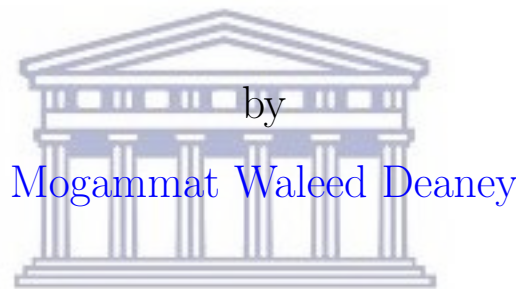


UNIVERSITY OF THE WESTERN CAPE

# A Comparison of Machine Learning Techniques for Facial Expression Recognition



Mogammad Waleed Deaney

UNIVERSITY of the  
WESTERN CAPE  
A thesis submitted in fulfillment for the  
degree of Master of Science

in the  
Faculty of Science  
Department of Computer Science

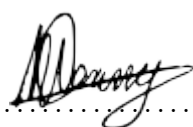
Supervisor: Isabella Venter  
Co-supervisor: Mehrdad Ghaziasgar  
Co-supervisor: Reginald Dodds

August 2018

# Declaration of Authorship



I, Mogammat Waleed Deaney, declare that this thesis “A Comparison of Machine Learning Techniques for Facial Expression Recognition” is my own work and that it has not been submitted before for any degree or assessment at any other university, and that all the sources I have used or quoted have been indicated and acknowledged by means of complete references.

Signature:  .....

Date: 28 May 2018 .....

*“If I have seen further than others, it is by standing on the shoulders of giants.”*

Isaac Newton

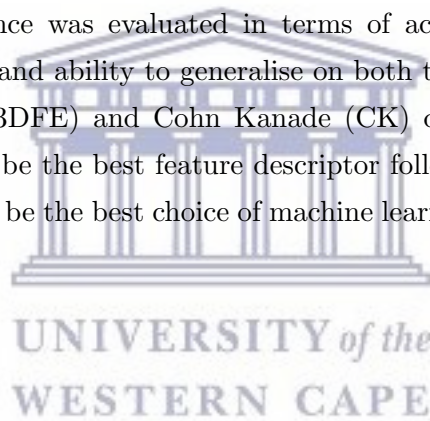


UNIVERSITY *of the*  
WESTERN CAPE

# Abstract

A machine translation system that can convert South African Sign Language (SASL) video to audio or text and vice versa would be beneficial to people who use SASL to communicate. Five fundamental parameters are associated with sign language gestures, these are: hand location; hand orientation; hand shape; hand movement and facial expressions.

The aim of this research is to recognise facial expressions and to compare both feature descriptors and machine learning techniques. This research used the Design Science Research (DSR) methodology. A DSR artefact was built which consisted of two phases. The first phase compared local binary patterns (LBP), compound local binary patterns (CLBP) and histogram of oriented gradients (HOG) using support vector machines (SVM). The second phase compared the SVM to artificial neural networks (ANN) and random forests (RF) using the most promising feature descriptor—HOG—from the first phase. The performance was evaluated in terms of accuracy, robustness to classes, robustness to subjects and ability to generalise on both the Binghamton University 3D facial expression (BU-3DFE) and Cohn Kanade (CK) datasets. The evaluation first phase showed HOG to be the best feature descriptor followed by CLBP and LBP. The second showed ANN to be the best choice of machine learning technique closely followed by the SVM and RF.



## Keywords

Facial expression recognition, Feature extraction, Machine learning, Support vector machine, Random forest, Artificial neural network, Local binary patterns, Compound local binary patterns, Histogram of oriented gradients

# Acknowledgements

First and foremost, I thank the Almighty God for blessing me and granting me the knowledge, wisdom and guidance to reach my goals and complete this thesis.

I would like to express my sincerest gratitude to my supervisors to Prof. Isabella Venter, Dr Mehrdad Ghaziasgar and Mr Reg Dodds. I feel honoured and privileged to have been guided by your insights and knowledge. I appreciate and value all the patience, motivation and advice you have extended to me throughout the years.

To my colleges and friends—Mr Da Costa, Mr De la Cruz, Mr Erasmus, Mr Jacobs, Mr Kakoko, Ms Mohamed, Mr Om, Mr Patience, Ms Walters and Mr Wu—thank you for all your support.

Many thanks to the National Research Foundation (NRF) and the Telkom/Cisco/Aria Technologies Africa Centre-of-Excellence—department of Computer Science—at the University of the Western Cape for their financial assistance.

Lastly, to family, thank you for always being there for me.



# Publications

- **Title:** A Comparison of Facial Feature Representation Methods for Automatic Facial Expression Recognition

**Author:** Waleed Deaney, Isabella Venter, Mehrdad Ghaziasgar, Reg Dodds

Published in the SAICSIT conference in Thaba 'Nchu, 2017



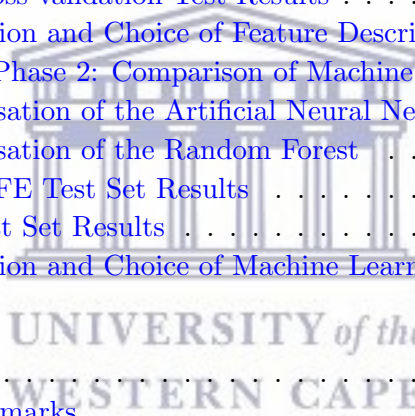
UNIVERSITY *of the*  
WESTERN CAPE

# Contents

<b>Declaration of Authorship</b>	<b>i</b>
<b>Abstract</b>	<b>iii</b>
<b>Keywords</b>	<b>iii</b>
<b>Acknowledgements</b>	<b>iv</b>
<b>Publications</b>	<b>vi</b>
<b>List of Figures</b>	<b>viii</b>
<b>List of Tables</b>	<b>x</b>
<b>Abbreviations</b>	<b>xi</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Background and Motivation . . . . .	1
1.2 Research Question . . . . .	3
1.3 Research Objectives . . . . .	3
1.4 Premises . . . . .	4
1.5 Thesis Outline . . . . .	4
<b>2 Related Work</b>	<b>6</b>
2.1 Local vs Global Methods . . . . .	6
2.2 Facial Expression Recognition System Components . . . . .	7
2.3 Feature Extraction . . . . .	8
2.3.1 Local Binary Patterns . . . . .	9
2.3.2 Local Ternary Patterns and Compound Local Binary Patterns . . . . .	12
2.3.3 Histogram of Oriented Gradients . . . . .	14
2.4 Classification . . . . .	17
2.4.1 Support Vector Machines . . . . .	17
2.4.2 Artificial Neural Networks . . . . .	18
2.4.3 Random Forests . . . . .	20



2.5	Comparison of Machine Learning Techniques . . . . .	20
2.6	Conclusion . . . . .	21
<b>3</b>	<b>Research Design and Methodology</b>	<b>23</b>
3.1	Research Philosophy . . . . .	23
3.2	Design Science Research . . . . .	25
3.3	The Design of This Research . . . . .	27
3.3.1	Artefact Design and Development . . . . .	27
3.3.2	Artefact Demonstration . . . . .	30
3.4	Conclusion . . . . .	48
<b>4</b>	<b>Results and Analysis</b>	<b>49</b>
4.1	Evaluation of Phase 1: Feature Selection . . . . .	50
4.1.1	Optimisation of the LBP . . . . .	50
4.1.2	Optimisation of the CLBP . . . . .	51
4.1.3	Optimisation of the HOG . . . . .	52
4.1.4	BU-3DFE Test Set Results . . . . .	53
4.1.5	CK Test Set Results . . . . .	58
4.1.6	CK Cross-validation Test Results . . . . .	59
4.1.7	Discussion and Choice of Feature Descriptor . . . . .	60
4.2	Evaluation of Phase 2: Comparison of Machine Learning Techniques . . . . .	62
4.2.1	Optimisation of the Artificial Neural Network . . . . .	62
4.2.2	Optimisation of the Random Forest . . . . .	63
4.2.3	BU-3DFE Test Set Results . . . . .	64
4.2.4	CK Test Set Results . . . . .	67
4.2.5	Discussion and Choice of Machine Learning Technique . . . . .	69
<b>5</b>	<b>Conclusion</b>	<b>71</b>
5.1	Future Work . . . . .	72
5.2	Concluding Remarks . . . . .	72
	<b>Bibliography</b>	<b>73</b>





# List of Figures

1.1	SASL translation system . . . . .	2
2.2	FER system components . . . . .	7
2.3	The original LBP operator . . . . .	9
3.4	DSR process model highlighting activity 3 of Section 3.2 . . . . .	29
3.5	Proposed FER artefact . . . . .	29
3.6	DSR process highlighting activity 4 of Section 3.2 . . . . .	30
3.7	Phase 1: Feature selection - Face detection . . . . .	32
3.11	Phase 1: Feature selection - Feature extraction . . . . .	34
3.12	Examples of varying number of points $P$ and radius $R$ of the LBP operator . . . . .	34
3.13	The LBP operator with parameters $R = 2$ and $P = 8$ applied to image in pixel representation . . . . .	35
3.14	Original facial image (left) and LBP image (right) . . . . .	35
3.15	Example of a uniform pattern . . . . .	35
3.16	Formation of the LBP feature vector . . . . .	36
3.17	Illustration of the generation of the CLBP code . . . . .	37
3.18	Generation of the sub-CLBP code . . . . .	37
3.19	Facial image partitioned into cells (left), blocks formed from $2 \times 2$ cells (centre) and 50% block overlap (right) . . . . .	39
3.20	Phase 1: Feature selection - Classification . . . . .	39
3.21	Two classes separated by a hyper-plane in SVM . . . . .	40
3.22	Phase 2: Comparison of machine learning techniques - Classification . . . . .	42
3.25	The structure of a simple decision tree . . . . .	46
4.1	DSR process highlighting activity 5 of Section 3.2 . . . . .	49
4.2	Phase 1: Feature selection - Evaluation . . . . .	50
4.3	Performance of each feature descriptor per facial expression class on the BU-3DFE test set . . . . .	54
4.4	Similarity of ‘Anger’ (top) and ‘Sad’ (bottom) expressions in the test set . . . . .	56
4.5	Similarity of ‘Anger’ (top) and ‘Disgust’ (bottom) expressions . . . . .	56
4.6	Histogram of the number of test subjects that achieved each number of correctly recognised images (out of seven) for each of the three feature descriptors . . . . .	57
4.7	Accuracy of each feature descriptor per facial expression class across all subjects on the CK dataset . . . . .	59
4.8	Phase 2: Comparison of machine learning techniques - Evaluation . . . . .	62
4.9	Optimisation results of artificial neural network . . . . .	63
4.10	Optimisation results of the Random forest . . . . .	64

---

4.11 Facial expression accuracies across machine learning techniques . . . . .	65
4.12 Test subject performance across machine learning techniques . . . . .	67
4.13 Comparison of facial expressions across machine learning techniques on the CK dataset . . . . .	68



UNIVERSITY *of the*  
WESTERN CAPE

# List of Tables

3.1	Distribution of labelled expressions in the CK dataset . . . . .	32
4.1	LBP Cross-validation optimisation scores (%) . . . . .	51
4.2	CLBP Cross-validation optimisation scores (%) . . . . .	52
4.3	HOG cross-validation optimisation scores (%) . . . . .	52
4.4	Performance of each feature descriptor on BU-3DFE test set . . . . .	53
4.5	Confusion matrix for LBP (%) (BU-3DFE) . . . . .	54
4.6	Confusion matrix for CLBP (%) (BU-3DFE) . . . . .	55
4.7	Confusion matrix for HOG (%) (BU-3DFE) . . . . .	55
4.8	Overall recognition results for the CK dataset . . . . .	58
4.9	Comparison of cross-validation accuracies on CK dataset . . . . .	60
4.10	Summary of feature descriptor results . . . . .	61
4.11	Performance of machine learning techniques on the BU-3DFE test set . . . . .	64
4.12	Confusion matrix SVM (BU-3DFE) . . . . .	66
4.13	Confusion matrix ANN (BU-3DFE) . . . . .	66
4.14	Confusion matrix RF (BU-3DFE) . . . . .	66
4.15	Performance of each machine learning technique on CK test set . . . . .	68
4.16	Summary of comparison of machine learning techniques . . . . .	69

UNIVERSITY of the  
WESTERN CAPE

# Abbreviations

<b>2D</b>	<b>T</b> wo <b>D</b> imensional
<b>3D</b>	<b>T</b> hree <b>D</b> imensional
<b>ANN</b>	<b>A</b> rtificial <b>N</b> eural <b>N</b> etwork
<b>AU</b>	<b>A</b> ction <b>U</b> nit
<b>BU-3DFE</b>	<b>B</b> inghamton <b>U</b> niversity <b>3D</b> <b>F</b> acial <b>E</b> xpression
<b>CLBP</b>	<b>C</b> ompound <b>L</b> ocal <b>B</b> inary <b>P</b> attern
<b>CK</b>	<b>C</b> ohn <b>K</b> anade
<b>CPU</b>	<b>C</b> entral <b>P</b> rocessing <b>U</b> nit
<b>CSU</b>	<b>C</b> ollarado <b>S</b> tate <b>U</b> niversity
<b>DSR</b>	<b>D</b> esign <b>S</b> cience <b>R</b> esearch
<b>FACS</b>	<b>F</b> acial <b>A</b> ction <b>E</b> ncoding <b>S</b> ystem
<b>FER</b>	<b>F</b> acial <b>E</b> xpression <b>R</b> ecognition
<b>GB</b>	<b>G</b> igabyte
<b>GPU</b>	<b>G</b> raphics <b>P</b> rocessing <b>U</b> nit
<b>HOG</b>	<b>H</b> istogram of <b>O</b> riented <b>G</b> radients
<b>IS</b>	<b>I</b> nformation <b>S</b> ystems
<b>JAFFE</b>	<b>J</b> Apanese <b>F</b> emale <b>F</b> acial <b>E</b> xpression
<b>LBP</b>	<b>L</b> ocal <b>B</b> inary <b>P</b> attern
<b>LibSVM</b>	<b>L</b> ibrary of <b>S</b> upport <b>V</b> ector <b>M</b> achines
<b>LTP</b>	<b>L</b> ocal <b>T</b> enary <b>P</b> attern
<b>ML</b>	<b>M</b> aximum <b>L</b> ikelihood
<b>MLP</b>	<b>M</b> ulti <b>L</b> ayer <b>P</b> erceptron
<b>PCRF</b>	<b>P</b> airwise <b>C</b> onditional <b>R</b> andom <b>F</b> orest
<b>RAM</b>	<b>R</b> andom <b>A</b> ccess <b>M</b> emory
<b>RBF</b>	<b>R</b> adial <b>B</b> asis <b>F</b> unction

<b>RF</b>	<b>R</b> andom <b>F</b> orest
<b>ROC</b>	<b>R</b> eceiver <b>O</b> perating <b>C</b> haracteristic
<b>SASL</b>	<b>S</b> outh <b>A</b> frican <b>S</b> ign <b>L</b> anguage
<b>SUSAN</b>	<b>S</b> mallest <b>U</b> nivalue <b>S</b> egment <b>A</b> ssimilating <b>N</b> ucleus
<b>SVM</b>	<b>S</b> upport <b>V</b> ector <b>M</b> achine
<b>SVM-RBF</b>	<b>S</b> upport <b>V</b> ector <b>M</b> achine— <b>R</b> adial <b>B</b> asis <b>F</b> unction
<b>SVM-POLY</b>	<b>S</b> upport <b>V</b> ector <b>M</b> achine— <b>P</b> olynomial <b>K</b> ernel



UNIVERSITY *of the*  
WESTERN CAPE

# Chapter 1

## Introduction

### 1.1 Background and Motivation

Communication is an integral part of life. By definition it is a means for us to exchange information with one another. It allows us to share our experiences, complete our daily activities and pass on relevant information and ideas in both social and professional aspects of our life. Verbal communication is considered to be one of the foremost forms of communication. It is a skill that is learned without much effort by people in the hearing community despite its complexities.

The hearing impaired and Deaf<sup>1</sup> communities use a visual or non-uttered form of communication called *sign language* in order to communicate with one another. The constitution of the Republic of South Africa recognises South African Sign Language (SASL) as the official language for Deaf South African communities. Despite this, communication and interactions between deaf and hearing individuals is complex as there is no standard form of communication and understanding between the two parties [1]. This often leads to feelings of annoyance and resentment which results in these communities feeling marginalised and isolated, and unable to fully reap the benefits of communication with the broader society [2].

It has been estimated that as many as 235 000 people in South Africa are profoundly deaf in both ears and use SASL as their main language [3]. Using interpreters to assist the non-hearing community may be impractical as skilled SASL interpreters are scarce

---

<sup>1</sup>People born without hearing and who are unable to communicate in a spoken language.

and costly [4]. The use of interpreters can incur issues of privacy when the signer does not want the interpreter to know personal information about themselves especially when dealing with the exchange of sensitive information. To help alleviate these problems, a translation system to translate sign language to speech or text and vice versa would be of great assistance to the community. The system will assist the Deaf and hearing impaired communities to communicate without the use of interpreters and would also attend to issues of privacy such as a medical consultation.

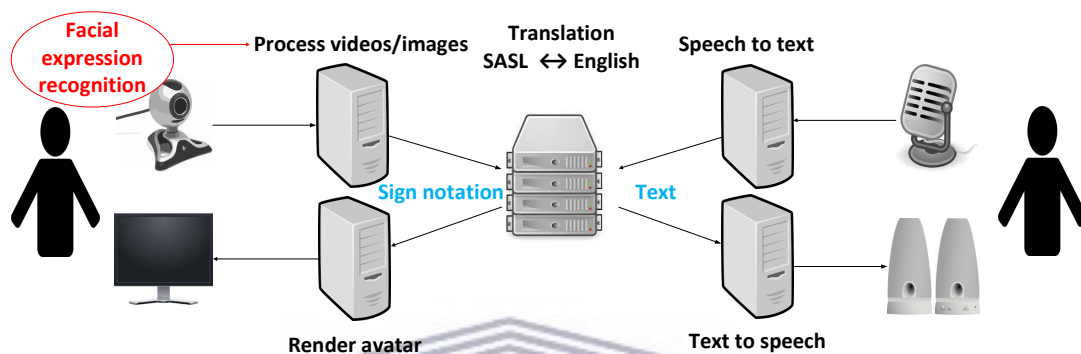


FIGURE 1.1: SASL translation system

Figure 1.1 illustrates an automatic SASL translation system which translates SASL to English, and vice versa. However to make such a complex system a reality would require a multifaceted approach including image processing, artificial intelligence, natural language processing and linguistics [5]. A significant aspect of any SASL translation system its ability to interpret SASL gestures from videos using a web cam.

When recognising sign language, research indicates that there are five core parameters which should be considered, namely, (1) hand shape, (2) hand orientation, (3) hand location, (4) hand motion, and (5) facial expressions [6]. The research into the SASL translation system has therefore been focused on these core parameters. Previous research towards the translation system have successfully implemented the system's hand shape [7, 8], hand location [2, 9], hand motion [5, 10, 11] and facial expressions [12, 13] recognition capabilities.

This research focuses on the implementation of an automatic facial expressions recognition (FER) system. Facial expressions provide a means for conveying our emotional state and intentions [14] and are understood across many cultures [15]. This makes FER research an interesting task as it impacts fields such as human computer interaction across a variety of applications.

The research aims to recognise facial expressions and to compare feature extraction techniques: local binary patterns (LBP), compound local binary patterns (CLBP) and histogram of oriented gradients (HOG) as well as machine learning techniques: support vector machines (SVMs), random forests (RFs), and artificial neural networks (ANNs). Each of these feature extraction and machine learning techniques mentioned has its own set of parameters which need to be optimised. Once optimum values are found, a classification model needs to be trained in order to classify data, and then built to test the system on unseen data. All of these factors will be attended to in this research in order to provide the criteria for an informed decision on choosing a machine learning and feature extraction technique for facial expression recognition.

