the reliability of LBPs for medical image processing, and one part of said research was neonate pain. They tested regular LBP, an extension known as Local Ternary Patterns (LTP), and Elongated Quinary Patterns (EQP). LTP is a 3-valued coding (-1,0,1) that includes a threshold around zero for the evaluation of the local gray-scale difference. Elongated quinary patterns is an extension of LTP that uses 5 values (-2,-1,0,1,2).Local The classification was done by SVMs, and they also used COPE database [76]. It is shown that not only for neonate pain detection, but for other medical image analysis, quinary pattern approach for feature extraction gives better results compared to the other variants of LBPs.

4.9.4 Pain Intensity

In 2012, Kaltwang et al. [78] extended the previously done researches in pain intensity in neonates to adults. They used AAMs for feature extraction and Relevance Vector Regression (RVR) for classification. Their research mainly focused on continuous intensity estimation of pain in adults. They have extracted three sets of features from AAMs.

First set was the 66 AAM landmark points that was aligned via procrustes analysis. For the second set of features, they applied Discrete Cosine Transform (DCT) to the first feature set. Final feature set was LBP extracted from previously aligned faces. For classification, they used RVR which models the target regression function by using representative cases. They studied pain intensity detection from PSPI and FACS AU intensities. Kaltwang et al. [78] reported that pain intensity detection from FASC AU intensities received a higher classification rate compared to direct pain intensity detection from PSPI. In 2009, Ashraf et al. [79] conducted a set of experiments in order to analyze the effects of sequence-level and frame-level labeling of pain. For pain intensity labeling they derived the intensities of AUs, and used the sum of total intensities of AUs that are relative to pain. They also used AAMs for feature extraction, and SVM for classification. The features they extracted were; S-PTS, S-APP and C-APP. Their research showed that when using frame by frame labeled dataset, classification of pain is more successful compared to sequence-level labeled dataset al.so, they showed that C-APP + S-PTS were the most robust method for feature extraction. One of the most remarkable studies including pain intensity detection was done by Hammal and Cohn in 2012 [80]. They used the UNBC McMasters database [7] to analyze the intensity of pain in humans. They divided the PSPI pain level of subjects into four labels; no pain, trace, mediocre, and heavy. They used an appearance based feature extraction method with the help of AAM fitting, and four different SVMs to classify between pain intensities. The feature extraction method used in [80] is the normalized canonical normalized appearance of the face (CAPP). They received a classification success between 40% and 67% with 5-folds leave-one-out testing method. Another study conducted on pain intensity was done by Chew et al. in 2012 [81]. They divided the PSPI level of frames into six levels and used the same approach of [80] for feature extraction (CAPP). Their study mainly focused on different SVM methods and with modified correlation filter (MCF), they were able to receive above 61% recognition success for pain intensity classification. Another research that focused on pain intensity was done by Rudovic et al. in 2013 [82]. Their focus on the subject was to create a new type of classifier that classifies with higher accuracy than SVMs, and gives more robust results when detecting the level of existence rather than giving binary presence-absence classification. They reached their goal using Conditional Ordinal Random Fields (CORF) for classifier, and LBP for feature extraction. Conditional ordinal random fields is an extension of logistic regression for sequential data. They extended the CORF method to relax the homoscedasticity assumption of CORF. Through experimentation, they came to the conclusion that accounting for heterogeneity in the dataset would give better results. Another application of random fields were proposed by Bousmalis et al. [18]. Their approach was to use infinite Hidden Conditional Random Fields (iHCRF). They suggested that since HCRFs were proven to be very successful at extracting the hidden patterns, infinite version of

HCRFs would be more than sufficient for pain intensity estimation. By their definition, iHCRF is a nonparametric model based on hierarchical Dirichlet processes, and is capable of learning the amount of necessary hidden states for classification [18]. They have shown that the method they applied which includes Markov-chain sampling for hyper-parameter learning, and iHCORF with Restaurant Franchise Rating Agencies analogy is more robust, and faster compared to any finite HCRF. They have tested their proposed method with four different labeling systems of pain; Agreement-disagreement of pain existence, agreement-disagreement and neutral to pain, pain recognition with two labels (strong vs weak pain), and pain recognition with three labels (strong, moderate and low pain). Through experimentation, they were able to show that iHCRF outperforms HCRF in all of the four cases, and were able to receive 88.4% recognition rate with two label pain intensity estimation in the UNBC-McMaster Database [7].

4.10 Summary

In this chapter, both the more recent and more commonly used FER methods are discussed. The review starts with the discussion of more commonly used FER databases as well as their content and formats. Then, the more commonly used and more robust face detection and extraction methods are explained. In the next part, the two main sections of FER which are; geometry based FER, and appearance based FER, are explained as well as their comparisons and hybrid systems between them. Afterwards, recent studies which focused on FACS AU intensity detection, and pain are discussed. In the next chapter (Chapter 5), methods applied in this research are explained.

Chapter 5

Methodology

5.1 Introduction

In Chapter 1, it is hypothesized that an effective facial action intensity detection system can be created by extracting geometric facial features and classifying by supervised machine learning methods. The methodology of this study is explained in this chapter. The contents of the methodology are feature extraction, feature selection and classification.

5.2 Feature Extraction

The suggested classification methods for this research are supervised machine learning methods. Supervised machine learning methods do not necessarily require feature extraction methods. Raw images can simply be fed to supervised machine learning methods. Yet it would be unrealistic to assume raw images would be sufficient for facial action intensity classification. Therefore a feature extraction method is required to extract only the relevant features. Features are mathematical representation of data. In this section, the methods for normalization and feature extraction from the data are explained.

5.2.1 Facial Grid and Normalization

The initial process of feature extraction in this study is facial grid detection. Facial grid of an individual can be detected by multiple methods such as active appearance models [9], or facial feature detectors [83]. Yet the main purpose of this study is not to create a robust method for facial grid detection, but to create a robust method for

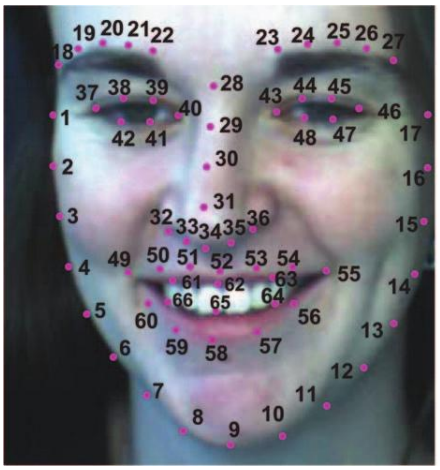


FIGURE 5.1: Facial feature points given in DISFA database [8].

facial action intensity detection. Therefore, the facial feature landmarks given in DISFA database [8], and UNBC-McMaster database [7] are used. The landmarks from DISFA database are given in Figure 5.1.

After receiving the facial feature points, a normalization process is applied. It should be noted that normalization are applied only to vectorial features Section 5.2.3, because angular features are rotation invariant.

The applied normalization process is to rotate the landmark points by the tip of the nose to make sure the nose is perpendicular to the x-axis.

Given the certain set of facial feature points $P^{(i)}_{(x,y)}$, the initial process is to move the center of the grid to $(0, 0)$, where P^{31}, the point on the tip of the nose, is accepted as the center of the grid. Then, the grid is rotated so that the nose is perpendicular to the x-axis.

The center of the grid $(P^{(C)}_{x,y})$ which is P^{31} is moved to the center $(0, 0)$.

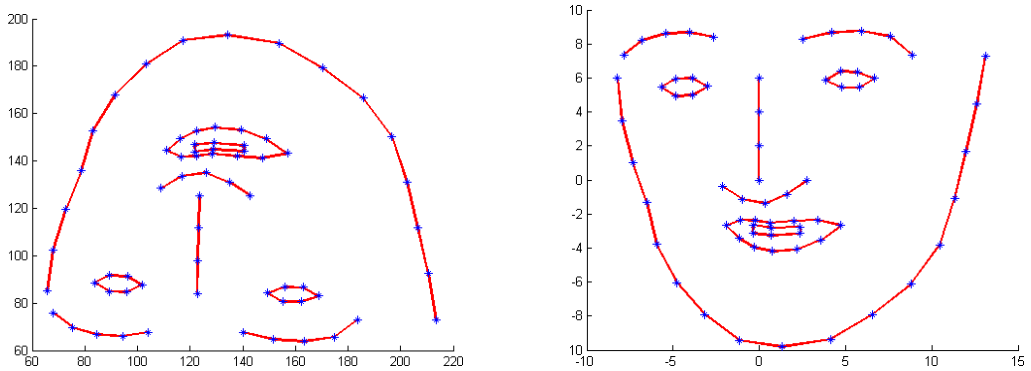
$$P^{(i)}_{x,y} \leftarrow P^{(i)}_{x,y} - P^{(C)}_{x,y} \forall i \in \{1, 2, ..., 65, 66\} \tag{5.1}$$

The distance (d) between $P^{(T)}$ and $P^{(B)}$ is calculated. $P^{(T)}$ represent the top point on the nose (P^{28}) and $P^{(B)}$ represent the lowest point on the nose $(P^{34}.)$

$$d = \sqrt{(P^{(T)}_x - P^{(B)}_x)^2 + (P^{(T)}_y - P^{(B)}_y)^2} \tag{5.2}$$

Rotation angle β is calculated.

$$\beta = \arccos(\frac{(P^{(T)}_x - P^{(B)}_x)}{d}) - \pi \tag{5.3}$$



(a) Before normalization (b) After normalization

FIGURE 5.2: AAM grids before and after normalization.

All the points in the grid is rotated so that the nose is perpendicular to the x-axis.

$$P_{x,y}^{(i)} \leftarrow P_{x,y}^{(i)} * \begin{bmatrix} cos(\beta) & -sin(\beta) \\ sin(\beta) & cos(\beta) \end{bmatrix} \tag{5.4}$$

π is subtracted from β because the coordinates in the image are upside down when mapped to the cartesian coordinates, and another π rotation is necessary to make sure the grid is perpendicular to the x-axis.