## 1.2 Research Question

The question thus is: “Which feature extraction and machine learning techniques are best suited for facial expression recognition?” This question can be translated into two sub-questions:

1. How do the feature extraction techniques, i.e., local binary patterns, compound local binary patterns and histogram of oriented gradients compare in the context of facial expression recognition?
2. How do the machine learning techniques, i.e., support vector machines, artificial neural networks and random forests, compare in the context of facial expression recognition?

The solution to these research questions should provide the criteria to make an informed decision when choosing a feature extraction and machine learning technique for classifying facial expression features as part of the complete SASL machine translation system.

## 1.3 Research Objectives

The objectives of the research are to extract and compare relevant facial feature descriptors and various machine learning techniques when classifying facial expressions. This



will be done by investigating and comparing the use of LBP, CLBP and HOG as facial feature descriptors. Using SVMs to classify the facial expression the most promising descriptor will be chosen. The chosen descriptors will be used as the final feature set to be classified with ANNs and RFs to determine which is best suited for the recognition of facial expressions in terms of accuracy. The objectives are as follows:

1. To optimise the feature extraction techniques LBP, CLBP and HOG using a SVM. The ideal parameters will be found for each feature extraction technique and compared.
2. To optimise and train a SVM, an ANN and a RF based on a set of facial expressions. The ideal parameters will be found for each machine learning technique and compared.

## 1.4 Premises

The premises of the FER system are:

- The expressions will be classified by associating them with the seven basic emotions namely, anger, fear, disgust, happiness, surprise, sadness, and the neutral expression.
- Only 2D images of the expressions with an uninhibited view of the user's face will be considered.

## 1.5 Thesis Outline

The rest of the thesis is organised as follows:

**Chapter 2:** Related work: This chapter reviews work related to facial expression recognition in order to build an understanding of how FER systems are generally executed. The related work demonstrates that SVMs, RFs, ANNs, LBP, CLBP and HOG have been used as accurate classification techniques.

**Chapter 3:** Research design: This chapter discusses the elements of research design in terms of the philosophy and the methodology of design science research which forms a

foundation for the research by developing a framework/artefact for the implementation and development of the FER system.

**Chapter 4:** Experimental results and analysis: This chapter describes the experimental results of the FER artefact and reports the computational accuracy of the optimisation procedures and testing procedures for each of the feature extraction techniques and the machine learning techniques.

**Chapter 5:** Conclusion: This chapter summarises the thesis by providing the key observations, interpretations and answers to the main research question and sub-questions.



## Chapter 2

# Related Work

Automatic facial expression recognition (FER) is a well-researched topic. The prominence of FER research is due to the impact it has across a number of fields such as human computer interaction and its application to other theoretical interests [14, 16–18]. This chapter discusses the key terms and components of FER. The chapter will also survey a study which compares machine learning techniques for a general classification problem.

### 2.1 Local vs Global Methods

The two main ways in which researchers implement automated FER systems are by using local or global methods. Local methods are associated with the facial action encoding system (FACS) and global methods with a set of prototypic facial expressions.

The FACS was originally developed by Hjortsjö [19] and extended by Ekman et al. [20, 21]. The FACS defines emotion by the appearance of key facial movements and has been useful to both animators and psychologists. Key facial movements are labelled as action units (AUs). AUs can refer to a single facial movement or a group thereof. The FACS captures the subtlety of facial expressions, however AU labels are purely descriptive. This means that in order to get an interpretation of the facial expression, AUs need to be tracked and converted into the emotional FACS system or an other similar system.

Instead of recognising detailed local methods of the face such as the FACS, FER systems attempt to use global methods, i.e., the entire face. The faces are represented

by a discrete set of prototypic facial expressions. The prototypic set of facial expressions has been shown to be recognisable across people of various cultures and social backgrounds [22]. The set consists of the facial expressions: anger, disgust, fear, sadness, happiness and surprise as shown in Figure 2.1. These expressions pose a six-class problem for FER systems, however in some cases FER systems attempt to recognise a seventh class the neutral expression.

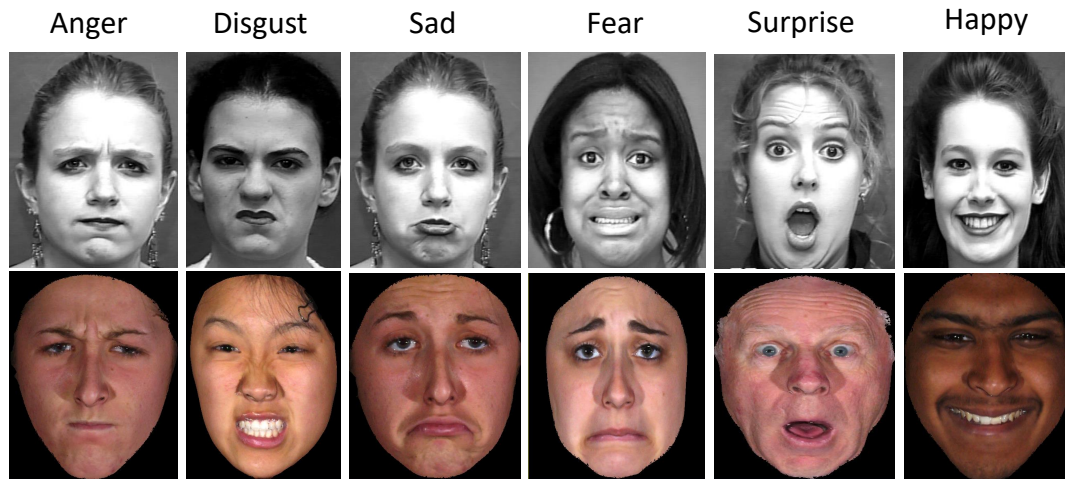


FIGURE 2.1: Six basic emotions. Images from [23, 24]

## 2.2 Facial Expression Recognition System Components

Image processing and machine learning techniques form the basis for research in automated FER systems. Figure 2.2 shows the three high-level components which are generally used to form most FER systems [17]. The components are: face detection, feature extraction, and classification.

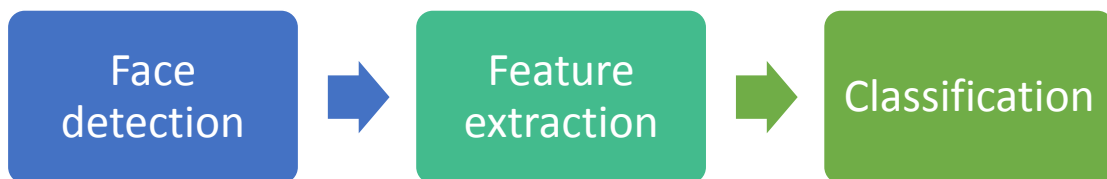


FIGURE 2.2: FER system components

Face detection is the first step of many FER systems and has been a topic of research for many years due to its many applications in computer vision software. The goal of

face detection is to determine whether any faces are present in an image. If present, the face detector then returns the face location.

Once the face is detected and isolated from the image, feature extraction is applied. The feature extraction component involves extracting “interesting” properties from the facial image to form a feature set.

Classification is the process that produces a predicted output based on the feature set given as input. The process usually consists of training and prediction stages. The training stage computes a model capable of using the features to predict classes. In the prediction stage features are fed to the model which produces a recognition output.

Researchers have mixed and matched numerous algorithms associated with each of the components and formed a multitude of FER systems. Only the feature extraction and classification components will be discussed in the subsequent sections. This is justified by the omission of face detection procedures in many FER studies [12] instead using images of only the face. However, in the subsequent sections, where possible, the face detection procedure of the reviewed FER system will be discussed.

## 2.3 Feature Extraction

Feature extraction in FER systems can be motion-based, model-based or appearance-based [12]. Motion-based feature extraction uses the displacement of pixels to provide information about the motion of the face. Model-based feature extraction uses statistical methods to build a set of model parameters to describe the face. Lastly, appearance-based features provide information on the texture or shape of the face using pixel properties.

Mushfieldt [12] researched various feature extraction techniques associated with each of the categories. He concluded that appearance-based techniques outperform model-based techniques and are on par with motion-based techniques. However motion-based techniques require illumination normalised image sequences whereas appearance-based techniques are more robust, using statistical methods and work on static images. Of the

appearance-based methods the local binary patterns (LBP) was compared to Gabor-wavelets. LBP was selected as the technique of choice due to the complexity and computational requirements of Gabor-wavelets.

The studies below expand on the research done on appearance-based methods. FER systems which use LBP are discussed in Section 2.3.1. Compound local binary patterns (CLBP) and local ternary patterns (LTP) are discussed in Section 2.3.2 and histogram of oriented gradients (HOG) is discussed in in Section 2.3.3.

### 2.3.1 Local Binary Patterns

Over the past decade LBP has become one of the leading texture-based feature extraction methods used in computer vision systems [25]. It is especially popular in the field of image processing. This is due to its ease of computation and robustness towards changes in illumination. The LBP was originally proposed by Ojala et al. [26]. It is used to describe an image in terms of texture and textural changes. The operator is applied to a grey scale image and produces a grey scale texture image.

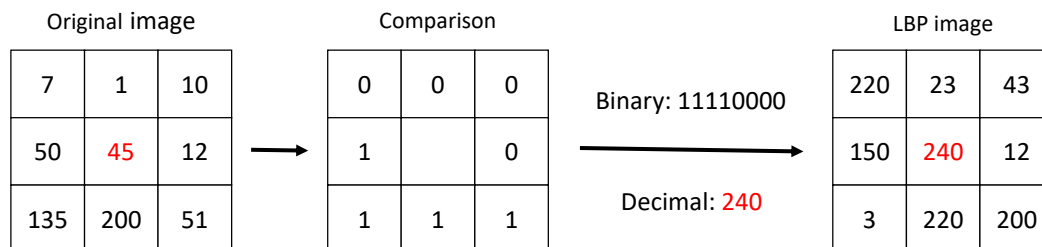


FIGURE 2.3: The original LBP operator

Figure 2.3 illustrates the computation of the original LBP operator which is applied to every  $3 \times 3$  pixel cell of an image. The center pixel is compared to each neighbouring pixel values using a threshold function. If the centre pixel value is greater than or equal to its neighbour write '0' else '1'. As a result an eight-bit binary number is produced. The binary pattern is then converted to decimal and used as the pixel value for the LBP image.

Feng et al. [27] used LBPs and a linear programming classification technique to recognise the six prototypic facial expression classes and the neutral expression. The images were preprocessed using the CSU (Colorado State University) Face Identification Evaluation

System [28]. This resulted in cropped images of size  $150 \times 128$  pixels which excludes non-face area. Figure 2.4 shows images used from the JAFFE dataset [29].



FIGURE 2.4: Original images (top) and images after preprocessing (below) [27]

The original LBP operator was applied to the normalised images. The resulting LBP image was divided into non-overlapping local regions of size  $10 \times 8$  pixels. Local histograms of each region were computed and concatenated to form a single feature vector representing the facial expression. The size of the feature vectors was reduced by discarding patterns with frequencies that fell below a certain threshold, which was the averaged sum of all the feature vectors of the training samples.

A linear programming technique was the classifier of choice. This technique generates a plane which minimizes an average sum of misclassified points belonging to two disjoint point sets. Twenty-one two-pair sets were generated from the seven-class expression problem: Happy-Disgust, Happy-Fear, etc. For the training, each of the 21 expression pairs was trained. The predicted class was generated by feeding the feature vector of each of the test samples to each of the classifiers.

The experiments were carried out on the seven-class JAFFE dataset [29] and used a 10-fold cross-validation evaluation scheme. The process was repeated 20 times and resulted in an average accuracy of 93.8%.

Mushfieldt et al. [12] researched facial expression recognition in the presence of rotation and partial occlusion of the face. The system catered for both frontal and rotated face segmentation. The Viola and Jones [30] frontal face detection was used to detect frontal faces. If the face was not detected it was considered a rotated face. The rotated face

was detected using a skin segmentation algorithm [2]. Furthermore, an eye detection algorithm [31] was implemented to normalise the orientation of the image.

The extended multi-scale LBP<sub>8,2</sub>, which has been shown to be an accurate representation of facial features [32], was applied to the normalised image. Uniform patterns [33] which decrease the feature size dramatically were also applied. This resulted in a uniform rotation invariant LBP texture image. The image was then divided into small regions of equal size. Histograms were computed for each region to form the final feature vector. Figure 2.5 illustrates a rotated and frontal face image and the LBP texture images of both.



FIGURE 2.5: Frontal (bottom) and rotated face (top) images and the extracted LBP texture images [12]

A SVM using the radial basis function (RBF) kernel was used for classification of the facial expressions. The one-against-one approach [34] was used for multi-class classification. The experiment was carried out on a pruned BU-3DFE dataset [24] which contains 50 subjects posing one frontal and one 60° image of the six prototypic facial expressions. The system was trained on ten subjects and the remaining 40 subjects were used exclusively as testing data. The training procedure optimised arbitrary values for the resolution and region size of the LBP image. The optimised frontal image size was 40 × 60 pixels with a region size of 8 × 10 pixels. The optimised rotated image size was 40 × 50 pixels with a region size of 8 × 5 pixels. The system achieved a 75% average accuracy for frontal faces and a 70% average accuracy for rotated faces.

From the studies above, it is seen that it is common practice to divide the LBP image into smaller equal regions, extract histograms from each region and concatenate these



histograms to represent the final LBP feature vector. This idea was proposed by [32] to consider local texture information of the face and has provided an improved feature representation of the face.

LBP's have been adapted to provide improved performance and other advantages when dealing with the rigours of image analysis problems. These adapted versions include local ternary patterns (LTP) and compound local binary patterns (CLBP) amongst others. These extensions modify various aspects of LBP's such as the comparison of the neighbouring pixels with the centre pixel, etc., but come at the cost of computation and feature vector size. They are described in the subsection that follows along with related studies that have applied them for FER.

### 2.3.2 Local Ternary Patterns and Compound Local Binary Patterns

LTP was introduced by Tan et al. [35] and extends LBP by adding an additional discrimination level to the LBP thresholding function. LTP provides increased robustness to noise when compared to LBP [35]. However, due to the application of a threshold constant, LTP is no longer strictly invariant towards gray-level transformations [36].

The LTP operator is applied to a grey scale image and produces a two grey scale texture images. As with LBP, LTP is applied to every  $3 \times 3$  pixel cell of an image.

For each  $3 \times 3$  cell the following procedure is carried out:

1. Choose a user-specified constant threshold  $t$  which sets the level of tolerance to noise.
2. Compare the centre pixel  $p_c$  in the cell with threshold  $t$  to each of the surrounding/neighbouring pixels  $\{p_n | n = 0, \dots, 7\}$  using the LTP threshold function in Equation 2.1. The comparison may start from any neighbouring pixel in any direction as long as it is applied consistently, i.e.:

$$f(p_c, p_n) = \begin{cases} 1 & p_n \geq p_c + t, \\ 0 & |p_n - p_c| < t, \\ -1 & p_n \leq p_c - t. \end{cases} \quad (2.1)$$

3. The comparison produces an eight digit number called the local ternary pattern. To reduce the feature dimensions the local ternary pattern is converted into an eight-bit upper pattern and an eight-bit lower pattern [35]. The upper pattern is formed by substituting the '-1' value with '0' and the lower pattern by substituting the '-1' value with '1'.
4. The eight-bit binary codes of the upper and lower patterns are converted to decimal and assigned as new values for each of the upper and lower LTP images at the position corresponding to the centre pixel.

Figure 2.6 illustrates the LTP encoding process.

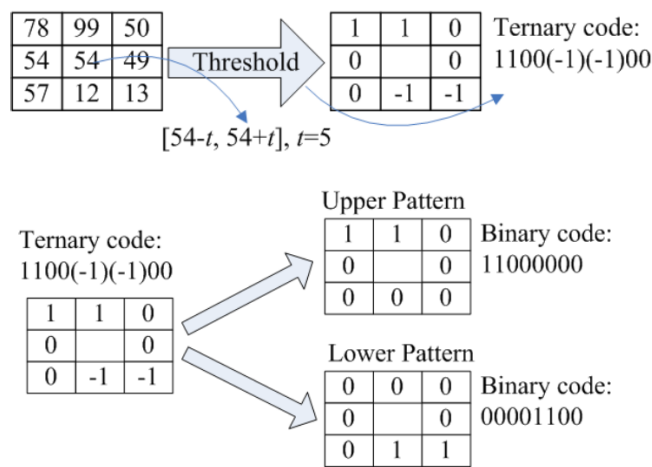


FIGURE 2.6: The LTP encoding process [35]

CLBP was introduced by Ahmed et al. [37]. CLBP assigns an extra bit for comparison between the average magnitude of the neighbourhood  $M_{avg}$  and the difference between the centre pixel and its neighbour. CLBP adds additional magnitude information when comparing the centre pixel to neighbouring values which is otherwise lost in LBP.

Ahmed et al. [37] successfully implemented each of LBP, LTP and CLBP as feature descriptors in three FER systems. The work compared LBP, LTP and CLBP on both the six-class prototypic facial expressions and the seventh-class, which included the neutral pose. The images were preprocessed by cropping images based on the position of the eyes. It is unclear which algorithm was used to determine the location of the eyes. This resulted in images of size  $150 \times 110$  pixels which excluded the non-facial area.

An experiment was carried out which compared LBP, LTP and CLBP. The LBP was implemented using the original LBP descriptor together with the uniform patterns extension [33]. The histogram procedure in [32] was applied which represented the LBP feature vectors. The CLBP operator was applied to the normalised image which computed two sub-CLBP images. The histogram procedure [33] was applied to both images which were concatenated to form the final CLBP feature vector. It is unclear which threshold value was used to implement LTP, as the LTP was not detailed in the work. Each descriptor was tested on three region sizes,  $3 \times 3$ ,  $5 \times 5$  and  $7 \times 6$ .

A SVM with the radial-basis function kernel was used as the classifier of choice. The SVM used a one-against-rest approach to achieve multi-class classification. An experiment was carried out on the CK [23] and the JAFFE [29] datasets on both six and seven classes. The six-class CK dataset contained 1224 labelled images. An additional 408 images labelled as the neutral expression were added for seven-class classification. A 10-fold cross-validation scheme was used to evaluate performance. A summary of best accuracies per descriptor is given in Table 2.1

TABLE 2.1: Summary of accuracy of LBP, LTP and CLBP in [37]

Descriptor	Accuracy (%)			
	CK dataset		JAFFE dataset	
	six-class	seven-class	six-class	seven-class
LBP	90.1	83.3	90.5	85.3
LTP	93.6	88.9	90.9	86.7
CLBP	<b>94.4</b>	<b>90.4</b>	<b>92.2</b>	<b>87.5</b>

The results indicate that the accuracies of each feature descriptor are exceptional. Upon comparison both the LTP and CLBP prove to be superior to LBP. The CLBP descriptor show the most promise with the highest accuracies across the CK and JAFFE datasets.

### 2.3.3 Histogram of Oriented Gradients

The histogram of oriented gradients (HOG) was first introduced by Dalal et al. [38]. The HOG descriptor uses a distribution of local intensity gradients or edge directions to form local histograms as image features. Research has shown that the use of the HOG descriptor can be extended to represent facial features which are characterised by local appearance and shape [16]. Given an image of the region of interest, the HOG algorithm is implemented as follows:

1. Calculate the gradient images—Compute the image for the vertical and horizontal gradients using an edge detection algorithm. From the resultant image compute the magnitude and direction of the gradient images
2. Calculate the histogram—Divide the image into cells of equal size. Select an appropriate orientation bin size in the ‘signed’ or ‘unsigned’ range of the gradients. For these cells compute a histogram of gradients.
3. Block normalization—Group cells to form overlapping blocks of equal size. Normalise each of the block histograms locally.
4. Compute the feature vector—Concatenate each of the normalised histogram blocks to form the final feature vector which represents the HOG descriptor.

Gritti et al. [16] investigated the use of local features for FER with face registration errors. Their research investigated and compared the use of the HOG, LBP and the LTP feature descriptors. The comparison included a novel approach of overlapping the local regions of LBP and LTP. The overlapping of local regions was a method investigated by [38] for HOG. Preprocessing consisted of scaling images based on the distance between the eyes which were manually located using [39]. This resulted in normalised face images of size  $108 \times 147$ .

The experiment also investigated the use of the HOG parameter variables suggested in [38]. This HOG implementation was used as a baseline to determine whether changing a variable of one of the parameters would have a significant effect on FER accuracy. The results indicated that the cell size and block size had the most significant impact on FER accuracies.

Furthermore a comparison was drawn between HOG, LBP and LTP. HOG was implemented using Prewitt filters [40] for gradient computation. Histograms were calculated with ‘signed gradients’ and 18 orientation bins. The block size was  $24 \times 24$  pixels and the cell size was  $8 \times 8$  pixels. The blocks were  $\frac{1}{2}$  overlapped. LBP was implemented using the  $LBP_{8,2}^{u2}$  extension. The region size used was  $6 \times 7$  pixels. LTP was implemented with the same extensions and a threshold value of six. Furthermore, as local regions were overlapped to form the HOG feature vector, the the LBP and LTP regions were overlapped. LBP regions were overlapped by  $\frac{1}{2}$  and LTP by  $\frac{3}{4}$ .

Multi-class classification was achieved using a linear SVM and the one-against-rest technique. The performance of the system was evaluated using a 10-fold cross-validation scheme on a pruned six-class CK dataset consisting of 310 images. Table 2.2 displays the results of their experiment.

TABLE 2.2: Linear SVM recognition performance (%) for feature descriptors [16]

<b>Feature descriptor</b>	<b>Performance</b>
LBP	90.9 $\pm$ 5.6
LBP-Overlap	92.9 $\pm$ 5.0
LTP	90.9 $\pm$ 4.9
LTP-Overlap	91.7 $\pm$ 5.6
HOG	92.7 $\pm$ 3.4

The results indicate that each feature descriptor achieves high performance with the LBP and LTP feature descriptors and are on par with one another. The performance of the LBP-Overlap and LTP-Overlap are seen to be superior compared to both LBP and LTP, however, the LBP-Overlap and LTP-Overlap feature dimensions are significantly larger than their LBP and LTP counterparts. That considered, the slight increase in performance cannot be justified. HOG was shown to be a promising feature descriptor with a high accuracy and a lower variance than the rest of the descriptors.

Unlike Gritti et al. [16], Chen et al. [41] proposed a method for FER based on facial components and HOG features. The system detected the face and isolated the eyes, nose and mouth. The HOG descriptor was applied to the components and concatenated to form the feature set. Figure 2.7 illustrates their system design.

The face was detected with the Viola-Jones face detector [30]. The face was resized to a resolution of  $156 \times 156$  on the JAFFE dataset and  $256 \times 256$  on the extended CK dataset. An upper and lower facial component was detected based on the location of the eyes. It is unclear from the literature how the eyes were detected. The upper component contained the eyes and eyebrows, whereas the lower component contained the mouth and nose. The upper facial component was resized to  $52 \times 106$  pixels and the lower component to  $78 \times 104$  pixels.

The HOG descriptor was applied to each of the facial components with cell size  $8 \times 8$  pixels, nine orientation bins and an ‘unsigned’ range. No further HOG implementation details were given. The HOG features of the facial components were concatenated and formed the final feature vector.

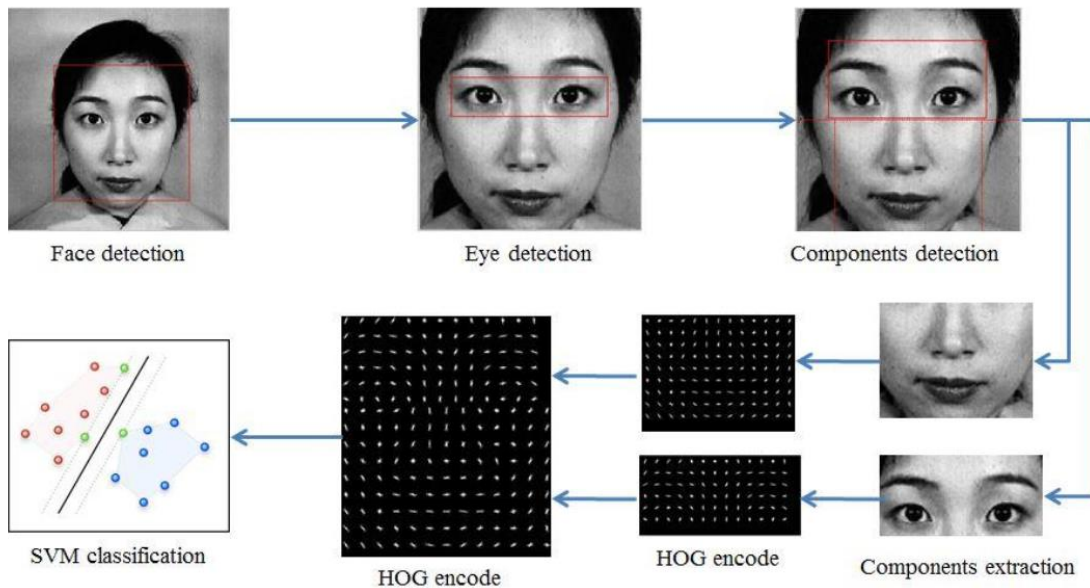


FIGURE 2.7: HOG based FER system design [41]

A linear SVM using the one-against-rest strategy was used to perform multi-class classification. The system was evaluated on the seven-class JAFFE dataset and the seven-class CK dataset. The CK dataset was pruned to contain only labelled data.

The system achieved an accuracy of 94.3% on the JAFFE dataset using a leave-one-sample-out testing strategy and 88.7% on the extended CK dataset using a leave-one-subject-out testing strategy. The experiment provided exceptional accuracies on both datasets.

## 2.4 Classification

Numerous machine learning techniques have been used for the classification of facial expressions such as random forests (RFs), artificial neural networks (ANNs), support vector machines (SVMs), linear discriminant analysis, and hidden Markov models among others ???. The literature in the sections below focusses on the popular supervised machine learning techniques SVMs, ANNs and RFs.

### 2.4.1 Support Vector Machines

The SVM was introduced by Vapnik et al. [42] as a machine learning technique for binary classification problems. It is a supervised machine learning technique: meaning

that given a set of labelled training data, usually vectors, the SVM training algorithm maps the data and separates them in order to predict to which class or category unseen data will belong. SVMs can also cope with multi-class classification, regression and outlier detection.

Shan et al. [43] proposed a robust facial expression system using a low computation discriminative feature space. The system used a template matching method with a weighted Chi-square statistic and SVMs for classification.

Faces were normalised to a fixed distance between the eyes [39]. This resulted in normalised facial images of size  $110 \times 150$  pixels. The  $LBP_{8,2}^{u_2}$  extension was applied to the image and the histogram procedure was used to build the feature vector. The region size used was  $6 \times 7$  pixel.

A SVM with the one-versus-rest approach was used for multi-class classification. The linear, polynomial and RBF kernels were used. Experiments were conducted on a pruned six-class CK dataset containing 320 images. The system was evaluated on a 10-fold cross-validation testing scheme. Their system achieved the following results: 87.2% using the linear kernel, 88.4% with the polynomial kernel and 87.6% using the radial basis function (RBF) kernel.

SVMs are a popular choice of machine learning technique for FER systems [12, 16, 37, 41, 43]. This is due to the many advantages of SVMs these include effectiveness in higher dimensional space, memory efficiency and versatility in terms of decision functions.

### 2.4.2 Artificial Neural Networks

An artificial neural network also known as a neural network, is a machine learning technique inspired by the structure of neural networks in the human brain. The brain is a highly complex information processor and purportedly contains  $10^{11}$  neurons with  $10^{14}$  interconnections in a complex network which allows us to perform certain computations. The ANN models this idea in computing terms. The following studies successfully implemented ANNs:

Khandait et al. [44] implemented a multi-layer perceptron (MLP) as the class of ANN for their FER system. The system segmented local face components using an array of

morphological image processing operations which included SUSAN edge detection [45] and prior face image knowledge. The local face components extracted were, the eyes, nose, mouth and eyebrows. The height, width and distance between these components were used as features. Figure 2.8 illustrates the extracted facial components.

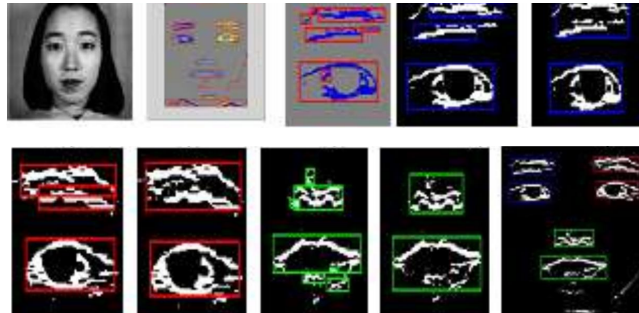


FIGURE 2.8: Extracted facial image components [44]

The MLP classifier consisted of 15-neuron input layers, two hidden layers and seven-neuron output layers. Experiments were carried out on a pruned JAFFE dataset of which 120 images were used for training and 30 images for testing. The system achieved an accuracy of 96.42% when recognising the seven facial expressions.

Similarly, Rázuri et al. [46] implemented a FER system that analysed the eye and mouth features of a facial image using an ANN. The eye and mouth regions of the image were extracted and merged together. The merged image was resized to a resolution of  $30 \times 40$  pixels and binarised using a threshold function. Figure 2.9 illustrates the procedure.

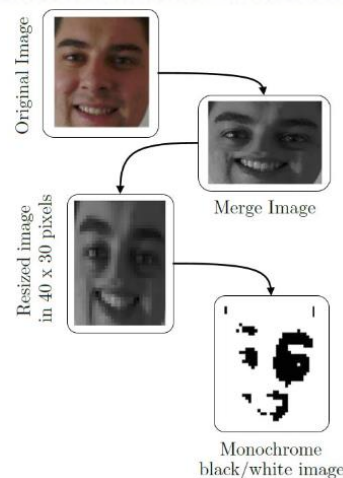


FIGURE 2.9: Preprocessing of images [46]

The binary image was used as input to the ANN. The class of ANN used was a feed-forward neural network trained by back-propagation. The input layer consisted of 1200



neurons corresponding to the size of the binary image. The sigmoid activation function was used along with one hidden layer. The system was evaluated on a pruned six-class CK dataset. The ANN was trained on 90 images and tested on 250 and achieved an average accuracy of 84%.

### 2.4.3 Random Forests

The random forest (RF) is a popular machine learning technique introduced by Breiman et al. [47]. RFs have been extensively used in computer vision and FER systems. This is due to their ability to handle high-dimensional data such as images and being suited for multi-class classification [48]. RFs are an ensemble of decision trees which collectively form a forest [47]. Decision trees as a classifier tend to over-fit. To overcome this RFs implement a technique called bootstrap aggregation or bagging, to form a powerful classifier.

Dapogny et al. [48] successfully implemented a FER framework using an RF and an extended version of RFs called pairwise condition random forests (PCRF). The PCRF classifier was used on high-dimensional temporal information. Additionally their research compared PCRF to RFs using static images. Local points were positioned on the face and used features. The out-of-bag error estimate [47] was used as the performance metric for testing on the CK and BU-3DFE datasets. The system achieved 93.2% on the CK dataset and 70% BU-3DFE dataset using RFs as the classifier.

## 2.5 Comparison of Machine Learning Techniques

Foster [8] compared machine learning techniques for SASL hand gestures. The system implemented a feature extraction method designed by Li et al. [7]. The feature extraction method isolated and tracked the hands using a combination of skin detection and motion detection techniques.

The system was tested on a self collected dataset consisting of ten SASL hand shapes produced by 12 ethnically diverse subjects. Figure 2.10 illustrates the SASL hand shapes.

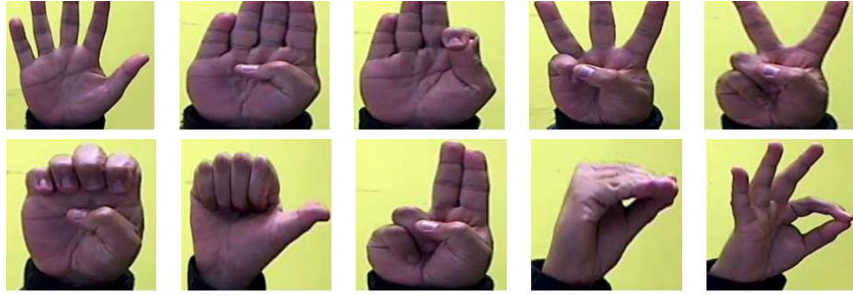


FIGURE 2.10: Ten SASL hand shapes [8]

His work contributed a comparison framework that can be used to optimise and compare machine learning techniques. His framework compared SVMs to ANNs and RFs in terms of accuracy and time taken to train, optimise and classify. The SVM used the radial basis function kernel for multi-class classification. The parameters tuned for the SVM were the cost of classification and the gamma values. The classifier used for the ANN was the multi-layer perceptron. The parameters optimised for the ANN were the number of hidden layers and neurons. The parameters tuned for the RF were the depth and the number of trees. Table 2.3 summarises the results and the analysis of the study.

TABLE 2.3: Summary of the results and the analysis in [8]

Factor	SVM	ANN	RF
Overall Accuracy (%)	84.3	<b>85.93</b>	81.33
Robust to Subjects	High	<b>Best</b>	High
Robust to Hand shapes	High	<b>Best</b>	High
Classification Time (s)	20.974	0.061	<b>0.033</b>
Optimisation Time (s)	<b>109</b>	3589	14916
Training Time (s)	<b>21</b>	39	101

It was concluded that all the machine learning techniques achieved good performance with the ANN as the best classifier followed by the SVM and RF. In terms of classification speed the RF proved to be the best followed by the ANN then the SVM. Overall, it was concluded that the ANN was the most suitable classifier, due to its accuracy, consistency, robustness and exceptionally high classification speed.

## 2.6 Conclusion

This chapter provided an overview of the three main components used by researchers to achieve FER. The chapter reviewed and discussed the use of LBP, LTP, CLBP, HOG, ANN, RF and SVM in FER systems. All of the above studies on FER systems clearly

demonstrate that these techniques achieved good accuracies for their respective FER systems. A study which compared machine learning techniques for hand gestures was also discussed. The studies demonstrated that the performance of a machine learning technique varies according to the given set of features. Therefore it is crucial to compare a variety of machine learning techniques for a set of features, to determine the optimal technique.

The next chapter discusses the way in which the research will be done in terms of structure, research philosophy, design, methods and methodologies.



## Chapter 3

# Research Design and Methodology

The previous chapter discussed the main components of FER systems and some implementations were considered. A comparison of various machine learning techniques used for solving general classification problems was also discussed. In this chapter the research design is considered. To ensure consistency of the research the philosophical stance is clarified in Section 3.1. The methodology adopted for this research is described in Section 3.2 and its implementation dealt with in Section 3.3. The chapter is concluded in Section 3.4.

### 3.1 Research Philosophy

Research philosophy is broadly defined as the development of knowledge and the nature of knowledge which can be regarded as the belief and the way in which a researcher collects, analyses and questions phenomena [49].

In order to define and understand the research philosophy in this research, the research process will be guided by Crotty [50], who suggests that there are four research elements which guide the research process namely: epistemology, theoretical perspective, methodology, and methods.

Crotty poses the following questions with regards to the research elements to help supervise the research design process:

1. What *methods* do we propose?
2. What *methodology* supervises the chosen methods?
3. What *theoretical perspective* forms the grounding for the methodology?
4. What *epistemology* informs the theoretical perspective?

The questions illustrate the relation between the research elements and this creates a hierarchy which forms the decision making process, represented in Figure 3.1. The use of the decision making process ensures the research is conducted in a clear and logical manner.

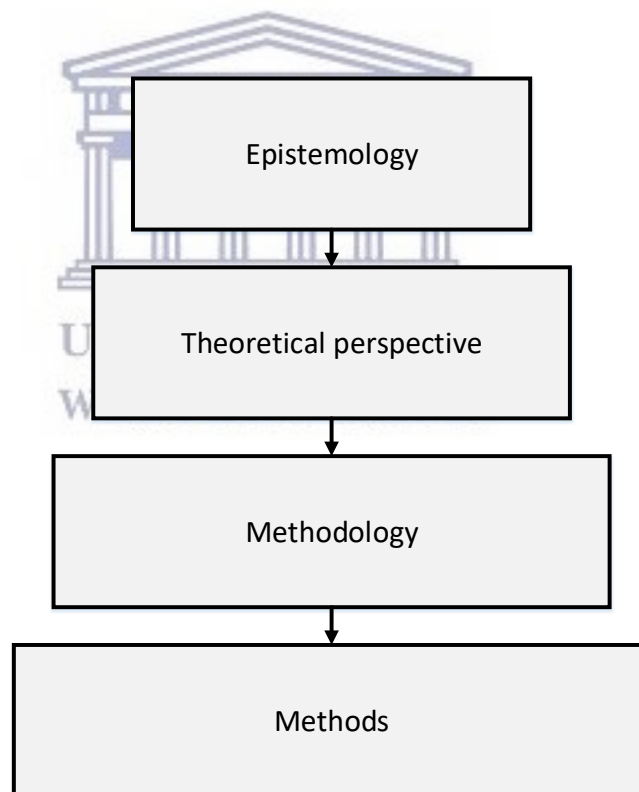


FIGURE 3.1: Decision making process [50]

Objectivism is the chosen *epistemological* stance in this research. Objective knowledge exists independently of the observer. Objectivism often asserts logic as a means of conceptual knowledge and claims that truths are absolute. The objective researcher ignores

emotions and intuitions and is therefore inclined to a quantitative style of research. This research is quantitative and takes an objective stance.

The *theoretical perspective* of this research is positivist. The objectivist epistemology guides the research towards a positivist stance. A positivist perspective assumes that properties of knowledge are measured directly through observation.

*Methodology* and *methods* are terms which are often used synonymously by researchers [50]. In this research, methodology refers to the systemic approach whereby the methods are governed and infers how the data are collected and analysed. Whilst methods refer to the tools, techniques and algorithms used to analyse data. Therefore the methodology supervises the chosen methods.

The objective epistemology together with the positivist theoretical perspective avails a range of research methodologies. Given this the methodology of choice is design science research. The methods used in this research are quantitative and refer to the algorithms and techniques forming the artefact.

## 3.2 Design Science Research

Design science research (DSR) is an outcome based research methodology which has been adopted in the fields of information systems and computer science. DSR offers a definitive instruction for evaluation and iteration within research projects. At the core of DSR is the development and implementation of design artefacts.

The first DSR framework for information systems (IS) was produced by Smith et al. [51]. Their work includes basic definitions for DSR and the artefact. Hevner et al. [52] refined DSR by conceptualising a widely accepted DSR framework consisting of seven DSR guidelines with the primary goal of helping researchers understand the DSR approach in IS research. Their work discusses DSR as a research paradigm but does not propose a process for performing DSR [53].

Peppers et al. [54] sought to bridge the gap by designing a DSR process through a synthesis of prominent previous DSR works by [52, 55, 56] amongst others. The result of the synthesis is a nominal process model consisting of six activities. The six activities are defined as follows:

1. *Problem identification and motivation.* To define a specific research problem and to justify a solution. This activity produces knowledge of the state of the problem and the importance of a solution.
2. *Objectives of a solution.* To deduce the objectives of a solution from the problem definition. This can be referred to as the research objectives. This activity produces knowledge of the state of the problem and its current solutions.
3. *Design and development.* To create the artefacts solution. This activity refers to the functionality, architecture and creation of the artefact. The outcome of this activity is knowledge of the theory that can be used as the solution.
4. *Demonstration.* To demonstrate the competency of the designed artefact as a solution to the problem. This includes experimentation or simulation using self-collected data or public data-sets. The outcome of this activity is knowledge of how to use the artefact to solve the problem.
5. *Evaluation.* To examine and measure how well the artefact supports a solution. This activity involves observing the results of demonstrating the artefact and analysing it with regards to the objectives. The activity requires knowledge of relevant metrics and analysis techniques. The nature of the research will dictate whether an iteration is required back to activity 2 or 3 or leave subsequent improvements to future projects.
6. *Communication.* To communicate the problem and its importance, with regards to the utility and the design of the artefact, the rigour of its design, and its effectiveness to researchers and other relevant audiences when appropriate. The output is often a structured empirical process, i.e., an experiment, described in a research paper.