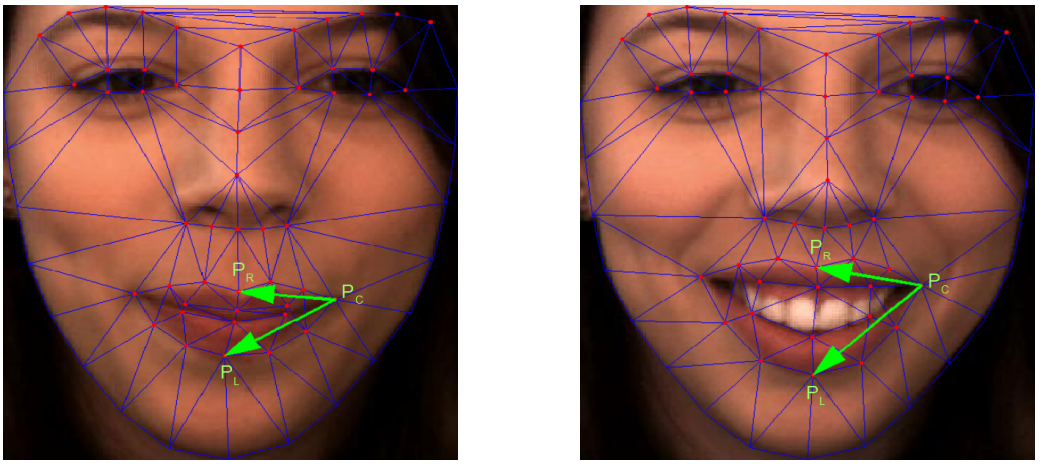
The last part of the normalization process is to make sure that the distances of the feature points are normalized. For that, the distance between points $P_x^{(T)}$ and $P_x^{(B)}$ is fixed by a predefined value ψ. In this study ψ was taken as 1. The facial feature points before and after normalization are given in Figure 5.2.

$$P_{x,y}^{(i)} \leftarrow \frac{P_{x,y}^{(i)}}{d} * \psi \tag{5.5}$$

5.2.2 Angular Features

In order to alleviate from the problems caused from in-plane rotations, rotationally invariant features are extracted. For a given center facial feature point P_C, the angle between two vectors directing from P_C to two facial feature points P_L and P_R is computed (see Figure 5.3). A feature vector $\mathbf{x} \in \mathbb{R}^{D\times 1}$ (D representing the size of the feature vector) is created by concatenating all angles $\left\{x^i\right\}_{i=1}^{D}$, where $x^i \in \mathbb{R}$, computed from the set of ternary facial feature points $\left\{\left(P_C^i, P_L^i, P_R^i\right)\right\}_{i=1}^{D}$, i.e.,

$$x^i = \arccos\left(\overrightarrow{P_L^i P_C^i} \cdot \overrightarrow{P_R^i P_C^i}\right) \tag{5.6}$$



(a) Neutral Face (c) Smiling Face

FIGURE 5.3: Example P_C, P_L, P_R points, and vectors for feature extraction [8].

where $\overrightarrow{P_L^i P_C^i}$ and $\overrightarrow{P_R^i P_C^i}$ are unit vectors pointing from P_C^i to P_L^i and P_C^i to P_R^i, respectively. Selection of ternary facial feature points are shown in Figure 5.4.

The difference from the average feature vector $\hat{\mathbf{x}}_n$, that is computed from an average neutral face, is used as the final feature vector, i.e.,

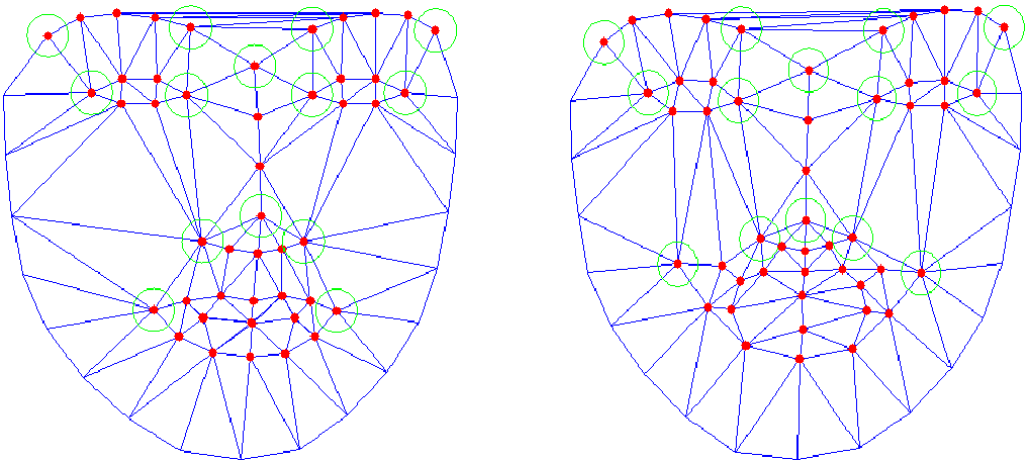
$$\mathbf{x} \leftarrow \mathbf{x} - \hat{\mathbf{x}}_n. \tag{5.7}$$

The difference between $\mathbf{x}$ and $\hat{\mathbf{x}}_n$ is used because the extracted angles already exist in a neutral face, and the changes in the angles are the actual values that are relevant for facial action intensity detection. This approach speeds up the training process rather than improve the recognition rate.

5.2.3 Vectorial Features

The second set of features extracted for classification is the displacement of facial feature points on the AAM grid compared to the neutral facial grids. A feature vector $\mathbf{x} \in \mathbb{R}^{D\times 1}$ is computed from concentrating $\{x^{(i)}\}_{i=1}^{\frac{D}{3}}$, $\{y^{(i)}\}_{i=1}^{\frac{D}{3}}$, and $\{l^{(i)}\}_{i=1}^{\frac{D}{3}}$, where $x^{(i)}$, and $y^{(i)}$ denotes the x and y components, $l^{(i)}$ denotes the magnitude of the displacement vector, and D represents the size of the feature vector. Given the facial feature points $\mathbf{P}_{(\mathbf{x},\mathbf{y})}{}^{(i)}$, and the neutral grid points $\mathbf{Ne}_{(\mathbf{x},\mathbf{y})}{}^{(i)}$ (derivation of $\mathbf{Ne}_{(\mathbf{x},\mathbf{y})}{}^{(i)}$ are explained in Chapter 6), the vectorial features are computed, i.e.,

$$x^{(i)} = \mathbf{Ne}_{(\mathbf{x})}{}^{(i)} - \mathbf{P}_{(\mathbf{x})}{}^{(i)} \tag{5.8}$$



(a) (b)

FIGURE 5.4: Selected Feature Points for Feature Extraction. Blue Grid: Complete AAM Grid, Red Dots: Selected feature points for feature extraction $\{P_R, P_L\}$, Green Circled Points: Selected feature points for angle computation P_C. a: Average Neutral Face Grid, b: AAM Grid from subject 11 with AU6B + AU12C + AU25C.

$$y^{(i)} = \mathbf{Ne}_{(\mathbf{y})}{}^{(i)} - \mathbf{P}_{(\mathbf{y})}{}^{(i)} \tag{5.9}$$

$$l^{(i)} = \sqrt{(\mathbf{Ne}_{(\mathbf{x})}{}^{(i)} - \mathbf{P}_{(\mathbf{x})}{}^{(i)})^2 + (\mathbf{Ne}_{(\mathbf{y})}{}^{(i)} - \mathbf{P}_{(\mathbf{y})}{}^{(i)})^2} \tag{5.10}$$

After the components of the vectorial features are computed, the final feature vector becomes:

$$\mathbf{x} = \left[\frac{x^{(i)}}{l^{(i)}}, \frac{y^{(i)}}{l^{(i)}}, l^{(i)} \right]_{i \in \frac{D}{3}} \tag{5.11}$$

Example vectorial features are given in Figure 5.5

5.3 Feature Selection

The suggested feature extraction methods result in a large feature vector which would increase the computation time. It is unrealistic to assume that all the elements of the feature vector would be relevant to the problem. To make sure that the system is computationally efficient and more robust, a genetic feature selection algorithm is employed.

Genetic algorithms are computation methods that replicates the biologic evolution of populations. Genetic algorithms start with an initial population $Q^{(1)}_{D \times P}$ with D amount of genes that are randomly created for P amount of subjects. Then every subject in the population goes through a fitness test $\Gamma\{Q^{(i)}\}$ where i represents the iteration of the

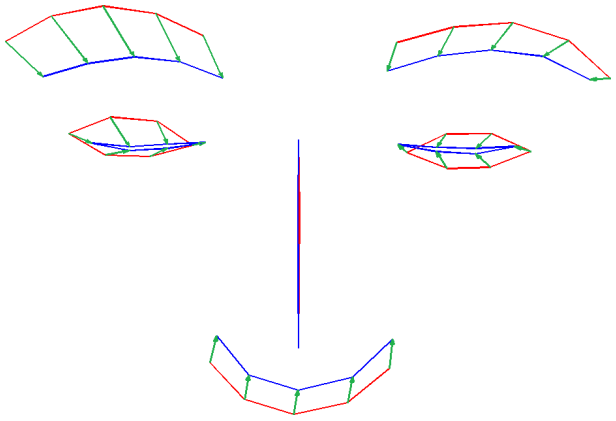


FIGURE 5.5: Vectorial features extracted from the upper grid of the face from a subject in UNBC McMasters Painful data database [7]. Red grids represent average neutral facial grid of the subject, blue grid represent the facial grid of the same subject with pain level 11 (by PSPI scale), green vectors represent the extracted feature vectors.

genetic algorithm. After the fitness test is finalized, the next population is created by crossing over the genes of selected subjects with a mutation chance and immigration of new subjects. This procedure continues either until the population converges to a set of genes or another condition is satisfied such as maximum number of populations.

The genes in the subjects are binary values representing whether the corresponding feature is used, or not. The fitness function $\Gamma\{Q^{(i)}\}$ for population is either logistic regression, or neural network classification error. After the fitness of each subject in the population $\omega^{(p)} = \Gamma\{Q_p^{(i)}\}\ \forall p \in P$ is computed, $\frac{P}{3}$ amount of subjects are selected from the population via rotating wheel directly proportional to $1 - \omega$ as parents for re-population. The remaining subjects are discarded, and another $\frac{P}{3}$ amount of subjects are created from the parents via crossover and mutation. The remaining $\frac{P}{3}$ subjects are taken as immigrating subjects, and are randomly created.

This method is re-iterated until the parent subjects are converged to a certain gene pool, or an iteration amount of 200 is reached.