Figure 3.2 illustrates the DSR process model of Peffers et al. The model is structured in a sequential manner with four possible entry points. The DSR process may start at any activity from 1–4 depending on the type and nature of the research, and then move on to the next activity.

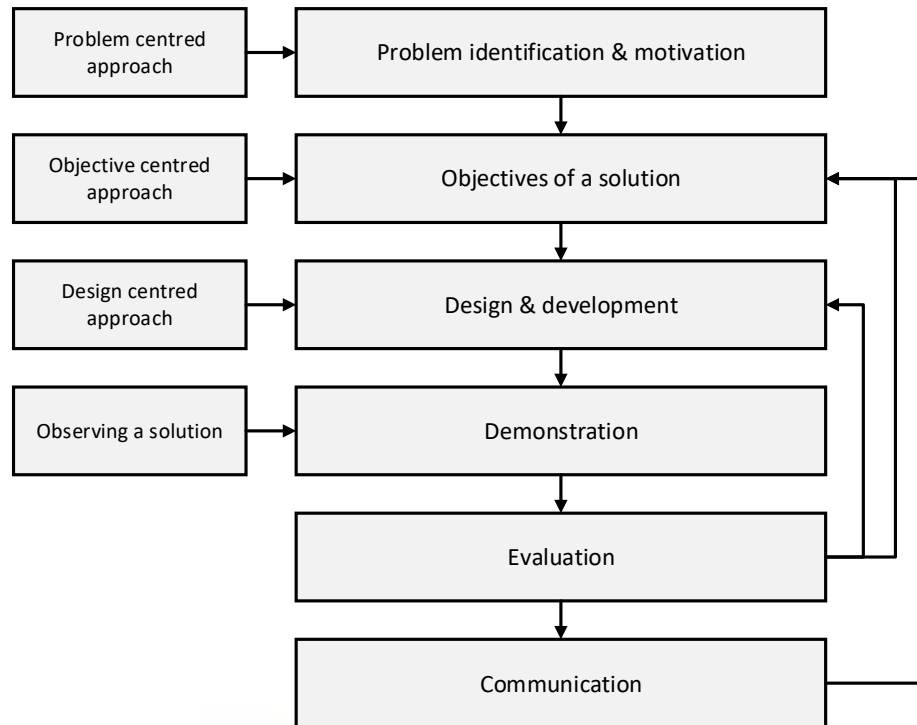


FIGURE 3.2: Design science research process model [54]

### 3.3 The Design of This Research

This research takes a problem centred approach as illustrated in Figure 3.3. The definition of the problem and motivation are discussed in Chapter 1.

Research objectives are detailed in Chapter 1. The related work in Chapter 2 supports and strengthens the insights towards building an artefact as a solution.

The design and development of the artefact and demonstration of the artefact is discussed in the later subsections. The system is evaluated in Chapter 5. Communication of the process and its results are through a conference paper [57] as well as through this thesis.

#### 3.3.1 Artefact Design and Development

This section discusses the design and development of the artefact as illustrated in Figure 3.4. The proposed artefact is structured using the general composition of a FER system namely: face detection, feature extraction and classification. To meet the objectives of the research, the artefact designed consists of two phases and follow a top-down approach, as depicted in Figure 3.5.



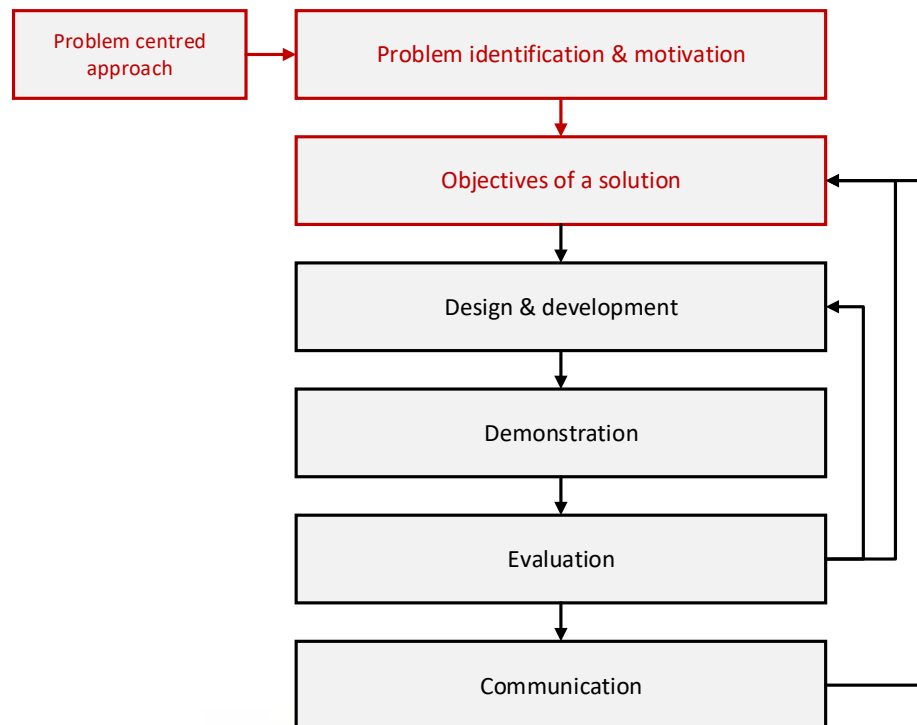


FIGURE 3.3: DSR process model highlighting activities 1 and 2 of Section 3.2

In phase one, the FER system is built with the: Viola-Jones face detection; LBP, CLBP and HOG feature extraction methods; and the SVM for classifier. Phase one is designed to compare the feature extraction methods and train, model and test the SVM. The objective is to select which feature extraction method is better suited for FER. The SVM is adopted as the classifier in this phase due to its prominence in FER research, its efficiency, generalisation capabilities and because it also limits the scope of the research by limiting the number of comparisons.

In phase two, the FER system is built with the preferred isolated feature extraction method based on the results of phase one and the ANN and RF machine learning techniques. Phase two is designed to train, model and test the ANN and RF. The objective is to compare the SVM results from Phase one to those for the ANN and RF.

As part of the design, two datasets were used in this research namely, the seven-class BU-3DFE [24] and the six-class CK [23] datasets. The CK dataset is considered to be a standard in FER research, however it is considerably smaller than the BU-3DFE with respect to labelled data. The use of the BU-3DFE dataset was thus considered important as it contains more labelled data. Both datasets were requested and permission was granted to download the datasets via a link given by the respective owners.

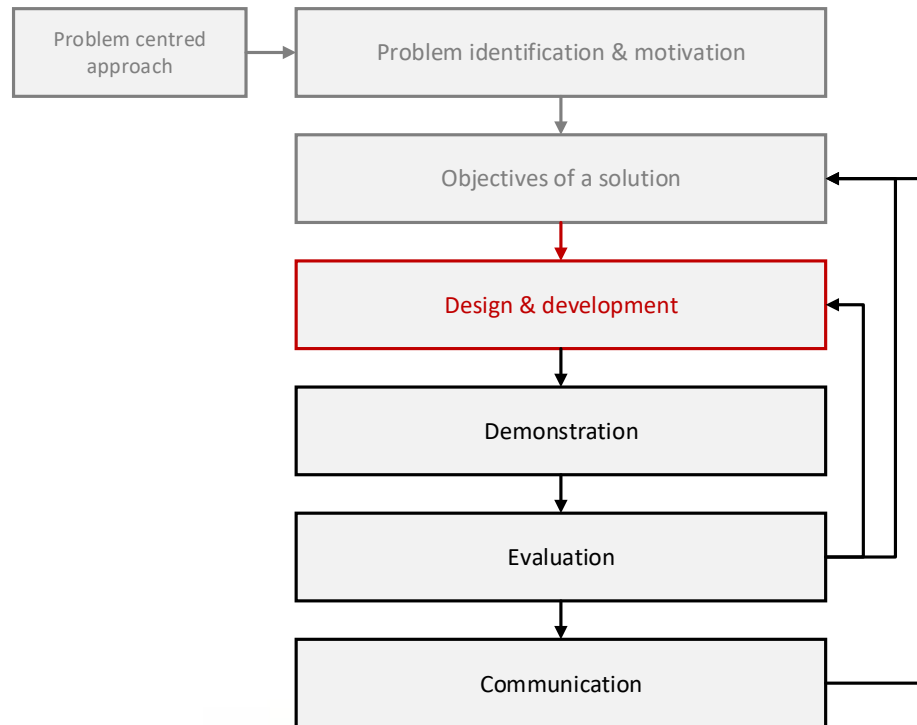


FIGURE 3.4: DSR process model highlighting activity 3 of Section 3.2

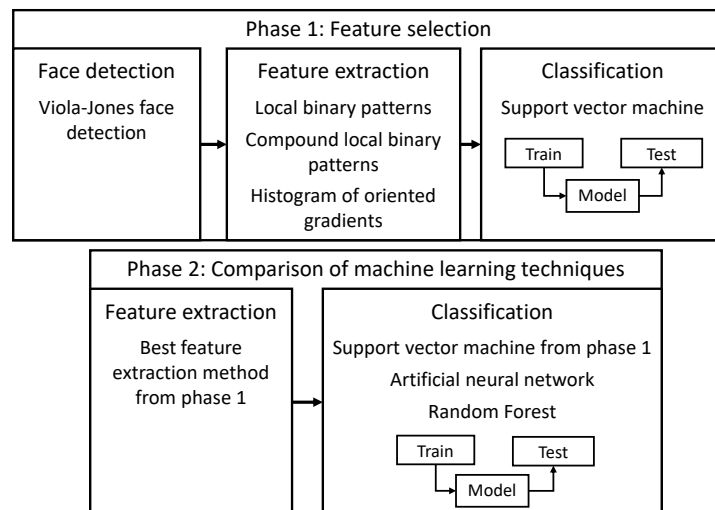


FIGURE 3.5: Proposed FER artefact

Two evaluating procedures were used. The first procedure consisted of splitting the BU-3DFE dataset into a training set and a testing set. However, the CK dataset was used solely as a test set. The motivation for this approach was to demonstrate how well the system generalises on completely unseen data taken under completely different conditions. This procedure was followed to evaluate the FER system's performance in Phase one and Phase two.

The second evaluation procedure consisted of using only the CK dataset. The FER system was evaluated using a  $k$ -fold cross-validation strategy and was only performed in Phase one. As shown in works referenced in Chapter 2, the  $k$ -fold cross-validation strategy is implemented by many researchers to evaluate the performance of their FER systems.

The next section demonstrates the artefact implementation by detailing the datasets used and the associated algorithms.

### 3.3.2 Artefact Demonstration

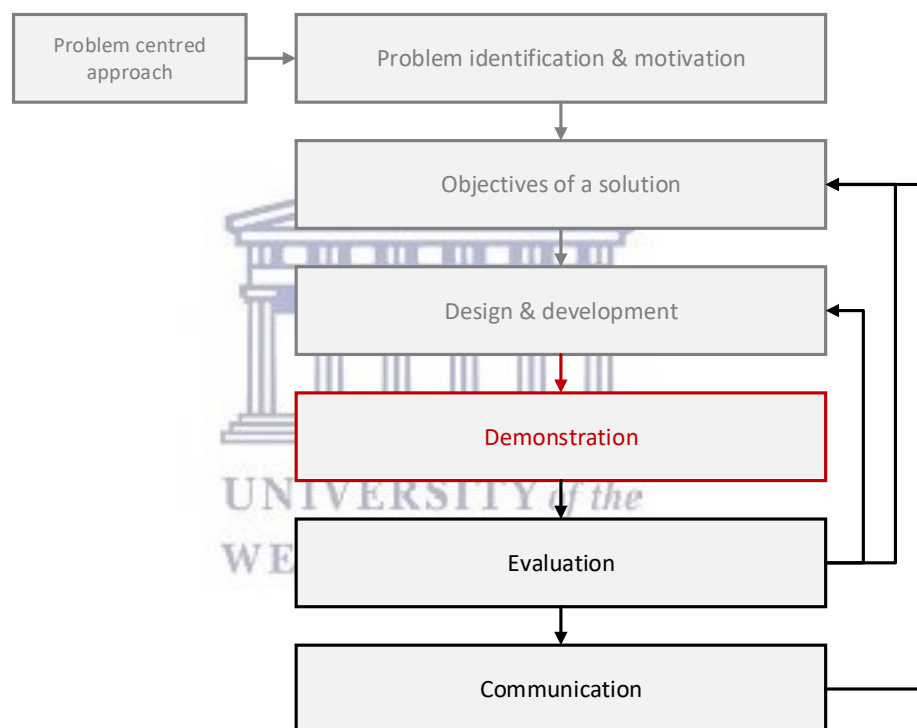


FIGURE 3.6: DSR process highlighting activity 4 of Section 3.2

This section demonstrates the use of the artefact as illustrated in Figure 3.6. In the subsequent sections the datasets and the algorithms associated with the designed artefact are discussed.

#### **BU-3DFE Dataset—Train and Test sets**

The Binghamton University 3D Facial Expression (BU-3DFE) dataset [24] contains 2D and 3D images of subjects posing with the six prototypic facial expressions: happiness,

sadness, surprise, anger, disgust, fear and the neutral pose. The facial expressions are posed by 100 male and female subjects of various ethnicities and age groups.

The 2D data consists of four images of each expression per subject. The four images illustrate different intensities of the subject's expression. The initial image illustrates the least intense pose of the expression and the last the most intense pose of the expression.

Only the most intense 2D expressions are used in this research. As mentioned above the data was randomly split into a training set and a test set. The training set consists of 70 subjects performing each of the seven facial expressions resulting in a total of 490 images. The remaining 30 subjects are used as test data, resulting in a total of 210 images.

#### **CK Dataset—Test set**

The CK dataset [23] consists of 123 subjects of different ethnicities. The subjects expressed six prototypic emotions and a seventh expression labelled as contempt. The dataset contains sequences of images varying in length of subjects performing one or more facial expressions. Each image sequence starts from a neutral face to the peak formation of a the labelled expression.

Each subject performed one or more of the facial expressions and some sequences were not labelled as one of the prototypic expressions. This is due the subject's formation of an expression not always being clear. The inconsistencies contribute to an unevenly distributed dataset.

Only the final image in a labelled sequence was used. The final image represents the peak formation of the expression. This results in a six-class dataset which contains 307 images. Table 3.1 summarises the distribution of the chosen images in the dataset across each of the facial expressions. The CK final dataset is used exclusively for testing in this research.

The FER artefact illustrated in Figure 3.5 is used to structure the subsequent sections. The implementation FER artefact will be discussed top-down starting at the face detection algorithm and ending with the classification algorithm.

TABLE 3.1: Distribution of labelled expressions in the CK dataset

Expression	Number of images
Anger	43
Disgust	59
Fear	25
Happiness/Joy	68
Sadness	28
Surprise	84
<b>Total</b>	<b>307</b>

### Face Detection

Face detection is the initial step in the feature selection phase as highlighted in Figure 3.7).

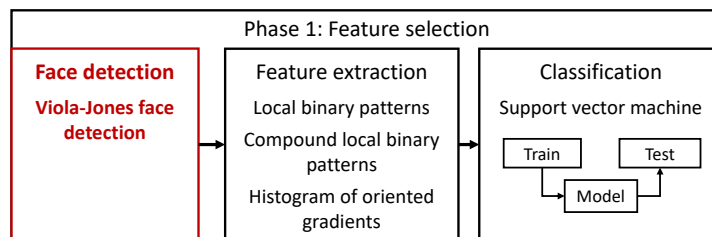


FIGURE 3.7: Phase 1: Feature selection - Face detection

The face is detected using the Viola-Jones algorithm [30]. The algorithm detects and delimits faces in images using weak-features known as Haar features. Haar features are a set of simple rectangles which contain “dark” and “light” areas. Figure 3.8 illustrates the two, three and four rectangular Haar features.

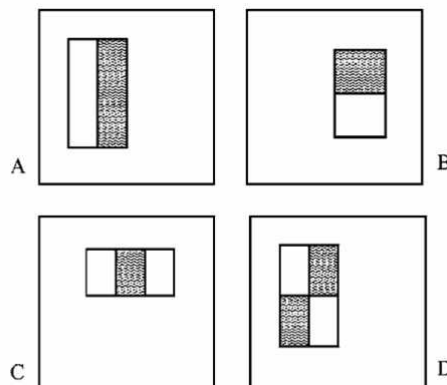


FIGURE 3.8: Forms of Haar features used in the Viola-Jones algorithm

The value of a single feature is calculated by subtracting the sum of pixels under the “dark” rectangle from the sum of pixels under the “light’ rectangles. The Haar features

are computed at varying scales and positions within an image sub-window. To speed up computation an image representation called the integral image is used to simplify the summing operation which allows the features to be computed in a constant time.

Among the features calculated, many are irrelevant. A modified Adaboost classifier is used to select important features that represent the face forming, a weighted sum of weak-classifiers. The process removes all irrelevant features by finding the best threshold to classify the face at a low error rate. To further increase performance, a cascade of classifiers is constructed.

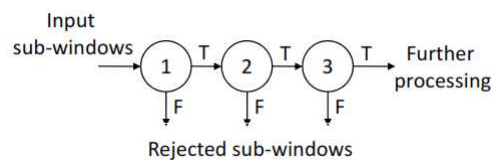


FIGURE 3.9: Cascade of classifiers [30]

The purpose of the cascade is to check whether the facial features are present within a sub-window of the image without having to evaluate each of the weak-classifiers. Figure 3.9 visually illustrates the cascade of classifiers. A sub-window of the image is used as input and each one of the weak-classifiers within the sub-window is tested. If one of the weak-classifiers does not represent the face the entire sub-window is rejected.

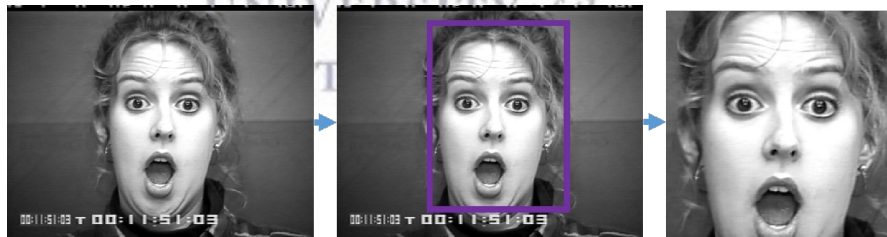


FIGURE 3.10: Face detection applied to input image

Figure 3.10 illustrates the detection of the face from the original input image. Once detected the face is cropped, isolated and resized for further processing.

### Local Binary Patterns

Following face detection, the next step is to implement the feature extraction methods highlighted in Figure 3.11).

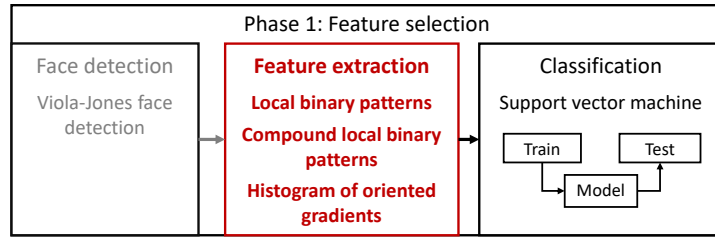
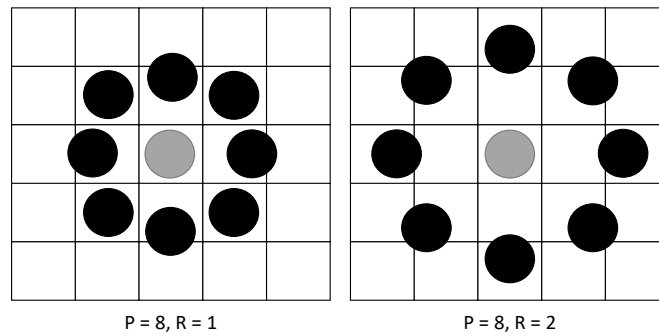


FIGURE 3.11: Phase 1: Feature selection - Feature extraction

An extension of the LBP operator is applied to the face image to capture details of varying scales. The extension allows the LBP operator to handle variable neighbourhood sizes. To account for the changes considered, two parameters were introduced: the number of points  $P$  in a circular neighbourhood and the radius of the circle  $R$ . Figure 3.12 illustrates an example of varying points  $P$  and radius  $R$  used to construct LBP. Parameters  $R = 2$  and  $P = 8$  provide an accurate representation of a facial image [12].

FIGURE 3.12: Examples of varying number of points  $P$  and radius  $R$  of the LBP operator

LBP is computed by comparing the neighbouring pixels to the centre pixel of the neighbourhood. Mathematically the computation of the LBP operator is represented by Equation 3.1 and the threshold function in Equation 3.2 where  $i_p$  is the neighbouring pixel value and  $i_c$  is the centre pixel.

$$LBP_{P,R}(i_c) = \sum_{p=0}^{n-1} 2^p f(i_p, i_c), \quad (3.1)$$

$$f(i_p, i_c) = \begin{cases} 0 & i_p < i_c, \\ 1 & i_p \geq i_c. \end{cases} \quad (3.2)$$

Figure 3.13 illustrates the computation of the LBP operator with a neighbourhood size of  $5 \times 5$  and parameters  $R = 2$  and  $P = 8$ . The comparison produces a binary representation of the LBP value which is then converted to decimal and used to form the LBP image.

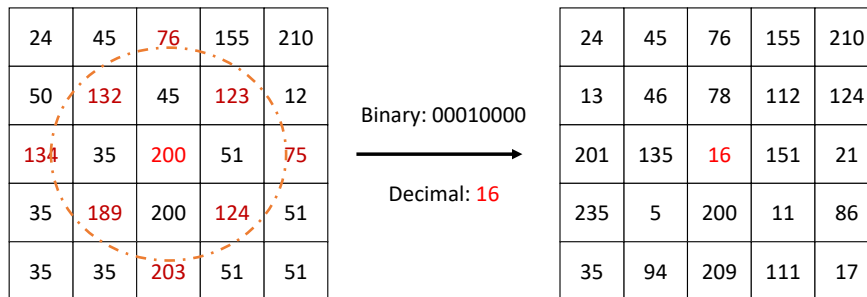


FIGURE 3.13: The LBP operator with parameters  $R = 2$  and  $P = 8$  applied to image in pixel representation

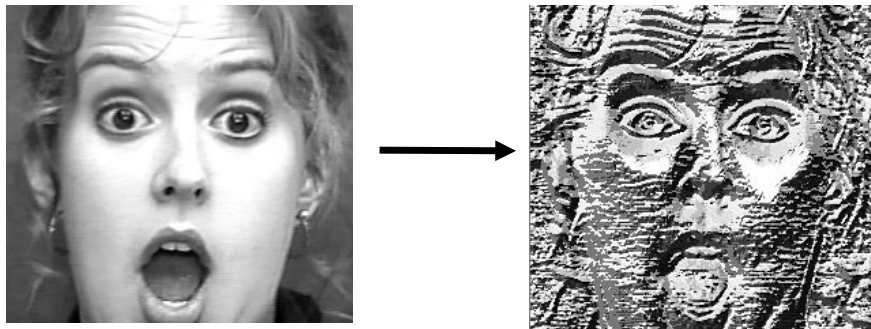


FIGURE 3.14: Original facial image (left) and LBP image (right)

The uniform patterns extension is applied to the LBP. A LBP is considered uniform if there are two or less transitions from 1–0 or 0–1 in the binary representation. Figure 3.15 illustrates a binary representation containing two transitions which is considered as a uniform pattern. The number of neighbours  $P$  determines the number of uniform patterns. For  $P = 8$  neighbours, there are a total 57 uniform patterns. All non-uniform patterns are combined and represented as one pattern—the 58th pattern. The uniform pattern extension thus considerably reduces the size of the feature vector [33].

**00010000**

FIGURE 3.15: Example of a uniform pattern



The LBP image is equally partitioned into non-overlapping cells to consider the shape information of the face. Histograms are extracted from each of the cells. The histogram is built using the range 0–58 representing uniform patterns. The final feature vector is formed by concatenating each of the histograms to form one spatially enhanced histogram representing the LBP image. The process of partitioning the facial image and forming the final feature vector from the histograms is illustrated in Figure 3.16.

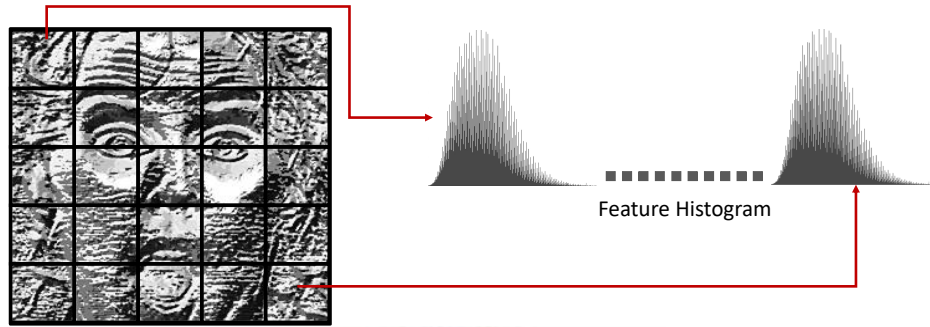


FIGURE 3.16: Formation of the LBP feature vector

### Compound Local Binary Patterns

CLBP assigns  $2P$  bits within a local neighbourhood of  $P$  pixels. The first bit mirrors the computation of the LBP operator whereby the centre pixel is compared to a neighbouring pixel and threshold using Eqn. 3.1. The second bit is computed by comparing the average magnitude of the neighbouring pixels  $M_{\text{avg}}$  to the absolute difference between the centre pixel and its neighbouring pixel. The CLBP thresholding function is defined as follows:

$$f(i_p, i_c) = \begin{cases} 11 & \text{if } i_p \geq i_c, |i_c - i_p| \leq M_{\text{avg}}, \\ 10 & \text{if } i_p \geq i_c, |i_c - i_p| > M_{\text{avg}}, \\ 01 & \text{if } i_p < i_c, |i_c - i_p| \leq M_{\text{avg}}, \\ 00 & \text{otherwise,} \end{cases} \quad (3.3)$$

here  $i_p$  is the pixel intensity of the neighbouring pixel and  $i_c$  the pixel intensity of the centre pixel in the neighbourhood. The average magnitude of the local neighbourhood  $M_{\text{avg}}$  represents the sum of all the neighbouring pixel intensities  $\sum_{p=1}^P i_p$  divided by the number of neighbours  $P$ . Figure 3.17 illustrates the generation of the CLBP code.

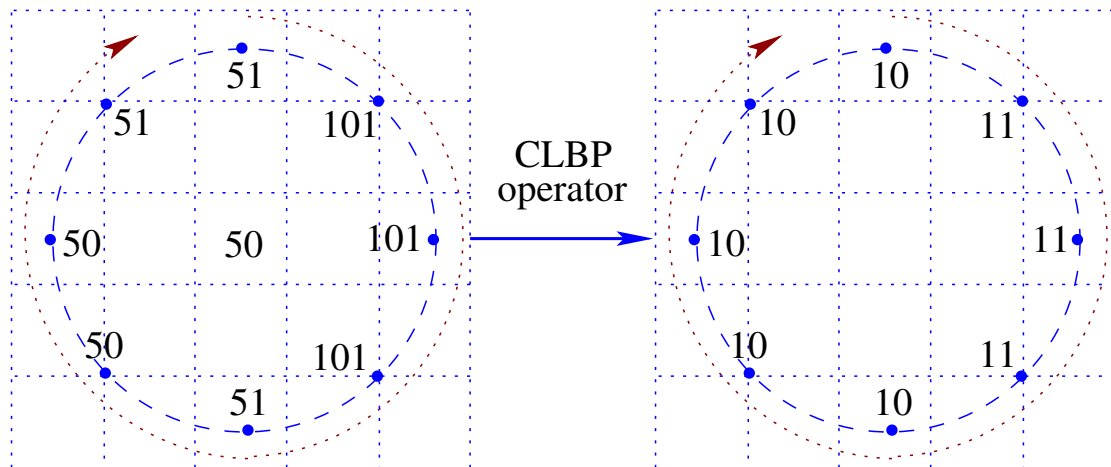


FIGURE 3.17: Illustration of the generation of the CLBP code

Similar to the LBP, the neighbourhood size and the parameters  $P$  and  $R$  may be extended with the CLBP. Using the parameters  $P = 8$  and  $R = 2$ , the CLBP is generated, resulting in a 16-bit binary code. The 16-bit binary code is divided into two equal sub-CLBP codes of eight-bits in length. The first eight-bit code is formed by merging the binary threshold values of the south, north, west and east neighbours. The second 8-bit code is comprised the remaining neighbours' threshold values. Each sub-CLBP code is then converted to decimal to form two sub-CLBP images. Figure 3.18 illustrates the generation of the sub-CLBP codes.

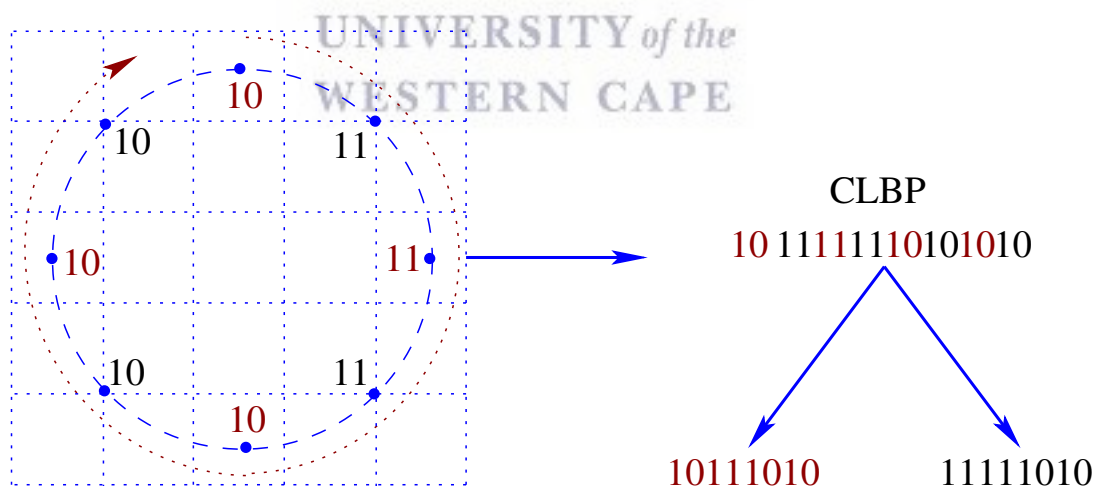


FIGURE 3.18: Generation of the sub-CLBP code

Similar to the LBP, the sub-CLBP images are then partitioned into equal cells and a histogram is built from the pixel intensities of each cell. The histograms are then concatenated for each of the sub-CLBP images. The final feature vector is produced by concatenating the histograms of the two sub-CLBP images.

## Histogram of Oriented Gradients

The HOG represents features by expressing the direction of edges and change thereof to capture contour, silhouette and some texture information. The first step involves calculating the gradient of the image. The gradients of the image are calculated for both the  $x$  and  $y$  directions separately. The result is two gradient images  $G_x$  and  $G_y$ , i.e.,

$$G_x = M_x * I, \quad G_y = M_y * I. \quad (3.4)$$

Where  $M_x$  and  $M_y$  are 1-D gradient-centred-point descriptive masks, where  $M_x$  is  $[-1, 0, 1]$  and  $M_y$ ,  $[-1, 0, 1]^T$ . The masks are convoluted over the facial image  $I$  producing gradient images  $G_x$  and  $G_y$ . The next step involves calculating gradient magnitude and orientation using the following Equations 3.5 and 3.6:

$$\sqrt{G_x^2 + G_y^2}, \quad (3.5)$$

$$\theta = \arctan \frac{G_y}{G_x}. \quad (3.6)$$

The image of gradient orientations is then divided into small regions called cells. An orientation histogram is built with nine bins uniformly spread over 0–180° representing unsigned gradients. Each pixel within the cell casts a vote based on its orientation and is placed into the appropriate bin, thus forming a basic orientation histogram.

To handle variations in contrast and illumination, cells are grouped together into blocks which are normalised separately. According to [38] block overlap greatly improves accuracy. The overlap between consecutive blocks is 50% which allows each cell to contribute more than once to the final descriptor. Figure 3.19 illustrates three facial images: the first image is divided into equally sized cells; the second into blocks made up of  $2 \times 2$  cells and lastly the 50% block overlap.

The L2-Hys method is used to normalise each block's histogram. L2-Hys is computed by taking the L2-norm, clipping the result and then re-normalising the block, as in [58]. All the normalised blocks are concatenated to form the final HOG feature vector.

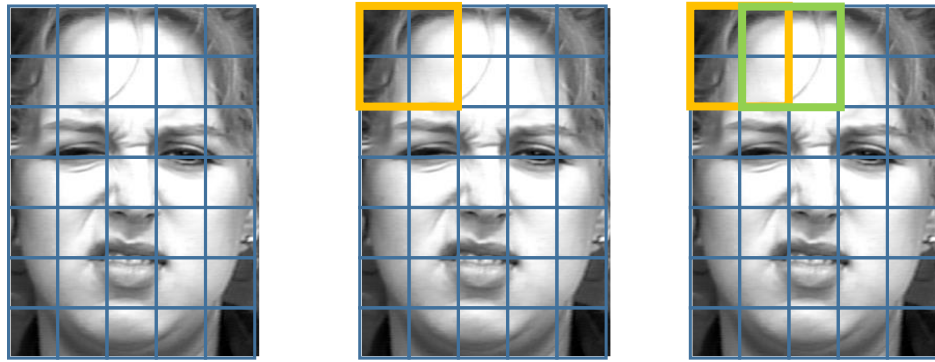


FIGURE 3.19: Facial image partitioned into cells (left), blocks formed from  $2 \times 2$  cells (centre) and 50% block overlap (right)

### Support Vector Machine

Following feature extraction methods, the next step is to implement the SVM as highlighted in Figure 3.20).

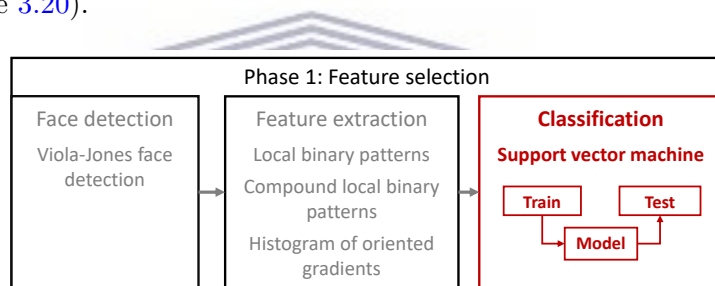


FIGURE 3.20: Phase I: Feature selection - Classification

A SVM is a supervised machine learning technique which performs classification by constructing a decision hyper-plane to separate two class labels. Figure 3.21 illustrates a Cartesian plane of two linearly separable classes of observations represented by dots and diamonds separated by a hyper-plane. Since many orientations of the hyper-plane exist, the goal of the classification process is to find the optimum hyper-plane which has the maximum margin between classes.

The maximum margin offers an optimisation problem which is dealt with by building a buffer zone around the hyper-plane. The class label points which define the margin are called ‘support vectors’. The SVM establishes the optimum hyper-plane by penalising the support vectors which fall on the opposite side of the hyper-plane indicated by the red arrows illustrated in Figure 3.21.

In most cases data points are not entirely separable. In order to separate classes in higher dimensions that are not linearly separable by the SVM classifier, a kernel function is

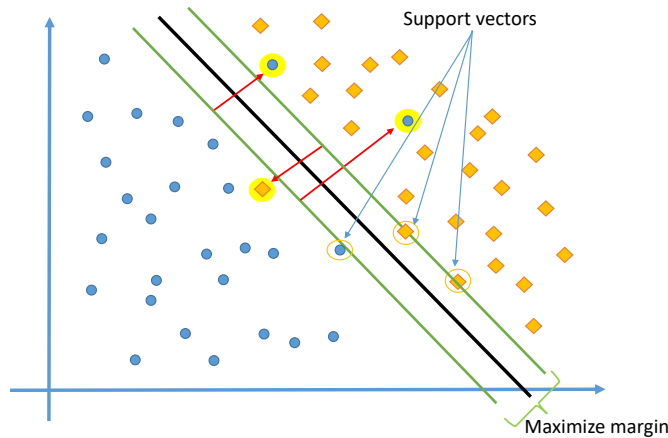


FIGURE 3.21: Two classes separated by a hyper-plane in SVM

used to map data. The kernel choice is problem-dependent. Given training samples  $\{x_i \in \mathbb{R}^p, i = 1, \dots, n\}$ , in two classes, and an indicator vector  $y \in \{1, -1\}$ , the support vector machine [42, 59] solves the following:

$$\min_{w,b,\zeta} \frac{1}{2} w^T w + C \sum_{i=1}^n \zeta_i, \text{ subject to } y_i(w^T \phi(x_i) + b) \geq 1 - \zeta_i, \text{ and } \zeta_i \geq 0, i = 1, \dots, n. \quad (3.7)$$

where  $w$  is the normal vector,  $b$  the interim term and  $\zeta$  the slack variable. Its dual is:

$$\min_{\alpha} \frac{1}{2} \alpha^T Q \alpha - e^T \alpha, \text{ subject to } y^T \alpha = 0, \text{ and } 0 \leq \alpha_i \leq C, i = 1, \dots, n. \quad (3.8)$$

where  $e$  is the vector of all ones,  $C > 0$  is the upper bound,  $Q$  is an  $n \times n$  positive semi-definite matrix,  $Q_{ij} \equiv y_i y_j K(x_i, x_j)$ , where  $K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$  is the kernel. Here training vectors are implicitly mapped onto a higher—perhaps infinite—dimensional space by the function  $\phi$ . Studies have indicated that the RBF kernel is a reasonable first choice [60]. As such, the RBF kernel is used in this research which is given as:

$$K(x_i, x_j) = \exp(-\gamma(\|\bar{x}, \bar{x}'\|_2^2)). \quad (3.9)$$

The decision function is:

$$\text{sgn}\left(\sum_{i=1}^n y_i \alpha_i K(x_i, x) + \rho\right). \quad (3.10)$$

SVMs are by definition limited to solving two-class problems but can be modified to

handle multi-class classification. Multi-class classification techniques involve a combination of several binary classifiers along with a strategic decision to choose a single class. To achieve multi-class classification, a one-against-one (OAO) [34, 61] approach is used. The approach creates a series of binary classifiers trained with each possible pair of classes. For the  $n$  multi-class problem, where  $n > 2$ ,  $\frac{(n-1)}{2}$ , binary classifiers are created. The final prediction is determined by which class received the most votes.

The training and optimising of the SVM is carried out using the grid-search function of the library for support vector machines (LibSVM) [62]. The grid-search function exhaustively iterates through  $C$  and  $\gamma$  hyper-parameter values of the SVM and RBF kernel. The  $C$  parameter indicates the cost of classification.  $C$  determines the width of the margin of the decision hyper-plane, as to how much it avoids misclassifying training examples. The  $\gamma$  parameter defines how far the influence of a single training example reaches.

A total of 110 different exponentially growing sequences of  $(C, \gamma)$  are traversed ranging between  $C = 2^{-5}, 2^{-3}, \dots, 2^{15}; \gamma = 2^{-15}, 2^{13}, \dots, 2^3$ . For each combination of  $(C, \gamma)$ , a five-fold cross-validation scheme is used. The  $(C, \gamma)$  combination with the highest cross-validation accuracy is selected.

The optimisation of the three feature descriptors is carried out in parallel with the optimisation of the SVM. The process of determining optimal parameters for each of the feature descriptors is a process of trial and error. The parameters optimised for LBP and CLBP are the resolution of the facial image and the cell sizes. The parameters optimised for the HOG descriptor are the cell size, block size and resolution of the facial image.