5.4 Classification

5.4.1 Logistic Regression

Logistic regression is a probabilistic classification method. The system learns a hypothesis $H_{\mathbf{\Theta}}$ that is parameterized by $\mathbf{\Theta} \in \mathbb{R}^{(D+1)\times 1}$ for feature vector $\mathbf{x}$ [84], i.e.,

$$H_{\mathbf{\Theta}}(\mathbf{x}) = \frac{1}{1 + e^{-(\mathbf{\Theta}_0 + \mathbf{\Theta}_R^T \mathbf{x})}}, \tag{5.12}$$

where,

$$\mathbf{\Theta} = \begin{bmatrix} \mathbf{\Theta}_0 \\ \mathbf{\Theta}_R \end{bmatrix}, \ \mathbf{\Theta}_0 \in \mathbb{R}, \ \mathbf{\Theta}_R \in \mathbb{R}^{D \times 1}, \text{ and } H_{\mathbf{\Theta}}(\mathbf{x}) \in [0, 1] \tag{5.13}$$

The parameters of $H_{\mathbf{\Theta}}$ are learned using a training set $\left\{\left(\mathbf{x}^i, \mathbf{y}^i\right)\right\}_{i=1}^{N}$ of size N, where for each feature vector $\mathbf{x}^i$, $\mathbf{y}^i$ represents the corresponding intensity which is mapped to the interval of $[0, 1]$. The parameters are iteratively learned by a minimizing mean square error between the real intensity values $\mathbf{y}^i$s and the predicted intensity values $\hat{\mathbf{y}}^i$s resulted from the regression, i.e.,

$$MSE(\mathbf{\Theta}) = \frac{1}{N} \sum_{i=1}^{N} \left(\mathbf{y}^i - H_{\mathbf{\Theta}}\left(\mathbf{x}^i\right)\right)^2. \tag{5.14}$$

As a minimizer of the mean square error function, gradient descent algorithm is employed.

$$\delta^{(i)} = \hat{\mathbf{y}}^{(i)} - \mathbf{y}^{(i)} \tag{5.15}$$

$$\mathbf{\Theta}_0 \leftarrow \mathbf{\Theta}_0 - \alpha \frac{1}{N} \sum_{i \in N} \delta^{(i)} \tag{5.16}$$

$$\mathbf{\Theta}_R \leftarrow \mathbf{\Theta}_R (1 - \alpha \frac{\kappa}{N}) - \alpha \frac{1}{N} \sum_{i \in N} \mathbf{x^{(i)}} * \delta^{(i)} \tag{5.17}$$

Where α, and κ stand for learning rate and regularization parameters respectively.

5.4.2 Feed Forward Neural Networks

The second classification method employed in this study is neural networks. Neural networks are machine learning methods that mimic the actual working method of neurons in brains. They consist of three parts; input layer, hidden layer(s), and output layer [84]. The suggested neural network is a one hidden layer network with four output nodes. The neural network has four output nodes because the pain levels are divided into four states. Therefore, the result of the neural network is derived by detecting the index of $max(u_i^{(3)})$ where $i \in [0, 4]$. The structure of the suggested neural network is in Figure 5.6.

For the proposed method, the output of the network is computed via propagating the activation function $g(S)$ values from the input layer to the output layer.

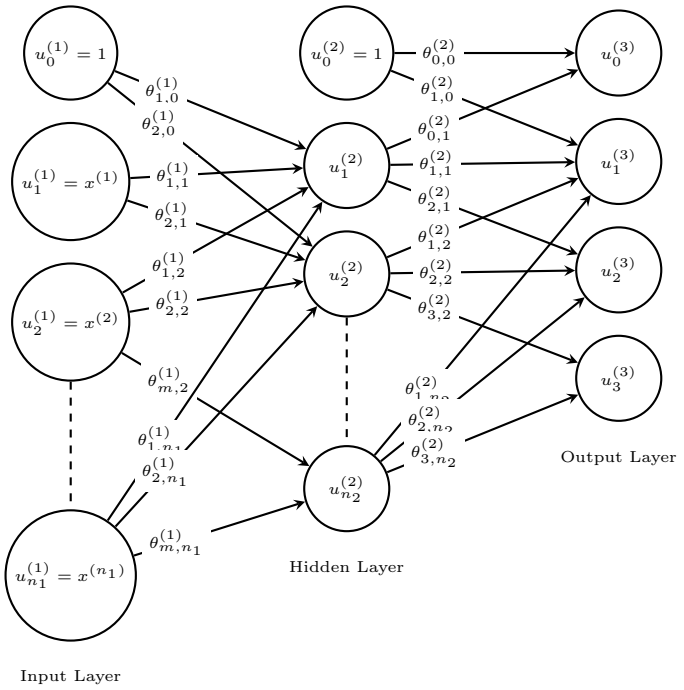


FIGURE 5.6: The structure of the 1 hidden layer, multi-class neural network

$$u_0^{(t)} = 1, t \in [1, 2] \tag{5.18}$$

$$u_i^{(1)} = \mathbf{x}^{(i)} \tag{5.19}$$

$$u_i^{(t)} = g(S_i^{(t)}), t \in [2, 3] \tag{5.20}$$

where

$$S_i^{(t)} = \sum_{j \in n_{(t-1)}} u_j^{(t-1)} * \theta_{j,i}^{(t-1)} \tag{5.21}$$

and

$$g(S) = \frac{1}{1 + e^{-S}} \tag{5.22}$$

Where, $u_i^{(t)}$ represents the activation value of i^{th} node of the t^{th} layer and $g(S)$ represents the activation function.

Forward propagating in a neural network is quite simplistic, but the training part is not only accomplished by forward propagating. For the training of neural network, gradient

descent algorithm is used, and for gradient descent to be accomplished, back-propagation is applied.

Since back-propagation is the application of gradient descent in neural networks, the weight update rule is derived from partial derivation of each θ value by the output [84].

$$\theta_{j,k}^{(i)} \leftarrow \theta_{j,k}^{(i)} - \alpha \frac{\partial F(\mathbf{\Theta})}{\partial \theta_{j,k}^{(i)}} \tag{5.23}$$

Where α, $F(\mathbf{\Theta})$, and $\theta_{j,k}^{(i)}$stand for learning rate, the cost function of neural network, and the θ value between the i^{th} and $(i+1)^{th}$ layer, mapping the k^{th} node of i^{th} layer to the j^{th} node of $(i+1)^{th}$ layer respectively.

The partial derivative $\frac{\partial F(\mathbf{\Theta})}{\partial \theta_{j,k}^{(i)}}$ in the learning algorithm is derived by the chain rule, starting from the output node, all the way to the input node.

Initially, the θ values that map the hidden layer, to the output layer are updated by the outcome of the neural network.

$$\frac{\partial F(\mathbf{\Theta})}{\partial \theta_{j,k}^{(2)}} = \frac{\partial F(\mathbf{\Theta})}{\partial u_j^{(3)}} \frac{\partial u_j^{(3)}}{\partial S_j^{(3)}} \frac{\partial S_j^{(3)}}{\partial \theta_{j,k}^{(2)}} \tag{5.24}$$

To calculate this partial derivative, the derivative of the sigmoid function needs to be derived.

$$\frac{d}{d_S} g(S) = g(S) * (1 - g(S)) \tag{5.25}$$

Having computed the $\frac{d}{d_S} g(S)$, the partial derivative for the last layer of thetas is calculated as such:

$$\frac{\partial F(\mathbf{\Theta})}{\partial \theta_{j,k}^{(2)}} = \frac{u_j^{(3)} - y_j}{u_j^{(3)}(1 - u_j^{(3)})} u_j^{(3)}(1 - u_j^{(3)}) u_k^{(2)} \tag{5.26}$$

Once the said equation is simplified, the rule for updating $\theta_{j,k}^{(2)}$ becomes:

$$\theta_{j,k}^{(2)} \leftarrow \theta_{j,k}^{(2)} - \alpha(u_j^{(3)} - y_j) u_k^{(2)} \tag{5.27}$$

The second part for updating the θ values is updating the $\theta_{j,k}^{(1)}$. The same manner of partial derivation is applied to derive the update rule [84].

$$\frac{\partial F(\mathbf{\Theta})}{\partial \theta_{j,k}^{(1)}} = \frac{\partial F(\mathbf{\Theta})}{\partial u_j^{(2)}} \frac{\partial u_j^{(2)}}{\partial S_j^{(2)}} \frac{\partial S_j^{(2)}}{\partial \theta_{j,k}^{(1)}} \tag{5.28}$$

Once this derivation is expanded, $\partial F(\mathbf{\Theta})$ becomes;

$$\frac{\partial F(\mathbf{\Theta})}{\partial \theta_{j,k}^{(2)}} = (u_l^{(3)} - y_l)\theta_{l,j}^{(2)} u_j^{(2)}(1 - u_j^{(2)})u_k^{(1)} \tag{5.29}$$

Finally, the rule for updating $\theta_{j,k}^{(2)}$ becomes;

$$\theta_{j,k}^{(2)} \leftarrow \theta_{j,k}^{(2)} - \alpha \sum_{l=0}^{3} [(u_l^{(3)} - y_l)\theta_{l,j}^{(2)}] u_j^{(2)}(1 - u_j^{(2)})u_k^{(1)} \tag{5.30}$$

The procedure of forward-propagation, and back-propagation is applied on the system using all the training samples until the system converges to a point, or another condition is satisfied. For the suggested system, training process ends either if 90% of the training samples are classified correctly, or an iteration amount of 500 is reached.

5.5 Summary

In this section, the computational methods for this research are explained in detail. Initially, facial grid derivation and normalization by nose are explained. Then, the acquisition process of two sets of geometric features (angular and vectorial), and feature selection process (genetic algorithm) are explained. Finally, the logistic regression training via gradient descent, and feed-forward neural networks as well as back-propagation derivation are shown. Experiments, results and discussion are given in the next chapter (Chapter 6).

Chapter 6

Experiments, Results and Discussion

6.1 Introduction

This research is conducted to answer three research questions.

1. How can the facial feature points and facial grid be used effectively to analyze the intensities of facial actions?
2. What type of geometric relations between facial feature points and facial grid would be more efficient and successful for facial action intensity detection?
3. Is logistic regression analysis technique or FFNN more effective for classifying facial actions and intensities?

This chapter contains the experimental setups and the results of said experiments that are conducted to answer these questions. FACS AU intensity detection is explained in Section 6.2 and pain intensity detection is explained in Section 6.3

6.2 FACS AU Intensity Detection

The very first experiment conducted to detect facial actions is to detect the FACS AUs from frontal facial images. This experiment was done with rotation-invariant angular features (Section 5.2.2) and logistic regression (Section 5.4.1). DISFA dataset [8] was used in this experiment. This experiment was conducted to answer research questions

one and two. In this section, the experimental setup and experiment results are given as well as an objective comparison with another study.

6.2.1 Experimental Setup

Rotation invariant angular features are used (Section 5.2.2) for the first experiment. As it was explained in the Methodology chapter (Chapter 5), said features are extracted via selected feature points. The difference between the feature vectors of samples and the average neutral face is used as final feature set. For this experiment, the average neutral face was derived by averaging the neutral faces in the DISFA dataset (neutral by FACS AUs). For this experiment, FACS AU presence detection was separated from AU intensity detection.

For each AU, a binary classifier for detecting the presence of AUs was trained. For the binary classification, 1000 samples (50% present, 50% absent) that are normally distributed amongst subjects were selected for training. Remaining samples were used for testing (129815 samples). Therefore intensity level 0 detection was separated from the rest of the intensity estimation.