### **Artificial Neural Network**

Following the classification of the SVM, the next step is to implement the ANN and RF which forms part of phase 2 as highlighted in Figure 3.22

The multi-layer perceptron (MLP) is a class of ANN used to perform classification. A perceptron consists of one or more artificial neurons arranged in a specific pattern [63]. Figure 3.23 illustrates the most basic perceptron consisting of one neuron. The perceptron receives inputs  $x_1, \dots, x_n$ , multiplies them with the corresponding weights

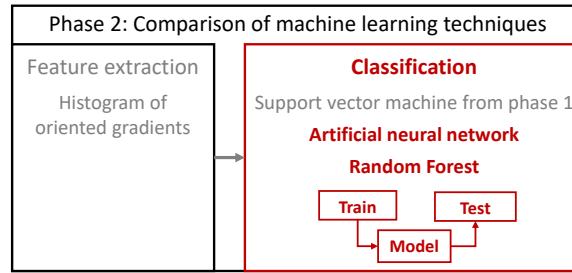


FIGURE 3.22: Phase 2: Comparison of machine learning techniques - Classification

$w_1, \dots, w_n$  and sums the result which is passed through an activation function producing the output.

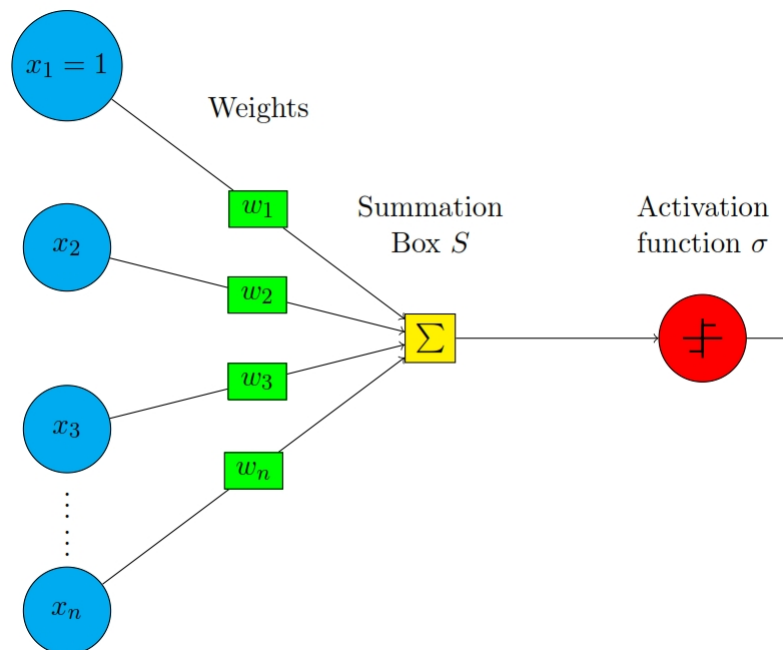


FIGURE 3.23: A simple perceptron consisting of one neuron [8]

The weighted sum is given by:

$$S = \sum_{i=1}^n w_i x_i. \quad (3.11)$$

The choice of activation function is problem dependent. There is no clear method of selecting an activation function. Selection is a process of trial and error. The MLP trains using backpropagation. More precisely, it trains using some form of gradient descent and the gradients are calculated using backpropagation.

The neural network is built by adding layers of perceptrons together, forming a multi-layer perceptron. The network consists of an input layer which receives features as input and an output layer which creates the resulting outputs. Hidden layers are the layers

in between the input and output layers. The hidden layer does not directly ‘see’ the feature inputs or outputs. Each layer can apply any function from the previous layer to produce an output. Hidden layers are constantly adjusted during the training of the neural network to transform the values of the inputs. The final hidden layer transfers the values to the output layer which transforms the values to outputs. Figure 3.24 illustrates a multilayer perceptron with one hidden layer.

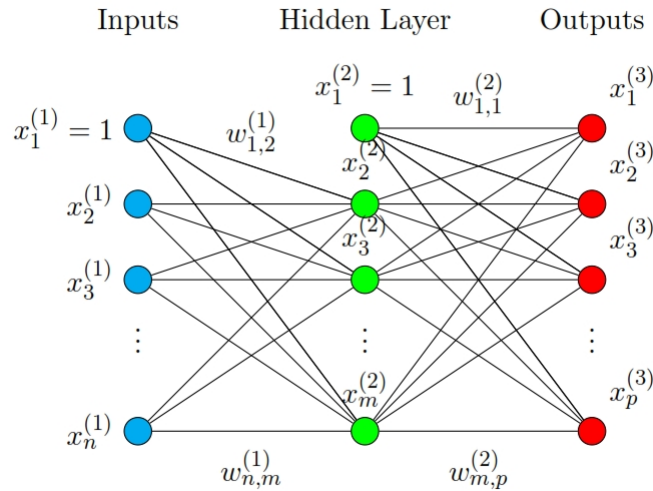


FIGURE 3.24: A multi-layer perceptron with one hidden layer [8]

The formal definition of a MLP is given as follows [63]: Given a set of training data  $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$  where  $x_i \in \mathbf{R}^n$  and  $y_i \in \{-1, 1\}$ , a one hidden layer one hidden neuron MLP learns the function  $f(x) = W_2 g(W_1^T x + b_1) + b_2$  where  $W_1 \in \mathbf{R}^m$  and  $W_2, b_1, b_2 \in \mathbf{R}$  are model parameters.  $W_1, W_2$  represent the weights of the input layer and hidden layer, respectively; and  $b_1, b_2$  represent the bias added to the hidden layer and the output layer, respectively.  $g(x) : \mathbf{R} \rightarrow \mathbf{R}$  is the activation function, set by default as the hyperbolic tan function. It is given as:

$$g(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}. \quad (3.12)$$

For binary classification,  $f(x)$  passes through the logistic function  $g(z) = \frac{1}{(1+e^{-z})}$  to obtain output values between zero and one. A threshold, set to 0.5, would assign samples of outputs larger or equal to 0.5 to the positive class, and the rest to the negative class.

If there are more than two classes,  $f(x)$  itself would be a vector of size  $(n_{classes})$ . Instead of passing through a logistic function, a soft-max function is used instead, which is



written as:

$$\text{softmax}(z) = \frac{\exp(z_i)}{\sum_{l=1}^K \exp(z_l)}. \quad (3.13)$$

where  $z_i$  represents the  $i$ th element of the input to soft-max function, which corresponds to class  $i$ , and  $K$  is the number of classes. The result is a vector containing the probabilities that sample  $x$  belong to each class. The output is the class with the highest probability.

MLPs use different loss functions depending on the problem type. For regression, MLP uses the Square Error loss function. This is expressed as:

$$\text{Loss}(\hat{y}, y, W) = \frac{1}{2} \|\hat{y} - y\|_2^2 + \frac{\alpha}{2} \|W\|_2^2, \quad (3.14)$$

beginning random initial weights, the MLP minimizes the loss function by constantly updating the weights. After computing the loss, a backward pass propagates it from the output layer to the previous layers, providing each weight parameter with an update value meant to decrease the loss. In gradient descent, the gradient  $\nabla \text{Loss}_W$  of the loss with respect to the weights is computed and deducted from  $W$ . This is expressed as:

$$W^{i+1} = W^i - \epsilon \nabla \text{Loss}_W^i. \quad (3.15)$$

where  $i$  is the iteration step, and  $\epsilon$  is the learning rate with a value larger than 0. The algorithm stops when it reaches a pre-set maximum number of iterations; or when the improvement in loss is below a certain small number.

Using the training data a grid-search is used to tune the hyper-parameters of the ANN and  $k$ -fold cross validation as the evaluation metric. For the scope of the research only one hidden layer was considered in the neural network. The grid-search is made up of  $n$  the number of neurons in the hidden layer which range from 1 – 100, additional to the typical number of neurons which used in MLP which range from 5 – 50 [64], and four activation functions. The activation functions are as follows:

- Identity/Linear activation function:

$$f(x) = x. \quad (3.16)$$

- Logistic/Sigmoid activation function:

$$f(x) = 1/(1 + \exp(-x)). \quad (3.17)$$

- Hyperbolic tan function (Tanh):

$$f(x) = \tanh(x). \quad (3.18)$$

- Rectified linear unit function (Relu):

$$f(x) = \max(0, x). \quad (3.19)$$

The grid-search exhaustively iterates through combinations of the activation function and numerous neurons resulting in total of 400 different combinations. For each combination a five-fold cross-validation scheme is used to evaluate the performance. The combination with the highest cross-validation accuracy is selected to train the model.

### Random Forest

A RF is comprised of a group decision trees to form a forest of trees [47]. In order to understand the RF algorithm it is important to understand the underlying decision tree structure.

Decision trees are a class of supervised learning algorithms associated with directed acyclic graphs. Decision trees use a top-down approach to classify input data. The decision tree poses a set of pre-defined questions associated with features of the inputs. Each question posed corresponds to a single node within the tree and each successive question is dependent on the criteria of the previous question. Thus in a classification context, a decision tree is a hierarchy which outputs a predicted class based on the terminal node produced.

Figure 3.25 illustrates an abstracted decision tree which contains a set of nodes and edges forming an organised hierarchy. The decision tree structure consists of two main types of nodes. Internal/split node—which include the root node—represented by circles and terminal/leaf nodes represented by squares in Figure 3.25. A boolean function or

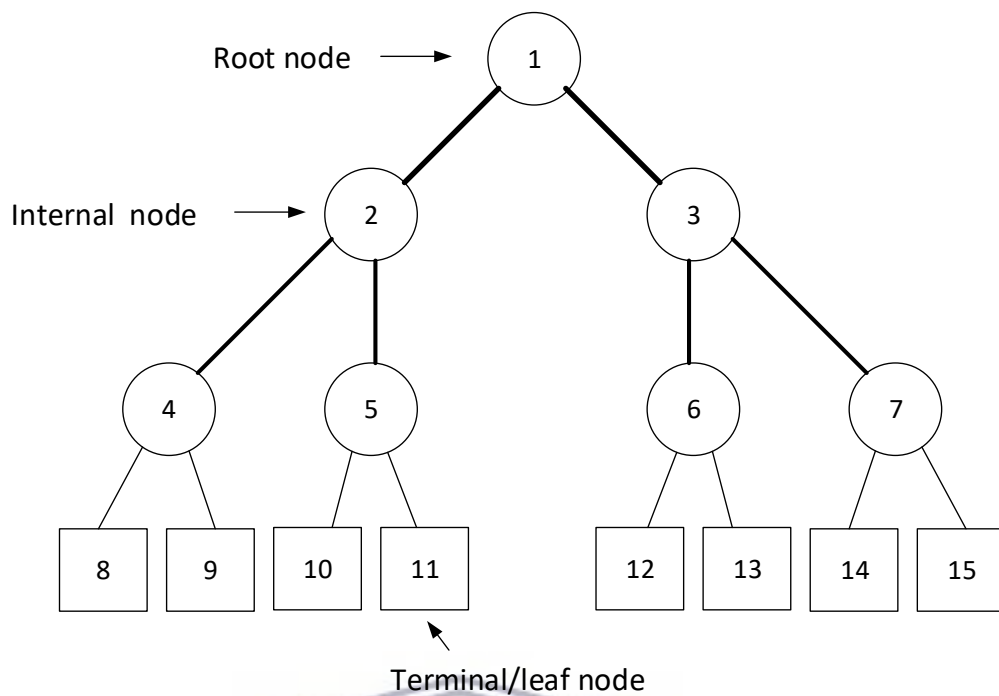


FIGURE 3.25: The structure of a simple decision tree

question is used at each internal node, the answer to which determines to which path the input is directed. The terminal node indicates the predicted output.

On their own, decision trees are very simple and poor classifiers [47] as they tend to over-fit data. However, ensemble methods that query and vote on various aspects of the data, can be used as a foundation to form a powerful classification technique with the RF. Ensemble methods group together ‘weak learners’, which are simple low accuracy predictors, to form a ‘strong learner’ capable of high-accuracy predictions. In terms of the ensemble, the decision trees represent the ‘weak learners’, which collectively form the ‘strong learner’, the RF classifier.

Breiman defines the RF classifier as follows [47]:

$$R = T_k(X, \Theta_k), k = 1, \dots, K. \quad (3.20)$$

Where the RF is denoted as  $R$  and  $K$  the collection of decision tree classifiers.  $T_k$  denotes the  $k$ -th tree in the forest and  $\Theta_k$  is a set of independently distributed samples used to generate unique decision trees. Each decision tree is considered unbiased and votes for the most popular class given input  $X$ .

The algorithm for the RF is as follows [64]: Given a training set of  $L$  labelled points  $\{(x_i, y_i) | i = 1, \dots, L\}$  where  $x_i$  represents a single training example and  $y_i$  is its class label,  $K$  decision trees are generated.

For each decision tree, given  $N$  the dataset of samples, a subset of the samples  $n$  are extracted using bootstrapping. Each subset of  $k$  samples are uniform and randomly selected without replacement. Bootstrapping is a process whereby unique random subsets of a dataset are used to train different classifiers. Furthermore,  $m_s$  features are chosen randomly from a total of  $M$  features available. During the construction of the tree, at each node, the best split positions are chosen from  $m_s$ , the random selection features. The nodes are then split into child nodes and iterated until the tree reaches a depth of  $D_s$  equal to a threshold  $D_{min}$ . The entire algorithm is repeated for each tree  $B$  resulting in a RF  $T_k(X, \Theta_k), k = 1, \dots, K$ .

The algorithm is formally defined as follows given the symbols defined above [64]:

---

**Algorithm 1** Random Forest algorithm

---

- 1: **for**  $k = 1$  to  $K$  **do**
  - 2:     Create a bootstrap sample  $Z^*$  of size  $n$  to build tree  $T_k$
  - 3:     **while** Nodes in current tree  $D_s < D_{min}$  **do**
  - 4:         Choose  $m_s$  features at random from the total  $M$  features
  - 5:         Pick the best feature split of  $m_s$  features
  - 6:         Split into 2 child nodes
  - 7:     Output Random forest  $T_k | k = 1, \dots, K$
- 

Using the training data a grid-search is used to tune the hyper-parameters of the RF and  $k$ -fold cross validation as the evaluation metric. For the scope of the research the tree depth and number of trees were considered as adjustable hyper-parameters in the neural network. The grid search consists of a number of trees in the range 1–100 and a tree depth of 1–20.

The grid-search exhaustively iterates through combinations of the chosen hyper-parameters. This results in a total of 2000 different combinations for the RF. For each combination a five-fold cross-validation scheme is used as the evaluation metric. The combination with the highest cross-validation accuracy is selected to train the model.

### 3.4 Conclusion

This chapter discussed the research design and methodology. The research design process leads to the use of the DSR methodology. The methodology was discussed and its implementation is presented throughout the thesis. The design, development and demonstration of the research artefact were also discussed. The next chapter discusses the evaluation of the artefact.



## Chapter 4

# Results and Analysis

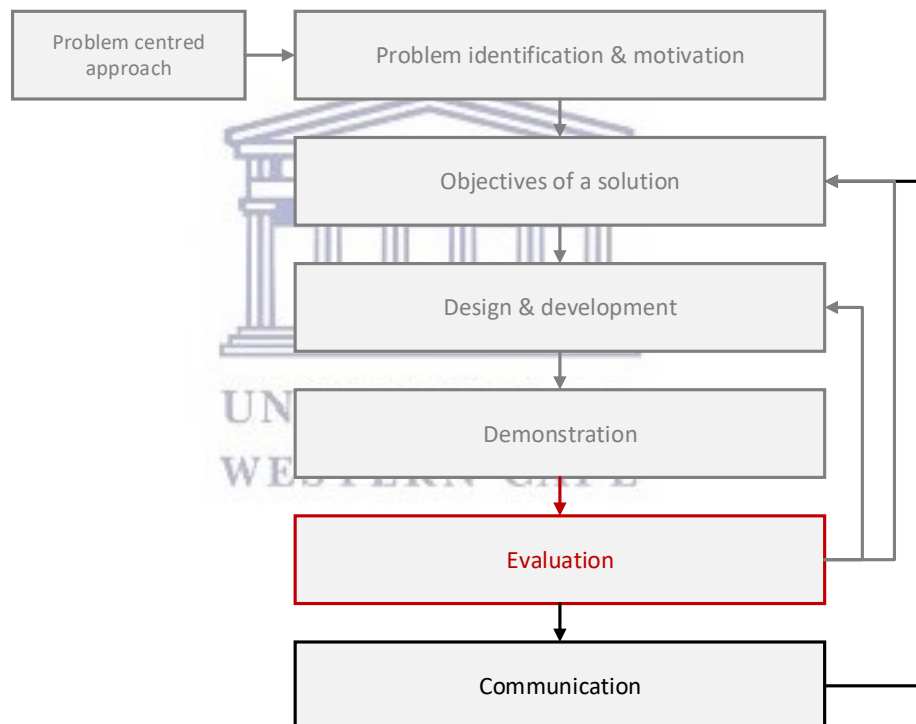


FIGURE 4.1: DSR process highlighting activity 5 of Section 3.2

This Chapter signals the start of the activity 5 of the DSR process (see 4.1 and provides an in depth evaluation of the designed artefact and by doing so provides the answers to the research questions outlined in Chapter 1. The Chapter is structured as follows: Section 4.1 describes the optimisation and testing of phase one—the feature selection process and Section 4.2 describes the optimisation and testing of phase two and provides the comparisons of machine learning techniques.

Accuracy is measured by dividing the total number of correctly classified images by the total number of images in the set. Where applicable, confusion matrices is used to analyse the difference between classes. In some cases for ease of reference the facial expressions are abbreviated in this chapter. When assessing accuracy, experiments were carried out on an Intel i7 3.8 GHz CPU with 16 GB DDR4 RAM and a GeForce GTX 580 GPU—3GB RAM. The operating system used was Windows 10.

## 4.1 Evaluation of Phase 1: Feature Selection

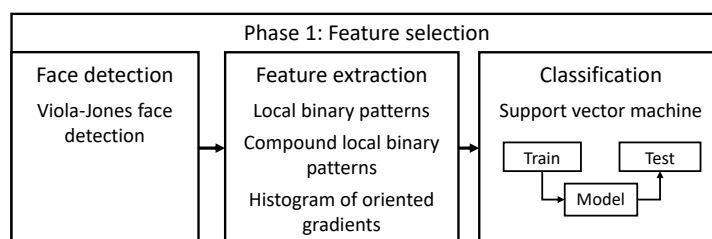


FIGURE 4.2: Phase 1: Feature selection - Evaluation

Phase 1 consisted of two evaluating procedures. The first procedure consisted of splitting the BU-3DFE dataset into a training set and a testing set. The training set was used to optimise the feature extraction techniques and the RBF parameters of SVM simultaneously as described in Section 3.3.2. The optimisation and the results are described in sub-sections 4.1.1– 4.1.3.

The optimised model of each feature extraction technique was evaluated and compared on the CK and BU-3DFE test sets. The results using the test sets of the BU-3DFE and CK datasets is described in Section 4.1.4 and 4.1.5. The test sets will be evaluated on overall accuracy, robustness toward facial expressions and test subjects and lastly generalising capability. In Section 4.1.6 the CK dataset is tested with a cross-validation scheme. The final choice of feature descriptor which will be used in phase two is discussed in Section 4.1.7.

### 4.1.1 Optimisation of the LBP

A grid-search approach was used in optimising the parameters of the LBP descriptor. The parameters optimised were the resolution of the facial image  $r$  and the cell sizes  $c$ .

The ranges of the resolution  $r$  were 40, 50 and 60 pixels and the ranges for the cell size  $c$  were 5 and 10 pixels. These arbitrary values were chosen due to their ease of divisibility and to limit the scope of the research.