After the problem with FACS AU presence is tackled, the remaining part of the study is the detection of FACS AU intensities. For the intensity estimation given that AU is present, equal number of samples per AU intensity excluding 0 were selected. Said samples were selected to have approximately the same amount of samples per subject for training. The remaining samples were used for testing. For example, 1000 samples were randomly selected for AU1 (200 per intensity), and the remaining samples (7778 samples) were used for testing. The training and testing process was repeated ten times with different training and testing datasets.

Logistic regression is employed in this study. By definition, logistic regression gives results between the interval $[0, 1]$. Since this study is conducted to detect the intensities of FACS AUs, intensities are distributed amongst the interval $[0, 1]$. The distribution of the intensities are A$\rightarrow$ 0.1, B$\rightarrow$ 0.3, C$\rightarrow$ 0.5, D$\rightarrow$ 0.7, and D$\rightarrow$ 0.9. For the AU presence-absence detection, AU absent is labeled as 0 and present is labeled as 1.

The reason why this research was not done by leave-one-subject-out cross-validation technique is because the AUs are not normally divided into subjects in DISFA (Table 6.1 Table 6.2). For example, 60% frames containing AU2 is divided amongst 3 of the 27 subjects (subjects 4,16 and 29), where the remaining subjects show very little AU 2. The distribution is not consistent with intensities as well. For example, all the samples with AU25 intensity E is distributed amongst 7 of the 27 subjects (subjects 3,4,6,13,16,22,25),

where some of the subjects contain only A and B. Furthermore, the dataset contains only 27 subjects (Table 6.1), and FACS coding was done by just two certified FACS coders. With such bias and small amount of subjects present, a very small portion of the dataset was taken for training, and the testing was done on the remaining samples to show that the method is generalizable.

	AU Info		# Frames for each AU Intensity					
	AU#	#EV	0	1	2	3	4	5
Upper-Face	1	161	122036	2272	1749	2809	1393	555
	2	111	123450	1720	934	3505	836	369
	4	238	106220	4661	7636	6586	4328	1383
	5	100	128085	1579	719	293	104	34
	6	170	111330	9157	5986	3599	601	141
Lower-Face	9	73	123682	1659	2035	3045	316	77
	12	250	100020	13943	6869	7233	2577	172
	15	97	122952	5180	1618	1017	47	0
	17	271	107884	6342	4184	2281	112	11
	20	99	126282	1591	1608	1305	28	0
	25	296	84762	9805	13935	15693	5580	1039
	26	321	105838	13443	7473	3529	314	217

TABLE 6.1: The number of AU events (EV) and total number of frames for each intensity level given in DISFA [8].

Upper Face AUs	AU1	AU2	AU4	AU5	AU6
Present	8778	7346	24549	2729	19484
Absent	122036	123450	106220	128085	111330

Lower Face AUs	AU9	AU12	AU15	AU17	AU20	AU25	AU26
Present	7132	30794	7862	22930	4532	46052	24976
Absent	123682	100020	122952	107884	126282	84762	105838

TABLE 6.2: Number of present and absent frames for each AU in DISFA [8].

6.2.2 Experiment Results

Conducted experiment shows that proposed rotation invariant features are simple yet very successful for detecting both the presence and intensity of AUs. The system achieves 91.57% and 92.38% recognition rate for detecting the presence and absence of AUs respectively. The resultant recognition rates for binary classification are given in Table 6.3. Also it is observed that the system recognizes 72% of AU intensities given the AU is existent. This recognition rate is weighted average of intensity recognitions on 12 AUs

coded in DISFA dataset [8]. It is suggested that with a dataset that has more subjects and less bias, proposed method would be suitable for subject invariant testing. The average recognition rates for this research are given in the Table 6.4. To further the analysis of this method, confusion matrices for each AU are given in Table 6.5, and the corresponding visual representations of confusion matrices are given in Figure 6.1.

	AU1	AU2	AU4	AU5	AU6	AU9	AU12	AU15	AU17	AU20	AU25	AU26	**AVG**
TRUE +	90.17	95.86	91.45	97.94	91.48	95.36	91.46	88.35	87.45	87.45	92.65	89.42	**91.57**
TRUE -	92.64	95.91	91.31	95.06	93.35	95.39	90.77	91.75	90.11	90.11	94.35	87.78	**92.38**
FALSE +	9.83	4.14	8.55	2.06	8.52	4.64	8.54	11.65	12.55	12.55	7.35	10.58	**8.41**
FALSE -	7.36	4.09	8.69	4.94	6.65	4.61	9.23	8.25	9.89	9.89	5.65	12.22	**7.62**

TABLE 6.3: Experiment 1: Resultant recognition rates for binary classification of AUs.

		AU1	**AU2**	**AU4**	**AU5**	**AU6**	**AU9**	**AU12**	**AU15**	**AU17**	**AU20**	**AU25**	**AU26**	**AVG**
ACCURACY(%)		69.95	65.05	61.38	76.48	77.24	74.85	73.92	76.98	72.36	74.67	73.30	67.87	**72.00**
5-Level Intensity Accuracy (%)	**'A'**	72.15	74.48	56.78	77.61	81.60	76.57	78.45	76.48	69.33	76.14	81.33	67.70	**74.05**
	'B'	73.11	75.86	61.63	72.69	77.93	84.72	69.90	83.51	78.30	83.87	74.92	74.18	**75.89**
	'C'	62.47	52.75	65.52	81.91	63.39	67.48	66.35	69.16	69.82	62.44	62.42	54.42	**64.84**
	'D'	69.84	87.91	60.40	72.73	83.79	85.47	80.53	80.00	88.46	100.00	83.44	73.68	**80.52**
	'E'	99.15	98.82	57.82	75.00	100.00	91.67	95.45	-	100.00	-	92.73	100.00	**91.06**

TABLE 6.4: Experiment 1: Resultant recognition rates for intensity classification of AUs.

AU1	A	B	C	D	E
A	**72.15**	24.93	2.75	0.11	0.06
B	15.34	**73.11**	10.56	1	0
C	1.76	19.94	**62.47**	14.98	0.84
D	1.19	2.55	17.84	**69.84**	8.58
E	0	0	0	0.85	**99.15**

(a) AU1

AU2	A	B	C	D	E
A	**74.48**	22.46	2.16	0.22	0.67
B	11.13	**75.86**	12.23	0.31	0.47
C	1.73	35.61	**52.75**	7.63	2.29
D	0.94	1.41	7.54	**87.91**	2.2
E	0	0	0	1.18	**98.82**

(b) AU2

AU4	A	B	C	D	E
A	**56.78**	37.82	4.13	0.96	0.31
B	18.11	**61.63**	17.55	1.76	0.95
C	1.26	18.9	**65.52**	12.47	1.85
D	0.94	2.65	21.89	**60.4**	14.11
E	0	0.18	2.23	39.77	**57.82**

(c) AU4

AU5	A	B	C	D	E
A	**77.61**	21.58	0.81	0	0
B	10.47	**72.69**	16.84	0	0
C	0	12.77	**81.91**	5.32	0
D	0	0	9.09	**72.73**	18.18
E	0	0	0	25	**75**

(d) AU5

AU6	A	B	C	D	E
A	**81.6**	17.46	0.76	0.17	0.01
B	13.62	**77.93**	7.79	0.58	0.09
C	2.68	24.57	**63.39**	8.71	0.65
D	0.25	2.49	7.73	**83.79**	5.74
E	0	0	0	0	**100**

(e) AU6

AU9	A	B	C	D	E
A	**76.57**	22.25	1.02	0.16	0
B	8.36	**84.72**	3.91	0.96	2.05
C	0.85	27.12	**67.48**	3.97	0.59
D	0	0.85	6.84	**85.47**	6.84
E	0	0	0	8.33	**91.67**

(f) AU9

AU12	A	B	C	D	E
A	**78.45**	19.89	1.5	0.1	0.06
B	16.48	**69.9**	12.83	0.57	0.22
C	1.94	20.2	**66.35**	11.01	0.51
D	0.09	0.36	15.11	**80.53**	3.91
E	0	0	0	4.55	**95.45**

(g)AU12

AU15	A	B	C	D	E
A	**76.48**	21.37	1.86	0.08	0.21
B	11.34	**83.51**	4.7	0.45	0
C	0.49	26.56	**69.16**	3.79	0
D	0	0	20	**80**	0
E	-	-	-	-	**-**

(h) AU15

AU17	A	B	C	D	E
A	**69.33**	27.04	2.64	0.36	0.63
B	13.73	**78.3**	7.14	0.35	0.48
C	3.39	23.8	**69.82**	2.89	0.1
D	0	0	11.54	**88.46**	0
E	0	0	0	0	**100**

(i) AU17

AU20	A	B	C	D	E
A	**76.14**	22.54	0.96	0.35	0
B	11.91	**83.87**	3.65	0.57	0
C	1.49	26.65	**62.44**	8.67	0.75
D	0	0	0	**100**	0
E	-	-	-	-	**-**

(j) AU20

AU25	A	B	C	D	E
A	**81.33**	18.45	0.18	0.02	0.01
B	13.6	**74.92**	10.67	0.43	0.37
C	0.84	14.18	**62.42**	21.64	0.93
D	0.13	0.12	12.1	**83.44**	4.22
E	0	0	0.12	7.15	**92.73**

(k) AU25

AU26	A	B	C	D	E
A	**67.7**	30.21	1.7	0.28	0.12
B	12.13	**74.18**	13.16	0.5	0.03
C	2.29	37.78	**54.42**	5.48	0.03
D	0	0	21.93	**73.68**	4.39
E	0	0	0	0	**100**

(l) AU26

TABLE 6.5: Confusion matrices for all AUs.

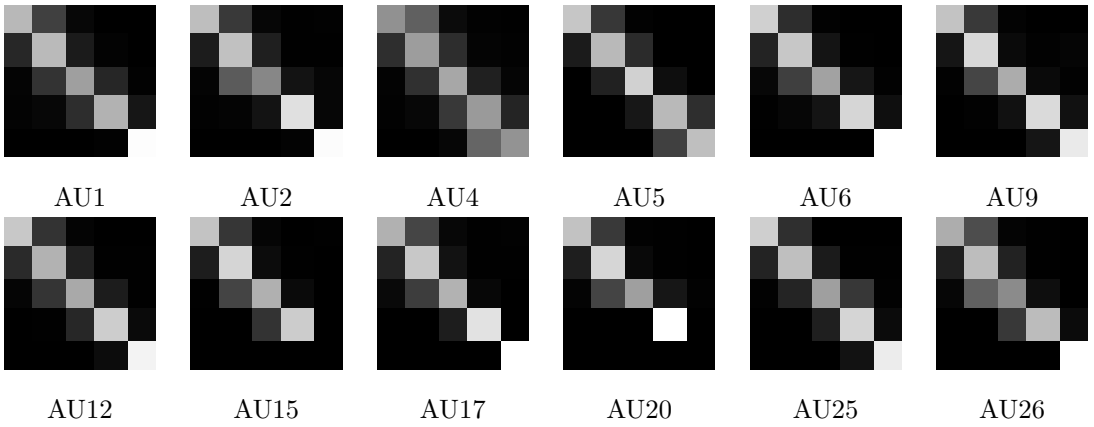


FIGURE 6.1: Visual representations of the confusion matrices given in Table 6.5.

6.2.3 Experiment Discussion

It is seen that vast majority of the time, AU intensities were confused with intensities neighboring them. Intensity C is the most confused, and intensity E is the most successfully recognized intensity. It was reported from the creators of DISFA dataset [8], ICC between certified FACS coders who coded DISFA ranged from 0.80 and 0.94. Even though 0.80 ICC is considered as high reliability, it shows that even certified FACS coders have disagreements. Therefore it is suggested by us that the received results are remarkable for even the certified FACS coders have disagreements between each other when differentiating between FACS AU intensities. Also the system was trained by mapping the intensities to $[0, 1]$ linearly. It is also suggested that the confusion between neighboring intensities would drop down if the mapping is done non-linearly.

For the first system angular features and logistic regression were used and average recognition rate of 92% for FACS AU presence-absence detection was received. Furthermore, 72% recognition success was achieved with FACS AU intensities. These results indicate that angular relations between facial feature points is an effective way of extracting features and logistic regression is a more than adequate classifier.

DISFA dataset [8] was made publicly available very recently. Therefore not many studies used this dataset for experimentation. To our knowledge, the study made by Mavadati et al. [8] (the publishers of DISFA) is the only one involving DISFA dataset. Therefore, a comparison between this study and theirs is necessary. The results received by Mavadati et al. is shown in Table 6.6.

When the methodology of this study is compared to the study conducted by Mavadati et al. [8], it is seen that there are two fundamental differences. Mavadati et al. did not separate FACS AU presence from intensity detection, and they used n-folds leave-one-subject out cross validation for testing. N-folds leave-one-subject out cross validation is one of

		AU1	AU2	AU4	AU5	AU6	AU9	AU12	AU15	AU17	AU20	AU25	AU26	AVG
Accuracy (%)		86.24	91.28	80.94	94.46	83.98	92.32	79.67	91.35	80.66	88.12	79.85	79.61	**85.71**
6-Level Intensity Accuracy (%)	**'0'**	88.06	92.45	84.36	95.49	87.69	93.61	84.55	93.12	82.71	89.02	82.11	82.52	**87.98**
	'A'	52.55	61.83	57.35	36.20	58.16	75.06	63.65	62.81	56.63	74.95	70.11	63.11	**61.03**
	'B'	77.74	66.26	69.10	66.03	65.69	64.62	68.97	59.83	69.93	55.53	76.25	68.03	**67.33**
	'C'	61.92	86.20	70.93	54.64	77.83	74.39	64.09	82.07	65.71	59.57	73.81	79.08	**70.85**
	'D'	64.49	56.98	64.82	53.54	59.57	60.26	59.72	00.00	46.43	07.14	85.97	85.07	**53.65**
	'E'	31.17	28.13	59.58	55.88	00.00	15.52	00.00	-	00.00	-	96.20	93.60	**38.01**

TABLE 6.6: Results received by Mavadati et al. [8].

the commonly used methods for testing because it shows that the suggested method is generalizable to any subject. As it was mentioned in the Section 6.2.1, the reason why n-folds leave-one-subject out cross validation was not used in this experiment is because most of the dataset is biased and that could lead to misleading results. Yet, to confirm that the suggested method is generalizable, only a very small portion of the dataset was used for training and training-testing process was repeated ten times. Secondly, in this study, FACS AU presence detection was separated from intensity detection. therefore the raw results are not comparable to the results received by Mavadati et al. [8]. Yet, this issue can easily be tackled by simple analysis of FACS AU presence and FACS AU intensity detection results. When the suggested system detects FACS AU presence first, then detects intensity, simple multiplication will give comparable results. When the analysis is performed, it is seen that this study results in average 66.51% recognition rate. Also, the result for 6-level intensity detection becomes 92.38%, 68.41%, 70.11%, 59.90%, 74.38% and 84.12 for each level of 6-level intensity respectively.

Mavadati et al. [8] conducted their study to compare three different feature extraction methods which are LBPH, HOG and Gabor wavelets with SVM classification. Amongst the three feature extraction methods, the most successful results were received by Gabor wavelets. Therefore the Gabor wavelet results received by Mavadati et al. are compared to the results of this study. The average 6-level intensity detection results of each study is given in Table 6.7. When the results from this study is compared to the study conducted by Mavadati et al., it is seen that the suggested system receives higher classification accuracy compared to Mavadati et al. [8]. It is seen that the suggested system classifies FACS AU absence and intensities 'A' and 'B' slightly better than Mavadati et al. [8]. With intensities 'D' and 'E', there is a remarkable gap between our results and Mavadati et al. [8], yet the results reported by Mavadati et al. suggest that their system is a better classifier for detecting intensity 'C'.

	Absent	**'A'**	**'B'**	**'C'**	**'D'**	**'E'**
Our Results	92.38	68.41	70.11	59.90	74.38	84.12
Mavadati et al. [8]	87.98	61.03	67.33	70.85	53.65	38.01

TABLE 6.7: The average 6-level intensity detection results from this study and Mavadati et al. [8].

From the Table 6.6, it is seen that Mavadati et al. received very low recognition rates for the detection of higher intensities ('D' and 'E') for FACS AUs AU1, AU2, AU6, AU9, AU12, AU15, AU17 and AU20. They received 53% to 95% recognition rates with the rest of the AUs. From the tables 6.1 and 6.2, it is seen that DISFA dataset contains very few amounts of frames for higher intensities for FACS AUs AU1, AU2, AU6, AU9, AU12, AU15, AU17 and AU20. It was mentioned before that Mavadati et al. used n-folds leave-one-subject-out cross validation for testing. Therefore it is possible that the lack of data for training when a subject is left out could have caused the low recognition rates. Avoiding lack of data problems was the reason why n-folds leave-one-subject-out cross validation was not employed in this experiment for testing. If their experiments were conducted on a different dataset that has larger variety of samples, they could have received higher recognition rates. Therefore the received results and the results from Mavadati et al. [8] are not comparable.

Despite the situation with higher intensities, both the suggested system and the system suggested by Mavadati et al. had enough samples for training and testing with the rest of the intensities. Therefore analyzing the results of both systems can provide valid comparison. From the tables 6.5, 6.3, 6.4, 6.6 and 6.7, it can be concluded that the suggested system receives higher classification rates compared to Mavadati et al. [8] when detecting FACS AU presence/absence and intensities 'A' and 'B'. Yet, Mavadati et al. [8] outperforms the suggested system by more than 10% when detecting intensity 'C'. Therefore it can be concluded that the suggested system is a more robust system compared to the system suggested by Mavadati et al. [8] in a general scale.

Finally it can be concluded that angular relations between facial feature points is one of the effective ways of extracting features for the analysis of facial actions. Therefore the first and second research questions are partially answered. Since the only classification method used in this experiment is logistic regression, the third research question is yet to be answered.

6.3 Pain Intensity Detection

The association of AU4, AU6, AU7, AU9, AU10, and AU43 to pain and PSPI scale [20] [21] was mentioned in Section 2.4. After the remarkable results received from first experiment (Section 6.2), the suggested method was improved and applied to pain intensity detection by PSPI scale. The first experiment was conducted solely for on the detection of FACS AUs, and only two of the research questions were partially answered. The second experiment, however, is aimed to answer all the research questions completely. In

this section, the experimental setup and experiment results for pain intensity detection are explained and shown. Also an objective comparison with other studies are given.

6.3.1 Experimental Setup

In the previous experiment, it was shown that angular features with logistic regression classifies FACS AUs very successfully. This experiment was conducted to extend the previous experiment. Therefore, the experiment is divided into two sets of experiments. The first one is to detect which feature set and which classification method is better for pain intensity detection from only frontal facial images (images with out of plane head rotation are pruned). The second one is to apply the selected method to the whole UNBC McMasters dataset including images with out of plane head rotation.

UNBC McMasters dataset [7] is coded for PSPI [21] pain scale, and the PSPI pain level of painful images was mapped down to four intensities 0 (PSPI 0), 1 (PSPI 1-5), 2 (PSPI 6-10), 3 (PSPI 11-15) for experimentation. The techniques used for pain intensity detection experiment consist of angular (Section 5.2.2) and vectorial (Section 5.2.3) features classified by logistic regression (Section 5.4.1) and FFNNs (Section 5.4.2). The neutral face that is necessary for feature extraction was derived differently for each part of the experiment. Average neutral face of each subject was used in his/her own face for the first part of the experiment and a general average neutral face was used for the second part.

Another aspect of this experiment is to select relevant features from the whole feature set which was done by genetic algorithms (Section 5.3). Also as mentioned in the previous experiment, non-linear mapping of pain intensities to $[0, 1]$ would promise better results, and to achieve that, a brute force algorithm was applied. For each level of pain, a parameter is iterated between 0 and 1 with a step of .05, and for each level, a small set of subjects are used for training and testing the logistic regression function with the tested threshold values. Finally, the threshold values that give the most successful results for classification were selected for experimentation.

Method #	Description
1	Angular features, feature selection by Neural Network classification
2	Vectorial features, feature selection by Neural Network classification
3	Angular, and vectorial features, feature selection by Neural Network classification
4	Angular features, feature selection by Logistic Regression classification
5	Vectorial features, feature selection by Logistic Regression classification
6	Angular, and vectorial Features, feature selection by Logistic Regression classification

TABLE 6.8: Experiment methods for pain intensity detection.

The experimentation was done by leave one subject out method, where out of 23 remaining subjects after out of plane head movement images are pruned, 22 is used for training the system, and 1 is used for testing.