TABLE 4.1: LBP Cross-validation optimisation scores (%)

$r = 40 \times 40$			$r = 50 \times 40$			$r = 60 \times 40$		
$c$	5	10	$c$	5	10	$c$	5	10
5	63.87	63.26	5	53.4	52.4	5	49.18	45.90
10	62.88	60.41	10	55.1	53.67	10	49.10	45.30

$r = 40 \times 50$			$r = 50 \times 50$			$r = 60 \times 50$		
$c$	5	10	$c$	5	10	$c$	5	10
5	61.60	58.75	5	63.06	63.88	5	55.71	57.34
10	59.80	55.91	10	64.20	62.04	10	55.71	55.71

$r = 40 \times 60$			$r = 50 \times 60$			$r = 60 \times 60$		
$c$	5	10	$c$	5	10	$c$	5	10
5	61.60	60.61	5	63.06	63.88	5	65.10	<b>66.70</b>
10	61.83	56.50	10	64.2	62.04	10	63.26	66.12

Table 4.1 summarises the cross-validation accuracies for the optimisation of the LBP operator. Table 4.1 shows that the optimal parameters resolution size was  $60 \times 60$  at a cell size of  $5 \times 10$  yielding a score of 66.70%. The optimum  $(C, \gamma)$  combination that provided this accuracy was  $(2^7, 2^{-15})$ . These parameters were used to train a final SVM model specific to the LBP feature descriptor.

#### 4.1.2 Optimisation of the CLBP

The CLBP operator was optimised in a way similar to that used for the LBP operator in terms of the resolution width and height combinations of 40 and 60 pixels, and cell sizes of 5 and 10 pixels.

Table 4.2 summarises the cross-validation scores for the CLBP operator. As illustrated in the table, the optimum resolution width and height were  $60 \times 60$  with a cell size of  $10 \times 10$ . These parameters yielded a cross validation score of 68.57%, which was obtained with the SVM hyper-parameters  $C = 2^3$  and  $\gamma = 2^{-15}$ . These parameters were used to train a final SVM model specific to the CLBP feature descriptor.



TABLE 4.2: CLBP Cross-validation optimisation scores (%)

$r = 40 \times 40$			$r = 50 \times 40$			$r = 60 \times 40$		
$c$	5	10	$c$	5	10	$c$	5	10
5	64.69	64.08	5	55.10	54.49	5	48.36	46.12
10	65.10	62.86	10	54.49	64.89	10	49.38	45.10

$r = 40 \times 50$			$r = 50 \times 50$			$r = 60 \times 50$		
$c$	5	10	$c$	5	10	$c$	5	10
5	57.96	58.77	5	65.31	66.93	5	54.69	56.73
10	57.34	57.75	10	63.67	64.89	10	54.90	55.30

$r = 40 \times 60$			$r = 50 \times 60$			$r = 60 \times 60$		
$c$	5	10	$c$	5	10	$c$	5	10
5	63.88	61.02	5	63.06	63.88	5	62.04	66.53
10	62.45	64.89	10	64.2	62.04	10	65.91	<b>68.57</b>

### 4.1.3 Optimisation of the HOG

The parameters optimised were the width and height of the image, i.e., the resolution size of the facial image  $r$  along with the cell size  $c$  and block dimensions  $b$  for the HOG descriptor. Resolution sizes  $r$  of  $64 \times 64$  and  $128 \times 128$  were considered. To reduce the complexity of the parameter search space, only square cells and blocks were considered, i.e, cell and block sizes of the same width as the height. The cell sizes  $c$  considered were  $4 \times 4$ ,  $8 \times 8$  and  $16 \times 16$ . The block dimensions  $b$  considered were  $2 \times 2$  and  $4 \times 4$ .

TABLE 4.3: HOG cross-validation optimisation scores (%)

$r = 128 \times 128$				$r = 64 \times 64$			
$b$	$c$			$b$	$c$		
	$4 \times 4$	$8 \times 8$	$16 \times 16$		$4 \times 4$	$8 \times 8$	$16 \times 16$
$2 \times 2$	45.71	56.12	70.81	$2 \times 2$	56.94	70.61	68.37
$4 \times 4$	57.14	65.51	<b>75.10</b>	$4 \times 4$	72.04	72.04	63.88

Table 4.3 summarises the cross-validation scores for the HOG descriptor. The results show that the highest cross-validation accuracy is 75.10% at a resolution of  $128 \times 128$  along with parameters  $c = 16 \times 16$  and  $b = 4 \times 4$ . The optimum SVM hyper-parameter combination yielded was  $C = 2^{11}$  and  $\gamma = 2^{-5}$ . These parameters were used to train a final SVM model specific to the HOG feature descriptor.

#### 4.1.4 BU-3DFE Test Set Results

Table 4.4 outlines the total number of correctly classified frames and accuracies of each feature extraction method using the BU-3DFE test set. Each of the three feature extraction methods performed well with overall accuracies of 60% and above. Table 4.4 shows that the HOG descriptor was the top-performer followed by CLBP and LBP.

TABLE 4.4: Performance of each feature descriptor on BU-3DFE test set

Descriptor	Total frames	Correct	Accuracy (%)
LBP	210	126	60.0
CLBP	210	136	64.8
HOG	210	139	66.2

It is noted that there is a slight discrepancy between the training set and test set results. This is due to the test set being completely exclusive of the training set and the similarity of certain expressions in samples of the test set which will be discussed later.

To provide insight into the robustness towards classes, Figure 4.3 highlights the performance of each descriptor across each facial expression class. Each facial expression class is comprised of 30 images. The accuracies are derived by dividing the number of correctly predicted images per class by the total number of images in the class.

Figure 4.3 shows that the most accurately predicted classes are ‘Happy’ and ‘Surprise’. Both classes performed well across the feature descriptors. ‘Surprise’ and ‘Happy’ are considered to be visually distinct and are somewhat easier to perform than most of the other facial expressions. ‘Fear’ and ‘Sad’ were generally the most difficult to recognise with ‘Fear’ possibly the most complex of the expressions to perform non-spontaneously.

HOG and CLBP are similar in accuracy across the majority of classes with both outperforming each other in two classes however HOG does so with a higher margin. In terms of accuracy, LBP was clearly eclipsed by both HOG and CLBP, however it has the lowest standard deviation in accuracies across classes—14%—implying that it is the most robust toward variations in classes. The standard deviation between classes was equally robust for HOG at 19% and CLBP at 16%. Considering that HOG and CLBP outperform LBP it is somewhat surprising that both descriptors are less robust towards variation in classes.

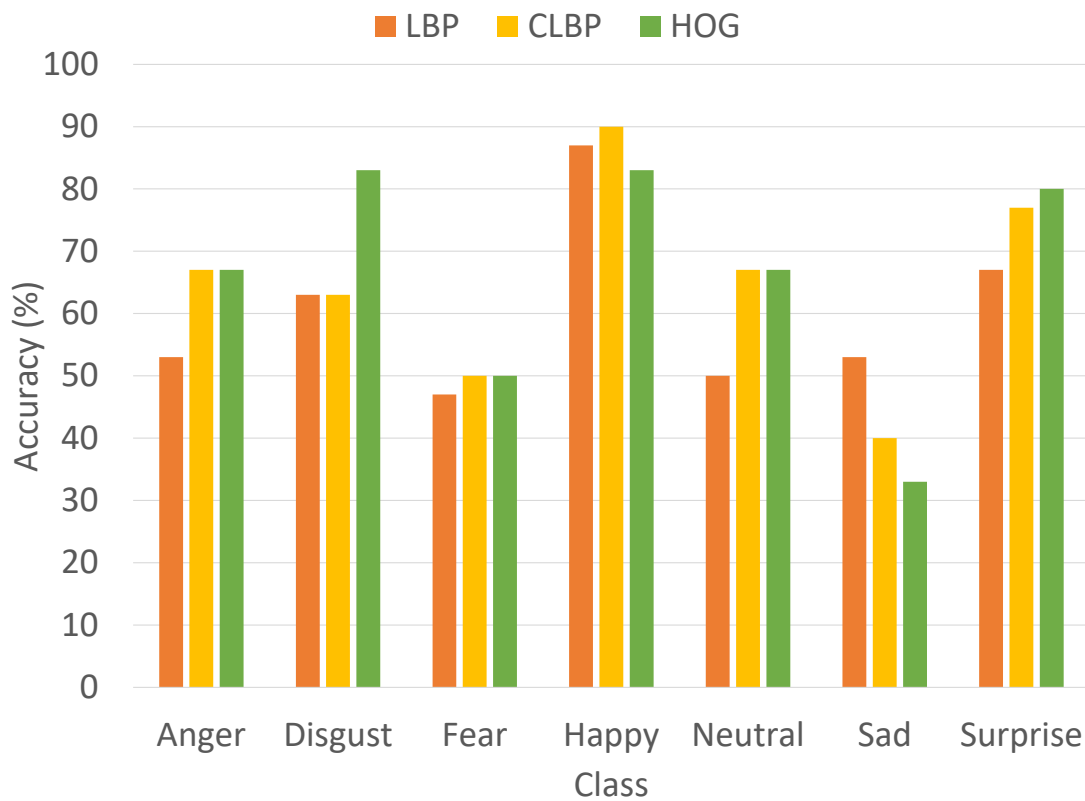


FIGURE 4.3: Performance of each feature descriptor per facial expression class on the BU-3DFE test set

To describe the performance of the classification model and highlight the variations across classes confusion matrices were built for each feature descriptor. The confusion matrix breaks down the performance of each facial expression class by analysing the correct and incorrect predictions with count values. For the sake of clarity, all confusion matrices are normalised and values of zero are left blank.

TABLE 4.5: Confusion matrix for LBP (%) (BU-3DFE)

Actual	Predicted						
	ANG	DISG	FEA	HAP	NEU	SAD	SURP
ANG	<b>53</b>	13	3		7	23	
DISG	17	<b>63</b>	10	7		3	
FEA	13	10	<b>47</b>	13	3	13	
HAP			10	<b>87</b>	3		
NEU	13		10	10	<b>50</b>	17	
SAD	27	7	13			<b>53</b>	
SURP		7	13	3	7	3	<b>67</b>

Table 4.5 is the confusion matrix for the LBP descriptor. It is observed that the expressions ‘Anger’ and ‘Sad’ are misclassified with one another. Likewise, ‘Disgust’ is

often confused with ‘Anger’. ‘Fear’ performs more erratically and is misclassified with a handful of the other classes.

TABLE 4.6: Confusion matrix for CLBP (%) (BU-3DFE)

Actual	Predicted						
	ANG	DISG	FEA	HAP	NEU	SAD	SURP
ANG	<b>67</b>	7	7	3	3	13	
DISG	20	<b>63</b>	3	3	7	3	
FEA	20	3	<b>50</b>	13		10	3
HAP	3			<b>90</b>			7
NEU	7		3	3	<b>67</b>	20	
SAD	11	3	10	3	7	<b>40</b>	
SURP	3	3	3	3	10		<b>77</b>

Table 4.6 is the confusion matrix for the CLBP descriptor. Upon inspection Table 4.6 is similar to that of Table 4.5: A similar trend is found for ‘Anger’ and ‘Sad’; a similar confusion between the ‘Disgust’ and ‘Anger’ classes; and ‘Fear’ is randomly spread across most of the classes.

TABLE 4.7: Confusion matrix for HOG (%) (BU-3DFE)

Actual	Predicted						
	ANG	DISG	FEA	HAP	NEU	SAD	SURP
ANG	<b>67</b>	10	10		3	7	3
DISG	10	<b>83</b>	3			3	
FEA	3.3	10	<b>50</b>	13	10	7	7
HAP	3		13	<b>83</b>			
NEU	3		27	10	<b>67</b>	13	
SAD	43	3	13		3	<b>33</b>	3
SURP		7	3	3	7		<b>80</b>

Table 4.7 is the confusion matrix for the HOG descriptor. Once again, similarities can be drawn with Table 4.6 and Table 4.7: ‘Sad’ predominantly confused with ‘Anger’; ‘Disgust’ is predominately misclassified as ‘Anger’; and ‘Fear’ is randomly misclassified as every other class. This indicates that the difficulty in classification may be attributed to incorrect labelling in the dataset.

Upon review of the test samples, some of the ‘Sad’, ‘Anger’ and ‘Disgust’ classes were found to be similar in appearance. Figure 4.4 displays samples of test subjects labelled as ‘Anger’ in the top row and ‘Sad’ in the bottom row. Figure 4.4 illustrates that the images are visually similar, and one can be easily misjudged for the other producing incorrect classification. A similar case using different classes is represented in Figure 4.5

which depicts subjects expressing ‘Anger’ on the top row, and ‘Disgust’ on the bottom row.



FIGURE 4.4: Similarity of ‘Anger’ (top) and ‘Sad’ (bottom) expressions in the test set



FIGURE 4.5: Similarity of ‘Anger’ (top) and ‘Disgust’ (bottom) expressions

Therefore the fluctuation in classification accuracies between classes can be connected to the quality of the BU-3DFE dataset. A similar observation was made by Mushfieldt in [12] whereby he stated that many of the samples in the BU-3DFE dataset are incorrectly labelled, especially in the samples labelled as ‘Fear’.

The final performance analysis involves comparing the robustness of each descriptor to variations across test subjects. A graph containing a per test subject analysis turned out to be convoluted, therefore a histogram of the number of correctly recognised images across the test subjects was built instead. Meaning that each test subject was given a score out of seven, since the test data per subject, contains one image of each of the seven classes expressed. Figure 4.6 illustrates a histogram representing the spread of the scores across test subjects.

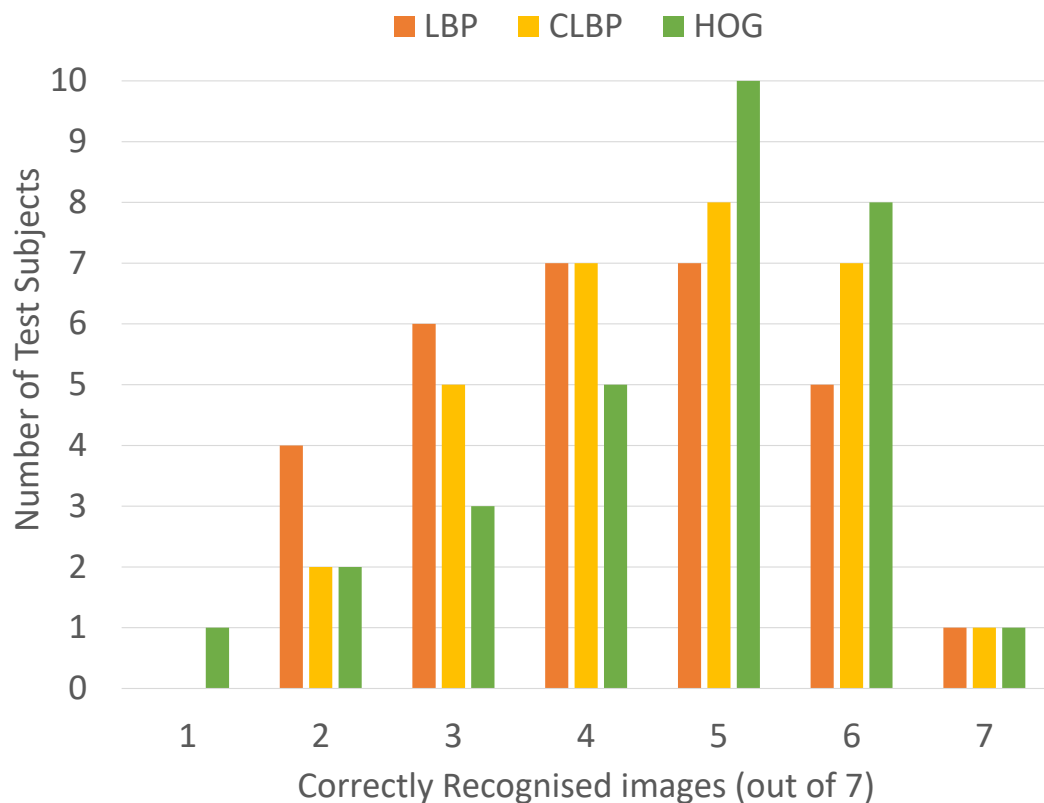


FIGURE 4.6: Histogram of the number of test subjects that achieved each number of correctly recognised images (out of seven) for each of the three feature descriptors

A distribution which is skewed to the right of the histogram would be ideal as it would represent better variation towards test subjects. This means that a large number of subjects would have more correctly recognised images. Figure 4.6 shows that for the LBP descriptor, the majority of the subjects achieved scores of three, four and five out of seven. Four subjects achieved a score for two out of seven, double that of the CLBP and HOG descriptors. However, CLBP provides greater robustness to test subjects compared to LBP, with the majority of test subjects distributed around the four, five and six out of seven mark. Lastly, HOG outperforms both LBP and CLBP, with the majority of test subjects scoring five or six out of seven. Furthermore, the histogram for LBP follows a normal distribution. A similar distribution is seen for CLBP, however, the spread of the results skew more to the right of the histogram. The HOG distribution peaks toward the right of the histogram which is ideal. Meaning HOG has a higher accuracy and more consistency per subject as compared to LBP and CLBP. Thus HOG is more robust towards test subjects than CLBP and LBP.

#### 4.1.5 CK Test Set Results

The optimised models trained for each feature descriptor were tested on the CK dataset. It should be noted that the CK dataset does not contain the neutral class, but the three models were trained to recognise this class. This test set tests the generalising capability of the model and consists of data which was captured under different conditions to the training set. It is also important to note that the number of images per-class are imbalanced.

TABLE 4.8: Overall recognition results for the CK dataset

Descriptor	Total frames	Correct	Accuracy (%)
LBP	307	189	61.56
CLBP	307	107	34.85
HOG	307	198	64.50

Table 4.8 summarises the number of correctly recognised frames for each feature extraction method on the CK dataset. Both LBP and HOG perform very well, with accuracies above 60%. This demonstrates the ability of the HOG and LBP descriptor to generalise well. The HOG descriptor outperforms both LBP and CLBP, with a high average accuracy of 64.50%. Surprisingly, the accuracy of CLBP is considerably lower than that of LBP and HOG. It is unclear why the performance of CLBP is lower on this dataset.

The classification accuracy of each facial expression class across all subjects and feature extraction methods is represented in Figure 4.7.

Figure 4.7 shows that CLBP achieves lower than 10% accuracy for three of the six classes and achieves an excellent accuracy of 81% for ‘Anger’. The general performance of the CLBP on this dataset is poor. LBP performs significantly better than the CLBP, with accuracies of 97% for ‘Happy’ and 89% for ‘Sad’. However, for four of the six classes, it scores an accuracy of below 45% which nevertheless is still better than CLBP. HOG once again proves to be superior to the LBP and CLBP, now in terms of generalisation potential. Four of the six classes achieve above 60% accuracy and one class above 40%.

HOG performed well for the expressions ‘Disgust’, ‘Happy’ and ‘Surprise’, with accuracies of 91%, 81% and 80% respectively. Clearly, HOG has excellent potential towards generalising, noting that the testing and training sets were captured and controlled under completely different conditions and with different subjects.