6.3.2 Experiment Results

Given the results of our study, it is clear that suggested angular features are better than vectorial features, but the combination of both is the one that gives the best results. From the Table 6.9, it is seen that each method has advantages and disadvantages when classifying pain. With method 6 , it is seen that high levels of pain are classified most successfully, but level 2 pain is misclassified in almost every occasion, and misclassified frames are not only confused with neighboring intensities. With method 2, the classification success is opposite to method 6. From this experiment, the most successful classification results were received with methods 3 and 6, yet method 3 seems the more stable one. The reason behind this conclusion is that amongst intensities 0, 1, and 2, classification is done successfully with method 3s. Only with intensity 3, there is a classification problem, and almost all of misclassification happened with intensity 2 which is the direct neighbor of 3. It is suggested that this happened because there are not many frames with high level of pain in the dataset, and the difference between level 3 frames and level 2 frames are almost indistinguishable by human eye. One other reason why method 6 was not selected is because method 6 has a major issue with classifying level 2. Only 7% of label 2 was successfully classified. If the misclassifications were only happening with one of the neighboring intensities (like it happened with method 3), then method 6 could be considered a good option, but misclassifications happened with 27%, 26% and 39% with intensities 0, 1 and 3 respectively.

Method 1	0	1	2	3
1	**0.28**	0.65	0.04	0.03
2	0.14	**0.72**	0.10	0.04
3	0.30	0.43	**0.08**	0.19
4	0.27	0.52	0.00	**0.21**

1

Method 2	0	1	2	3
1	**0.46**	0.25	0.22	0.08
2	0.22	**0.36**	0.38	0.04
3	0.00	0.17	**0.70**	0.13
4	0.00	0.71	0.00	**0.29**

2

Method 3	0	1	2	3
1	**0.44**	0.41	0.07	0.08
2	0.23	**0.60**	0.12	0.05
3	0.23	0.11	**0.62**	0.05
4	0.05	0.00	0.73	**0.23**

3

Method 4	0	1	2	3
1	**0.60**	0.32	0.01	0.07
2	0.42	**0.49**	0.02	0.07
3	0.48	0.12	**0.19**	0.21
4	0.81	0.00	0.00	**0.19**

4

Method 5	0	1	2	3
1	**0.53**	0.46	0.01	0.01
2	0.08	**0.83**	0.09	0.00
3	0.00	0.62	**0.36**	0.02
4	0.47	0.01	0.29	**0.23**

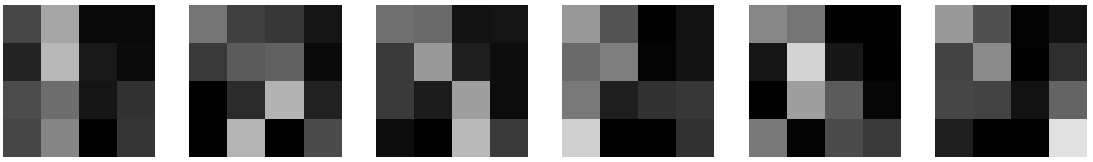
5

Method 6	0	1	2	3
1	**0.60**	0.31	0.01	0.07
2	0.26	**0.55**	0.01	0.18
3	0.27	0.26	**0.07**	0.39
4	0.12	0.00	0.00	**0.88**

6

TABLE 6.9: Confusion matrix results of six different feature extraction methods given in Table 6.8.

In the previous part of the experiment, it was concluded that neural networks with the combination of angular and vectorial features give the optimal results. Therefore method 3 was selected for the final set of experiments. For the second part of the



Method 1 | Method 2 | Method 3 | Method 4 | Method 5 | Method 6

FIGURE 6.2: Visual representations of the confusion matrices given in Table 6.9.

experimentation, the whole dataset including out of plane head rotations were used. Also instead of subjective neutral faces, a static average neutral face was used for feature extraction. The final experiments were binary classification (pain vs no pain), four level pain intensity detection (no pain, low pain, mediocre pain, and high pain) and three level pain intensity detection (low pain, mediocre pain, and high pain). Experimental results are given in Table 6.10, Table 6.11, and Table 6.12.

True Positive	78%
True Negative	80%
False Positive	22%
False Negative	20%

TABLE 6.10: Results of binary classification.

	0	1	2	3
0	**0.53**	0.34	0.06	0.07
1	0.35	**0.52**	0.08	0.05
2	0.21	0.26	**0.41**	0.12
3	0.10	0.07	0.61	**0.22**

TABLE 6.11: Results of four level pain intensity detection.

	1	2	3
1	**0.78**	0.14	0.08
2	0.10	**0.75**	0.15
3	0.05	0.35	**0.60**

TABLE 6.12: Results of three level pain intensity detection.

6.3.3 Experiment Discussion

From the Section 6.3.2, it is observed that pain levels are most generally confused with their neighboring intensities. This is due to the fact that the PSPI pain scale is distributed to four pain levels. For example, the PSPI pain level 6, which falls within the boundary of level 2 in our scale, is quite similar to PSPI level 5, which falls within the boundary of 1. Also it can be seen from the Figure 6.3, even the difference between



FIGURE 6.3: Some sequence of images from the UNBC-McMaster Database [7] with their PSPI pain scale labels.

images with very high stages of PSPI and lowest stage of PSPI is almost indistinguishable to human eye. Despite that, the final set of experiments show that the selected method gives remarkable results, especially with binary classification and three level pain intensity detection. From the Table 6.10 it is seen that the true positive and true negative rates of the suggested system are 78% and 80% respectively. From Table 6.12 it is seen that the classification success rates are 78% (low pain), 75% (mediocre pain) and 60% (high pain). Also the average recognition rate of the three level pain detection method is 77.70%. Yet, from Table 6.11, it is seen that the four level intensity detection method still confuses intensity 3 with intensity 2 very much. As it stands, four level pain intensity detection successfully classified each level of pain with 53% (no pain), 52% (low pain), 41% (mediocre pain), and 22% (high pain) recognition rates. Yet, if pain detection is separated from pain intensity detection the same way it was employed in experiment 1 (Section 6.2), the received results become 79% (no pain), 62% (low pain), 60% (mediocre pain), and 48% (high pain).

The first comparison is made with the study conducted by Lucey et al. [70]. Lucey et al. used S-PTS, S-APP, and C-APP as features and SVMs to classify between pain and no pain using the frames from UNBC McMasters dataset [7]. All possible combinations of said geometric features were compared in their study. Their results by means of area under the ROC curve are given in Figure 6.4. They received the most successful results with the combination S-PTS + S-APP + C-APP. The highest results received was 83%. From Table 6.10 it is seen that the suggested system classifies the presence of pain very successfully. From the Table 6.10 and Figure 6.4 it is seen that the suggested method performs similar to the study conducted by Lucey et al.. Yet, the study conducted by Lucey et al. [16] received higher recognition rate compared to the suggested method by

4%. The suggested method only slightly under-perform Lucey et al. when detecting the presence of pain [16]. Therefore it can be concluded that the suggested pain detection method is still a successful one.

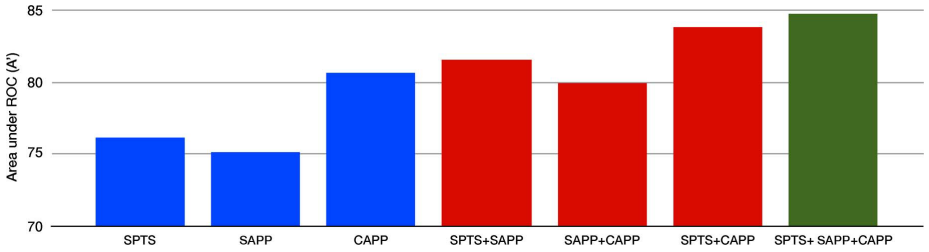


FIGURE 6.4: The results of Lucey et al. [16] by means of area under the ROC curve.

Since the comparison of pain vs no pain aspect of this experiment is already done, the only part left to compare is the intensity detection comparison. The second comparison is made by the work of Bousmalis et al. [18]. In their research, Bousmalis et al. focused on human behavior analysis for both pain and agreement. Their experimentation on pain is the only one that can be compared with ours due to the fact that they also used UNBC McMasters dataset [7], and n-folds cross validation method was applied by both this study and by Bousmalis et al. [18]. They had two different sets of experiments with UNBC McMasters database [7]. Firstly, they classified between no pain and high level of pain (Pain 1). Secondly, they divided pain into three levels which are low, mediocre, and high levels of pain (Pain 2). They used multiple neural network derivations for classification which are random fields, HCRF, and iHCRF. They showed that iHCRF is the more successful classifier amongst the three. Their research results are given in Table 6.13.

Method	**Pain 1**	**Pain 2**
Random	50%	Below 40%
HCRF	83.9%	53.4%
iHCRF	89.4%	57.7%

TABLE 6.13: Pain classification results from Bousmalis et al. [18]. Pain 1: No pain vs high pain. Pain 2: Detection of low, mediocre and high levels of pain.

Pain 1 (no pain vs high pain) experiment conducted by Bousmalis et al. [18] is not applicable to this study because this study involved all the levels of pain when classifying the presence of pain and Bousmalis et al. [18] used only high levels of pain. The only comparable results from Bousmalis et al. [18] is the Pain 2 (three level pain classification) experiment. From Table 6.12, it is seen that the suggested method classifies three levels of pain very successfully. Received recognition rates are 78%, 75%, and 60% for low, mediocre, and high levels of pain respectively. Also the average pain level classification with three levels of pain is 77.7% which is higher than what Bousmalis et al. [18] received

by 20%. Therefore it can be concluded that the suggested system is a remarkable method.

Finally it can be concluded that the combination of angular and vectorial features is the more preferable option compared to using either one of them when detecting pain intensity. Also from the results of the experiments, it can be said that using neural networks is a preferable option to logistic regression.

6.4 Summary

In this chapter, the two sets of experiments conducted and their results are explained and discussed. The first experiment was done to detect FACS AU intensities, and the second one was done to detect pain intensity by PSPI pain scale. The results in this chapter show that the suggested methods are quite effective to detect the intensity of facial actions. In Chapter 7, the summary and conclusion of the thesis is conducted.

Chapter 7

Thesis Summary and Conclusion

7.1 Summary

A novel facial action intensity detection system that uses geometric features and supervised machine learning is suggested and discussed in this thesis. A brief introduction was given in Chapter 1 as well as a detailed explanation of motivation and research questions for this thesis. In Chapter 2, the fundamentals of automated FER was discussed. Computational background and literature survey related to automated FER were explained in Chapters 3 and 4. Methodology of the suggested system in detail was given in Chapter 5. In Chapter 6, the experimental setups of both experiments, the results received from said experiments were given. Chapter 6 also included objective comparisons between recent studies and this one.

7.2 Research Questions and Answers

This study was conducted to answer these research questions:

1. How can the facial feature points and facial grid be used effectively to analyze the intensities of facial actions?
2. What type of geometric relations between facial feature points and facial grid would be more efficient and successful for facial action intensity detection?
3. Is logistic regression analysis technique or FFNN more effective for classifying facial actions and intensities?

The first and second questions were partially answered by the first experiment. It was concluded that angular relations between facial feature points is an effective way of using facial feature points and facial grid to analyze human facial actions. Also, the first experiment showed that angular features are efficient and successful for facial action intensity detection (Section 6.2). These conclusions were derived after a detailed comparison between this study and the study conducted by Mavadati et al. [8].

After receiving remarkable results from the first experiment, the study was extended to use vectorial features, genetic algorithms and neural networks as well. The classification system of the second experiment was also an extension from the first one. In the second experiment, PSPI scale, which is derived from the combinations of AU4, AU6, AU7, AU9, AU10, and AU43, was used.

The second experiment was conducted to give the most precise answers to all of the research questions. The second experiment showed that the combination of vectorial and angular features are the most effective and successful way of using facial feature points and facial grid for facial action intensity detection. The second experiment also showed that neural networks are slightly better classification methods for facial action intensity detection (Section 6.3). These conclusions were derived by detailed experimentation and objective comparison between two of the most recent studies involving pain.

As explained above, deriving geometric features from facial feature points and facial grid is one of the effective ways of using facial feature points and facial grid for facial action intensity analysis. Deriving angular and vectorial relations from facial feature points and facial grid are efficient and successful ways of feature extraction for facial action intensity detection. Finally, FFNN is a slightly better classifier than logistic regression for facial action intensity detection. Therefore it can be concluded that all three of the research questions were successfully answered.

7.3 Shortcoming and Challenges

For both of the suggested systems, one possible challenge would be the facial feature point detection. Both of the systems use geometric features derived from the facial feature points and the detection of said points would affect the robustness of the system greatly. Also after the detection of facial feature points the tracking of said points would be a challenge if one tries to apply the suggested system to a real time application. These two challenges could be overcome by detecting facial feature points in the initial frame of the real time stream and tracking the points using SIFT [27] or SURF [28].

7.4 Future Work

Research questions in this study were successfully answered. Yet it can not be concluded that there is no room for improvement. For one, only geometric features were used in this study. A possible improvement could be using appearance based or a combination of appearance and geometry based feature extraction method. This could potentially improve the results or perhaps lead for a definite answer. Also, nonlinear mapping of FACS AU intensities could be employed with logistic regression as it was done with pain intensity. Perhaps genetic algorithm could be used for feature selection as well as neural networks for FACS AU intensity detection. Therefore, from the received results it can be concluded that this study is successful for classifying intensities of facial actions, but there is a lot left to be exploited.

7.5 Conclusion

In this thesis, a novel facial action intensity detection system was explained, and as can be seen from the received results, it is concluded that facial action intensity detection is achievable via geometric features and supervised machine learning methods. The research questions given in Section 1.2 were successfully answered. Finally, the hypothesis "An effective and fast facial action intensity detection system can be created by extracting geometric facial features and classifying by supervised machine learning methods." is proven to be correct.

Bibliography

[1] Deniz Bingöl and Turgay Celik and Christian Omlin and Hima Vadapalli, "Facial action unit intensity estimation using rotation invariant features and regression analysis," in *IEEE International Conference on Image Processing 2014 (ICIP 2014)*, (Paris, France), Oct. 2014.

[2] Ekman, Paul and Friesen, Wallace V, "Manual for the facial action coding system," 1977.

[3] Lyons, Michael and Akamatsu, Shigeru and Kamachi, Miyuki and Gyoba, Jiro, "Coding facial expressions with gabor wavelets," in *Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on*, pp. 200–205, IEEE, 1998.

[4] Lucey, Patrick and Cohn, Jeffrey F and Kanade, Takeo and Saragih, Jason and Ambadar, Zara and Matthews, Iain, "The extended Cohn-Kanade dataset (CK+): A complete dataset for action unit and emotion-specified expression," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, pp. 94–101, IEEE, 2010.

[5] Yin, Lijun and Wei, Xiaozhou and Sun, Yi and Wang, Jun and Rosato, Matthew J, "A 3D facial expression database for facial behavior research," in *Automatic face and gesture recognition, 2006. FGR 2006. 7th international conference on*, pp. 211–216, IEEE, 2006.

[6] Savran, Arman and Alyüz, Ne¸se and Dibeklio˘glu, Hamdi and Çeliktutan, Oya and Gökberk, Berk and Sankur, Bülent and Akarun, Lale, "Bosphorus database for 3D face analysis," in *Biometrics and Identity Management*, pp. 47–56, Springer, 2008.

[7] Lucey, Patrick and Cohn, Jeffrey F and Prkachin, Kenneth M and Solomon, Patricia E and Matthews, Iain, "Painful data: The UNBC-McMaster shoulder pain expression archive database," in *Automatic Face & Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on*, pp. 57–64, IEEE, 2011.

[8] Mavadati, S and Mahoor, M and Bartlett, Kevin and Trinh, Philip and Cohn, J, "DISFA: A Spontaneous Facial Action Intensity Database," *Affective Computing, IEEE Transactions on*, vol. 4, no. 2, pp. 151–160, 2013.

[9] Tzimiropoulos, Georgios and Alabort-i-Medina, Joan and Zafeiriou, Stefanos and Pantic, Maja, "Generic active appearance models revisited," in *Computer Vision–ACCV 2012*, pp. 650–663, Springer, 2013.

[10] Maalej, Ahmed and Ben Amor, B and Daoudi, Mohamed and Srivastava, Anuj and Berretti, Stefano, "Local 3D shape analysis for facial expression recognition," in *Pattern Recognition (ICPR), 2010 20th International Conference on*, pp. 4129–4132, IEEE, 2010.

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

[12] Whitehill, Jacob and Omlin, Christian W, "Haar features for facs au recognition," in *Automatic Face and Gesture Recognition, 2006. FGR 2006. 7th International Conference on*, pp. 5–pp, IEEE, 2006.

[13] Shan, Caifeng and Gong, Shaogang and McOwan, Peter W, "Facial expression recognition based on Local Binary Patterns: A comprehensive study," *Image and Vision Computing*, vol. 27, no. 6, pp. 803–816, 2009.

[14] Chen, Jingying and Chen, Dan and Gong, Yujiao and Yu, Meng and Zhang, Kun and Wang, Lizhe, "Facial expression recognition using geometric and appearance features," in *Proceedings of the 4th International Conference on Internet Multimedia Computing and Service*, pp. 29–33, ACM, 2012.

[15] Jeni, László A and Girard, Jeffrey M and Cohn, Jeffrey F and De La Torre, Fernando, "Continuous AU intensity estimation using localized, sparse facial feature space," in *2nd International Workshop on Emotion Representation, Analysis and Synthesis in Continuous Time and Space (EmoSPACE)*, vol. 6, 2013.

[16] Lucey, Patrick and Cohn, Jeffrey F and Matthews, Iain and Lucey, Simon and Sridharan, Sridha and Howlett, Jessica and Prkachin, Kenneth M, "Automatically detecting pain in video through facial action units," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 41, no. 3, pp. 664–674, 2011.

[17] Ekman, Paul and Friesen, Wallace V and Hager, Joseph C, *Facial action coding system.* A Human Face Salt Lake City, 2002.

[18] Bousmalis, Konstantinos and Zafeiriou, Stefanos and Morency, L and Pantic, Maja, "Infinite hidden conditional random fields for human behavior analysis," *Neural*

Networks and Learning Systems, IEEE Transactions on, vol. 24, no. 1, pp. 170–177, 2013.

[19] Ekman, Paul and Friesen, Wallace V, "Facial action coding system: A technique for the measurement of facial movement. Palo Alto," *CA: Consulting Psychologists Press. Ellsworth, PC, & Smith, CA (1988). From appraisal to emotion: Differences among unpleasant feelings. Motivation and Emotion*, vol. 12, pp. 271–302, 1978.

[20] Prkachin, Kenneth M, "The consistency of facial expressions of pain: a comparison across modalities," *Pain*, vol. 51, no. 3, pp. 297–306, 1992.

[21] Prkachin, Kenneth M and Solomon, Patricia E, "The structure, reliability and validity of pain expression: Evidence from patients with shoulder pain," *Pain*, vol. 139, no. 2, pp. 267–274, 2008.

[22] Izard, CE, "revised, The Maximally Discriminative Facial Movement Coding System," 1983.

[23] Ruiz, José V and Moreno, Belén and Pantrigo, Juan José and Sánchez, Ángel, "Comparing Feature Point Tracking with Dense Flow Tracking for Facial Expression Recognition," in *Bioinspired Applications in Artificial and Natural Computation*, pp. 264–273, Springer, 2009.

[24] Ekman, P and Friesen, W, "Rationale and reliability for EMFACS Coders," *Unpublished manuscript*, 1982.

[25] Gonzalez, RC and Woods, RE, "Digital Image Processing 3 Edition Prentice-Hall," 2008.

[26] Viola, Paul and Jones, Michael, "Rapid object detection using a boosted cascade of simple features," in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 1, pp. I–511, IEEE, 2001.

[27] Lowe, David G, "Object recognition from local scale-invariant features," in *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, vol. 2, pp. 1150–1157, Ieee, 1999.

[28] Bay, Herbert and Tuytelaars, Tinne and Van Gool, Luc, "SURF: Speeded up robust features," in *Computer Vision–ECCV 2006*, pp. 404–417, Springer, 2006.

[29] Burges, Christopher JC, "A tutorial on support vector machines for pattern recognition," *Data mining and knowledge discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[30] Freund, Yoav and Schapire, Robert and Abe, N, "A short introduction to boosting," *Journal-Japanese Society For Artificial Intelligence*, vol. 14, no. 771-780, p. 1612, 1999.

[31] Friedman, Jerome and Hastie, Trevor and Tibshirani, Robert and others, "Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors)," *The annals of statistics*, vol. 28, no. 2, pp. 337–407, 2000.

[32] Schapire, Robert E and Singer, Yoram, "Improved boosting algorithms using confidence-rated predictions," *Machine learning*, vol. 37, no. 3, pp. 297–336, 1999.

[33] Mitchell, Melanie, *An introduction to genetic algorithms*. MIT press, 1998.

[34] Snyman, Jan, *Practical mathematical optimization: an introduction to basic optimization theory and classical and new gradient-based algorithms*, vol. 97. Springer, 2005.

[35] Seber, George AF and Lee, Alan J, *Linear regression analysis*, vol. 936. John Wiley & Sons, 2012.

[36] Hosmer Jr, David W and Lemeshow, Stanley and Sturdivant, Rodney X, *Applied logistic regression*. Wiley. com, 2013.

[37] Jain, Anil K and Mao, Jianchang and Mohiuddin, K Moidin, "Artificial neural networks: A tutorial," *Computer*, vol. 29, no. 3, pp. 31–44, 1996.

[38] Darwin, Charles, *The expression of the emotions in man and animals*. Oxford University Press, 1998.

[39] Kanade, Takeo and Cohn, Jeffrey F and Tian, Yingli, "Comprehensive database for facial expression analysis," in *Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on*, pp. 46–53, IEEE, 2000.

[40] Tong, Yan and Liao, Wenhui and Ji, Qiang, "Facial action unit recognition by exploiting their dynamic and semantic relationships," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, no. 10, pp. 1683–1699, 2007.

[41] Lienhart, Rainer and Maydt, Jochen, "An extended set of haar-like features for rapid object detection," in *Image Processing. 2002. Proceedings. 2002 International Conference on*, vol. 1, pp. I–900, IEEE, 2002.

[42] Mita, Takeshi and Kaneko, Toshimitsu and Hori, Osamu, "Joint haar-like features for face detection," in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, vol. 2, pp. 1619–1626, IEEE, 2005.

[43] Geetha, A and Ramalingam, Vennila and Palanivel, S and Palaniappan, B, "Facial expression recognition–A real time approach," *Expert Systems with Applications*, vol. 36, no. 1, pp. 303–308, 2009.

[44] Zhao, Xiaowei and Chai, Xiujuan and Niu, Zhiheng and Heng, Cherkeng and Shan, Shiguang, "Context modeling for facial landmark detection based on Non-Adjacent Rectangle (NAR) Haar-like feature," *Image and Vision Computing*, vol. 30, no. 3, pp. 136–146, 2012.

[45] Cristinacce, David and Cootes, Tim, "Automatic feature localisation with constrained local models," *Pattern Recognition*, vol. 41, no. 10, pp. 3054–3067, 2008.

[46] Saragih, Jason and Goecke, Roland, "A nonlinear discriminative approach to AAM fitting," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pp. 1–8, IEEE, 2007.

[47] Saragih, Jason and Göcke, Roland, "Learning AAM fitting through simulation," *Pattern Recognition*, vol. 42, no. 11, pp. 2628–2636, 2009.

[48] Tsalakanidou, Filareti and Malassiotis, Sotiris, "Real-time 2D+3D facial action and expression recognition," *Pattern Recognition*, vol. 43, no. 5, pp. 1763–1775, 2010.

[49] Valstar, Michel and Martinez, Brais and Binefa, Xavier and Pantic, Maja, "Facial point detection using boosted regression and graph models," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pp. 2729–2736, IEEE, 2010.

[50] Zhu, Xiangxin and Ramanan, Deva, "Face detection, pose estimation, and landmark localization in the wild," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pp. 2879–2886, IEEE, 2012.

[51] Tang, Hao and Huang, Thomas S, "3D facial expression recognition based on automatically selected features," in *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on*, pp. 1–8, IEEE, 2008.

[52] Maalej, Ahmed and Amor, Boulbaba Ben and Daoudi, Mohamed and Srivastava, Anuj and Berretti, Stefano, "Shape analysis of local facial patches for 3D facial expression recognition," *Pattern Recognition*, vol. 44, no. 8, pp. 1581–1589, 2011.

[53] Mpiperis, Iordanis and Malassiotis, Sotiris and Strintzis, Michael G, "Bilinear models for 3-D face and facial expression recognition," *Information Forensics and Security, IEEE Transactions on*, vol. 3, no. 3, pp. 498–511, 2008.

[54] Li, Xiaoxing and Jia, Tao and Zhang, Hao, "Expression-insensitive 3D face recognition using sparse representation," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 2575–2582, IEEE, 2009.

[55] Segundo, Maurício Pamplona and Silva, Luciano and Bellon, Olga Regina Pereira and Queirolo, Chauã C, "Automatic face segmentation and facial landmark detection in range images," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 40, no. 5, pp. 1319–1330, 2010.

[56] Kotsia, Irene and Buciu, Ioan and Pitas, Ioannis, "An analysis of facial expression recognition under partial facial image occlusion," *Image and Vision Computing*, vol. 26, no. 7, pp. 1052–1067, 2008.

[57] Wu, Tingfan and Bartlett, Marian S and Movellan, Javier R, "Facial expression recognition using gabor motion energy filters," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, pp. 42–47, IEEE, 2010.

[58] Gu, Wenfei and Xiang, Cheng and Venkatesh, YV and Huang, Dong and Lin, Hai, "Facial expression recognition using radial encoding of local Gabor features and classifier synthesis," *Pattern Recognition*, vol. 45, no. 1, pp. 80–91, 2012.

[59] Wang, Yubo and Ai, Haizhou and Wu, Bo and Huang, Chang, "Real time facial expression recognition with adaboost," in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, vol. 3, pp. 926–929, IEEE, 2004.

[60] Yang, Peng and Liu, Qingshan and Metaxas, Dimitris N, "Boosting coded dynamic features for facial action units and facial expression recognition," in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pp. 1–6, IEEE, 2007.

[61] Yang, Peng and Liu, Qingshan and Cui, Xinyi and Metaxas, Dimitris N, "Facial expression recognition using encoded dynamic features," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pp. 1–8, IEEE, 2008.

[62] Zhao, Guoying and Pietikainen, Matti, "Dynamic texture recognition using local binary patterns with an application to facial expressions," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, no. 6, pp. 915–928, 2007.

[63] Gritti, Tommaso and Shan, Caifeng and Jeanne, Vincent and Braspenning, Ralph, "Local features based facial expression recognition with face registration errors," in *Automatic Face & Gesture Recognition, 2008. FG'08. 8th IEEE International Conference on*, pp. 1–8, IEEE, 2008.

[64] Jiang, Bihan and Valstar, Michel Fran¸cois and Pantic, Maja, "Action unit detection using sparse appearance descriptors in space-time video volumes," in *Automatic Face & Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on*, pp. 314–321, IEEE, 2011.

[65] van der Maaten, Laurens and Hendriks, Emile, "Action unit classification using active appearance models and conditional random fields," *Cognitive processing*, vol. 13, no. 2, pp. 507–518, 2012.

[66] Fasel, Beat and Luettin, Juergen, "Recognition of asymmetric facial action unit activities and intensities," in *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, vol. 1, pp. 1100–1103, IEEE, 2000.

[67] Marian Bartlett and Gwen Littlewort and Mark Frank and Claudia Lainscsek and Ian Fasel and Javier Movellan, "Automatic Recognition of Facial Actions in Spontaneous Expressions," *Journal of Multimedia*, vol. 1, no. 6, 2006.

[68] Mahoor, Mohammad H and Cadavid, Steven and Messinger, Daniel S and Cohn, Jeffrey F, "A framework for automated measurement of the intensity of non-posed facial action units," in *Computer Vision and Pattern Recognition Workshops, 2009. CVPR Workshops 2009. IEEE Computer Society Conference on*, pp. 74–80, IEEE, 2009.

[69] Savran, Arman and Sankur, Bulent and Taha Bilge, M, "Regression-based intensity estimation of facial action units," *Image and Vision Computing*, vol. 30, no. 10, pp. 774–784, 2012.

[70] Lucey, Patrick and Cohn, Jeffrey F and Prkachin, Kenneth M and Solomon, Patricia E and Chew, Sien and Matthews, Iain, "Painful monitoring: Automatic pain monitoring using the UNBC-McMaster shoulder pain expression archive database," *Image and Vision Computing*, vol. 30, no. 3, pp. 197–205, 2012.

[71] Chen, Jixu and Liu, Xiaoming and Tu, Peter and Aragones, Amy, "Learning person-specific models for facial expression and action unit recognition," *Pattern Recognition Letters*, vol. 34, no. 15, pp. 1964–1970, 2013.

[72] Hill, Marilyn L and Craig, Kenneth D, "Detecting deception in pain expressions: The structure of genuine and deceptive facial displays," *Pain*, vol. 98, no. 1, pp. 135–144, 2002.

[73] Larochette, Anne-Claire and Chambers, Christine T and Craig, Kenneth D, "Genuine, suppressed and faked facial expressions of pain in children," *Pain*, vol. 126, no. 1, pp. 64–71, 2006.

[74] Littlewort, Gwen C and Bartlett, Marian Stewart and Lee, Kang, "Automatic coding of facial expressions displayed during posed and genuine pain," *Image and Vision Computing*, vol. 27, no. 12, pp. 1797–1803, 2009.

[75] Gholami, Behnood and Haddad, MM and Tannenbaum, Allen R, "Relevance vector machine learning for neonate pain intensity assessment using digital imaging," *Biomedical Engineering, IEEE Transactions on*, vol. 57, no. 6, pp. 1457–1466, 2010.

[76] Brahnam, Sheryl and Nanni, Loris and Sexton, Randall, "Introduction to neonatal facial pain detection using common and advanced face classification techniques," in *Advanced Computational Intelligence Paradigms in Healthcare–1*, pp. 225–253, Springer, 2007.

[77] Nanni, Loris and Lumini, Alessandra and Brahnam, Sheryl, "Local binary patterns variants as texture descriptors for medical image analysis," *Artificial intelligence in medicine*, vol. 49, no. 2, pp. 117–125, 2010.

[78] Kaltwang, Sebastian and Rudovic, Ognjen and Pantic, Maja, "Continuous pain intensity estimation from facial expressions," in *Advances in Visual Computing*, pp. 368–377, Springer, 2012.

[79] Ashraf, Ahmed Bilal and Lucey, Simon and Cohn, Jeffrey F and Chen, Tsuhan and Ambadar, Zara and Prkachin, Kenneth M and Solomon, Patricia E, "The painful face–pain expression recognition using active appearance models," *Image and Vision Computing*, vol. 27, no. 12, pp. 1788–1796, 2009.

[80] Hammal, Zakia and Cohn, Jeffrey F, "Automatic detection of pain intensity," in *Proceedings of the 14th ACM international conference on Multimodal interaction*, pp. 47–52, ACM, 2012.

[81] Chew, Sien W and Lucey, Simon and Lucey, Patrick and Sridharan, Sridha and Conn, Jeff F, "Improved facial expression recognition via uni-hyperplane classification," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pp. 2554–2561, IEEE, 2012.

[82] Rudovic, Ognjen and Pavlovic, Vladimir and Pantic, Maja, "Automatic Pain Intensity Estimation with Heteroscedastic Conditional Ordinal Random Fields," in *Advances in Visual Computing*, pp. 234–243, Springer, 2013.

[83] Vukadinovic, Danijela and Pantic, Maja, "Fully automatic facial feature point detection using Gabor feature based boosted classifiers," in *Systems, Man and Cybernetics, 2005 IEEE International Conference on*, vol. 2, pp. 1692–1698, IEEE, 2005.

[84] Turgay Celik, "Lecture Notes on Adaptive Computation and Machine Learning, Lecture: Supervised Learning." http://www.cs.wits.ac.za/~celik/COMS4024/docs/supervised_learning.pdf, 2014. Accessed on October 2014.