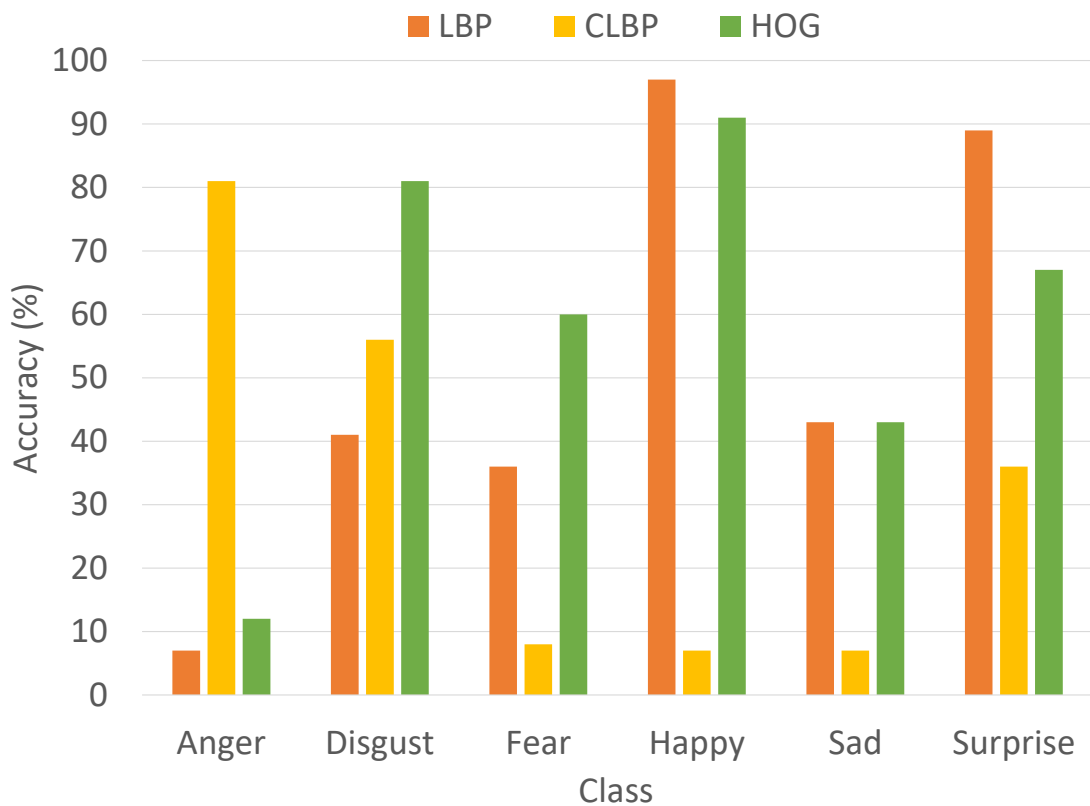


FIGURE 4.7: Accuracy of each feature descriptor per facial expression class across all subjects on the CK dataset

A per-subject analysis on the CK dataset was not possible due to the unbalanced nature of the dataset; there are large variations in the number of images across test subjects, and not all subjects performed all facial expressions.

#### 4.1.6 CK Cross-validation Test Results

It should be noted that a direct comparison of results is not possible due to the difference in evaluation metrics of the research. The cross-validation test was done solely for comparison purposes with researchers who have used cross-validation and the six-class CK dataset as their final evaluation criterion, however a direct comparison is still not possible in this case either.

A five-fold cross validation scheme was used alongside the grid-search to find the optimum SVM-RBF hyper-parameters for each of the feature extraction technique. It should be noted that the LBP, CLBP and HOG parameters used were the optimised values from



the BU-3DFE dataset. Table 4.9 illustrates the comparison of cross-validation accuracies.

TABLE 4.9: Comparison of cross-validation accuracies on CK dataset

Study	LBP (%)	CLBP (%)	HOG (%)	Optimisation Dataset	Total Test Images
Shan et al. [43]	88.4	-	-	CK	310
Ahmed et al. [37]	90.1	94.4	-	CK	1124
Gritti et al. [16]	90.9	-	92.7	CK	320
This research	84.4	85.7	91.2	BU-3DFE	307

Table 4.9 indicates that excellent cross-validation accuracies are attained across each feature descriptor. For this test each descriptors accuracy increases dramatically. HOG performs exceptionally well and similar to the trend in the above results is followed by CLBP and LBP.

As aforementioned a direct comparison between results is not be possible due to two factors. First, the CK dataset has images which are not labelled and this situation leads to inconsistent data. In the case of Ahmed, 1224 image sequences were used for classification compared to Shan with 310 images. Similarly, Gritti used 320 in their study and in this research 307 images were used. Second each of the studies optimised their FER systems for use specifically on the CK dataset, whereas in this research, the feature descriptors were implemented using the optimised values produced from BU-3DFE training set. Nevertheless, the results achieved by this research are excellent and comparable to the aforementioned studies for the HOG descriptor.

#### 4.1.7 Discussion and Choice of Feature Descriptor

The factors considered when comparing HOG, CLBP and LBP were: overall accuracies, accuracy across facial expressions, robustness towards test subjects, and ability to generalise. Table 4.10 summarise the results.

First, the overall accuracies on the BU-3DFE test set revealed that the HOG achieved the highest accuracy, exceeding the overall accuracy of the LBP by 6.2% and that of the CLBP by 1.4%. Thus HOG and CLBP could be considered comparable. However, the cross-validation accuracy on the CK test set revealed a large increase in accuracies for

TABLE 4.10: Summary of feature descriptor results

Dataset	Factor	LBP	CLBP	HOG
BU-3DFE	Overall Accuracy (%)	60.0	64.8	<b>66.2</b>
	Robust to Subjects	High	High	<b>Best</b>
	Robust to Classes	High	High	<b>Best</b>
CK	Cross-Validation Accuracy (%)	84.4	85.7	<b>91.2</b>
	Overall Accuracy (%)	61.56	34.85	<b>64.50</b>
	Robust to Classes	High	High	<b>Best</b>
	Generalisation Capability	High	High	<b>Best</b>

the feature descriptors. HOG outperforms both CLBP and LBP. Thus HOG emerges the winner in this regard.

Second, with regards to facial expression classes, CLBP and HOG were comparable, however, HOG had higher accuracies across most classes. LBP was outperformed by both HOG and CLBP. With regards to robustness to variations in test subjects, HOG emerged as the better descriptor. HOG achieved higher scores per-subject and was more consistent than both the CLBP and the LBP.

Finally, the ability to generalise to varied test data was considered by means of an ambitious use of a dataset that was completely different from the training set; the CK dataset. HOG emerged as the winner. The descriptor achieved high accuracies consistently for the majority of the facial expression classes despite the challenging nature of the test. Of note is that the LBP descriptor also proved to be robust however less so than the HOG. CLBP proved inferior to even the LBP, being unable to maintain an acceptable accuracy over most classes.

It seems evident to conclude that the HOG is a more robust feature descriptor than the LBP and the CLBP. Furthermore, the CLBP can generally be considered to be superior to the LBP, but the LBP has greater potential in terms of its ability to generalise.

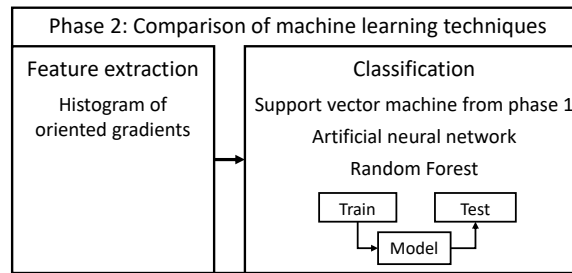


FIGURE 4.8: Phase 2: Comparison of machine learning techniques - Evaluation

## 4.2 Evaluation of Phase 2: Comparison of Machine Learning Techniques

As a consequence of the results in phase one, the HOG descriptor is selected as the feature descriptor of choice. The training and optimisation of the ANN and the RF using the HOG descriptor is described in Section 4.2.1 and 4.2.2. The comparison of the SVM, ANN and RF take place in Sections 4.2.3 and 4.2.4. The final choice of machine learning technique is discussed in Section 4.2.5.

### 4.2.1 Optimisation of the Artificial Neural Network

A grid-search was used to optimise the hyper-parameters of the ANN. Using one hidden layer, the parameters optimised were the number of neurons in the hidden layer along with an array of activation functions. The neurons used ranged from 1–100. The activation functions used were the: logistic, identity, tanh and relu activation functions. A five-fold cross validation scheme was used to assess the performance.

Figure 4.9 illustrates the optimisation results of the ANN. From Figure 4.9, the majority of activation functions stabilise with accuracies of 70%+ when the ANN models are trained with 25 or more neurons in the hidden layer. However, in the case of the logistic activation function, the performance drops intermittently regardless of the number of neurons in the ANN model. It is unclear why the fluctuation occurs.

The highest ranking combination of hyper-parameters is the identity activation function coupled with 47 neurons in the hidden layer resulting in a cross validation accuracy of 78.16%. These parameters are used to form the final ANN model.

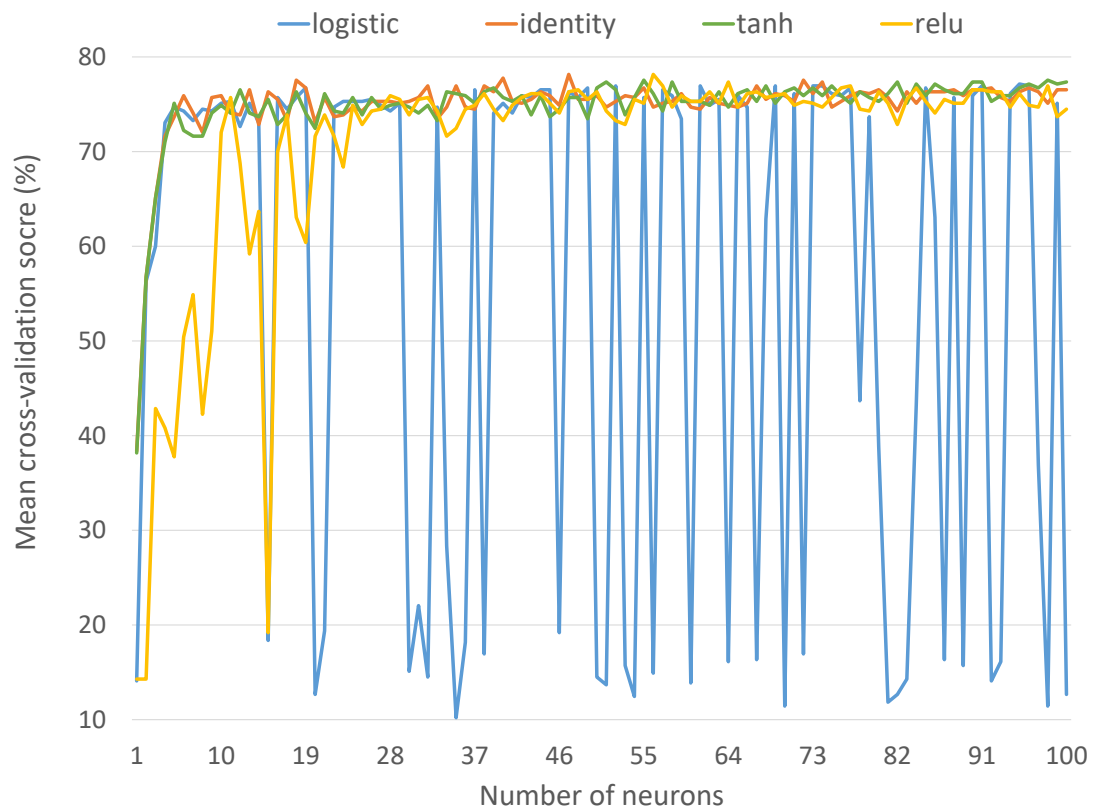


FIGURE 4.9: Optimisation results of artificial neural network

#### 4.2.2 Optimisation of the Random Forest

A grid-search was used to optimise the hyper-parameters of the RF. The parameters optimised were the depth of the tree and the number of trees. The depth of the tree ranged from 1–20 and the number of trees from 1–100. As with the optimisation of the SVM and ANN, a five-fold cross validation scheme was used to assess the performance.

Figure 4.10 illustrates the optimisation results of the RF. From Figure 4.10, it is clear that tree depths of 1–6, have considerably lower accuracies than depths  $> 6$ , which is to be expected due to the low complexity of such RFs.

The highest ranking combination of hyper-parameters is 13 for the depth of tree with 84 nodes, resulting in a cross validation accuracy of 73.06%. These parameters are used to form the final RF model.

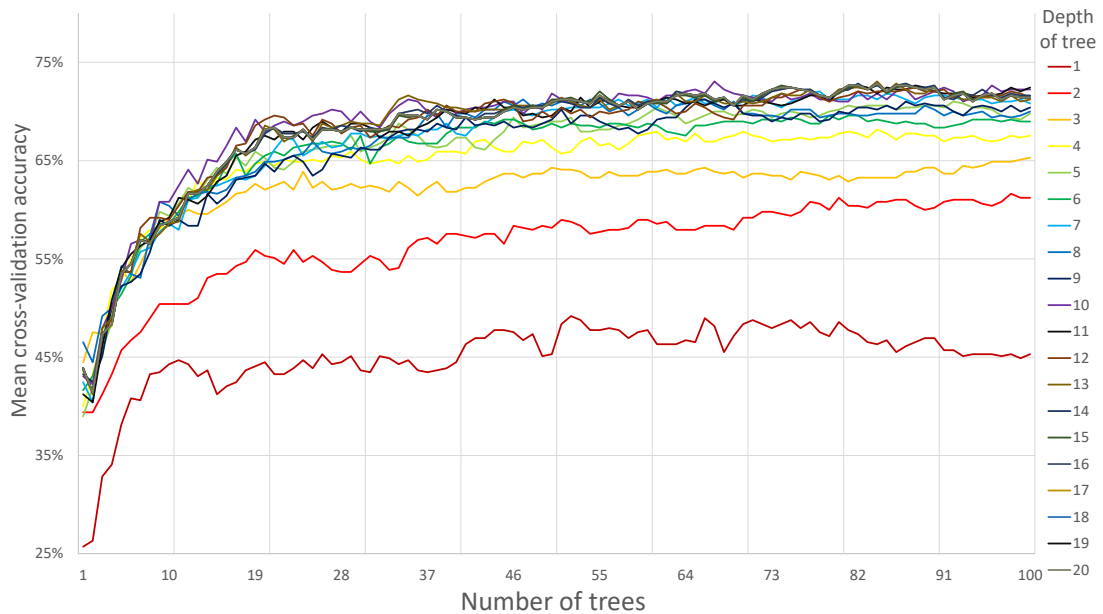


FIGURE 4.10: Optimisation results of the Random forest

TABLE 4.11: Performance of machine learning techniques on the BU-3DFE test set

Classifier	Total frames	Correct	Accuracy (%)
RF	210	133	63.33
ANN	210	138	65.71
SVM	210	139	66.20

### 4.2.3 BU-3DFE Test Set Results

Table 4.11 displays the total number of correctly classified frames and accuracies of each machine learning technique (classifier) used on the BU-3DFE test set. Each machine learning technique performed well with high overall accuracies. Table 4.4 shows that the accuracies of the SVM, ANN, and RF are quite comparable with a difference of six misclassified frames between them. It is noted that there is once again a slight discrepancy between the training set and test set results. The similarity in the discrepancies of the results affirm that the test set has incorrectly labelled samples. To further inspect the results, a histogram highlighting the performance of the facial expression classes was built.

Figure 4.11 provides a histogram showing the comparative performance of facial expression classes across machine learning techniques. Figure 4.11 shows that the ‘Happy’ and ‘Surprise’ achieve accuracies of 80% and above for each machine learning technique. The

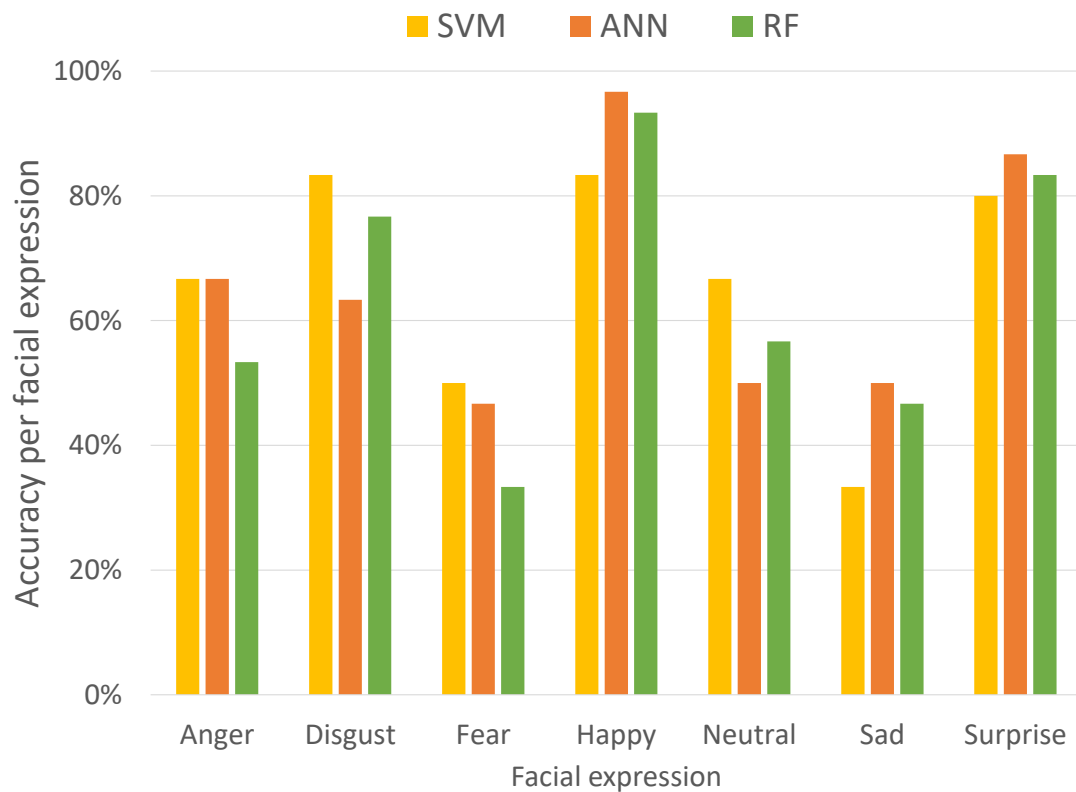


FIGURE 4.11: Facial expression accuracies across machine learning techniques

performances of both the ‘Fear’ and ‘Sad’ classes are under-par with accuracies of 50% and lower. A similar trend was found when comparing the feature descriptors.

The SVM and ANN have the highest accuracies in three classes, whilst the RF performs second best in five of the classes. The standard deviation across classes for the SVM and ANN are 19% and 22% for the RF. Furthermore, the ANN has a higher maximum accuracy 97% and a minimum accuracy of 47% compared to the SVM which has a maximum of 83% and a minimum of 33%. This implies that the ANN is more consistent and robust toward variations in classes followed by the SVM then the RF.

For further comparison confusion matrices were built for each classifier to highlight the variation in classes. For clarity of comparison the ‘neutral’ and ‘surprise’ classes were omitted from the confusion matrices. Tables 4.12, 4.13, 4.14 show the partitioned confusion matrix for the SVM, ANN and RF respectively.

The confusion matrices show clear trends. In the case of the SVM and the ANN the ‘Anger’ class is misclassified mostly as ‘Fear’ or ‘Disgust’. ‘Disgust’ is predominately misclassified as Anger across all classifiers. In the case of ‘Fear’, although it is quite spread

TABLE 4.12: Confusion matrix SVM (BU-3DFE)

Actual	Predicted				
	AN	DI	FE	HA	SA
AN	<b>67</b>	10	10		7
DI	10	<b>83</b>	3		
FE	3	10	<b>50</b>	13	7
HA			13	<b>83</b>	
SA	43	3	13	3	<b>33</b>

TABLE 4.13: Confusion matrix ANN (BU-3DFE)

Actual	Predicted				
	AN	DI	FE	HA	SA
AN	<b>60</b>	10	13	3	3
DI	27	<b>63</b>	7		3
FE	17		<b>47</b>	17	3
HA			3	<b>97</b>	
SA	30	3	13		<b>50</b>

TABLE 4.14: Confusion matrix RF (BU-3DFE)

Actual	Predicted				
	AN	DI	FE	HA	SA
AN	<b>53</b>	10			23
DI	13	<b>77</b>	7		
FE	13	3	<b>33</b>	23	3
HA			3	<b>93</b>	
SA	23	10			<b>47</b>

across most classes, it is misclassified as ‘Happy’ across classifiers. The ‘Happy’ class performs well across classifiers. Lastly ‘Sad’ is predominantly misclassified as ‘Anger’ across classifiers. From the evidence presented by the confusion matrices, it is safe to assume that the models act similarly towards the test data as the feature descriptors. This shows that some of the misclassification may not be due to the classifier, but may be due to the quality of the data as highlighted previously in Section 4.1.4.

For further analysis, as explained for Figure 4.6, a histogram is used to inspect the robustness of the machine learning techniques toward subjects. Figure 4.12 is a histogram illustrating the spread of scores for the seven expressions across test subjects per machine learning technique.

Figure 4.12 shows that the RF and SVM are normally distributed, but they both peak toward the right side of the histogram at a score of five. The histogram spread of the SVM is more skewed to the right than the RF. This indicates that the SVM is more

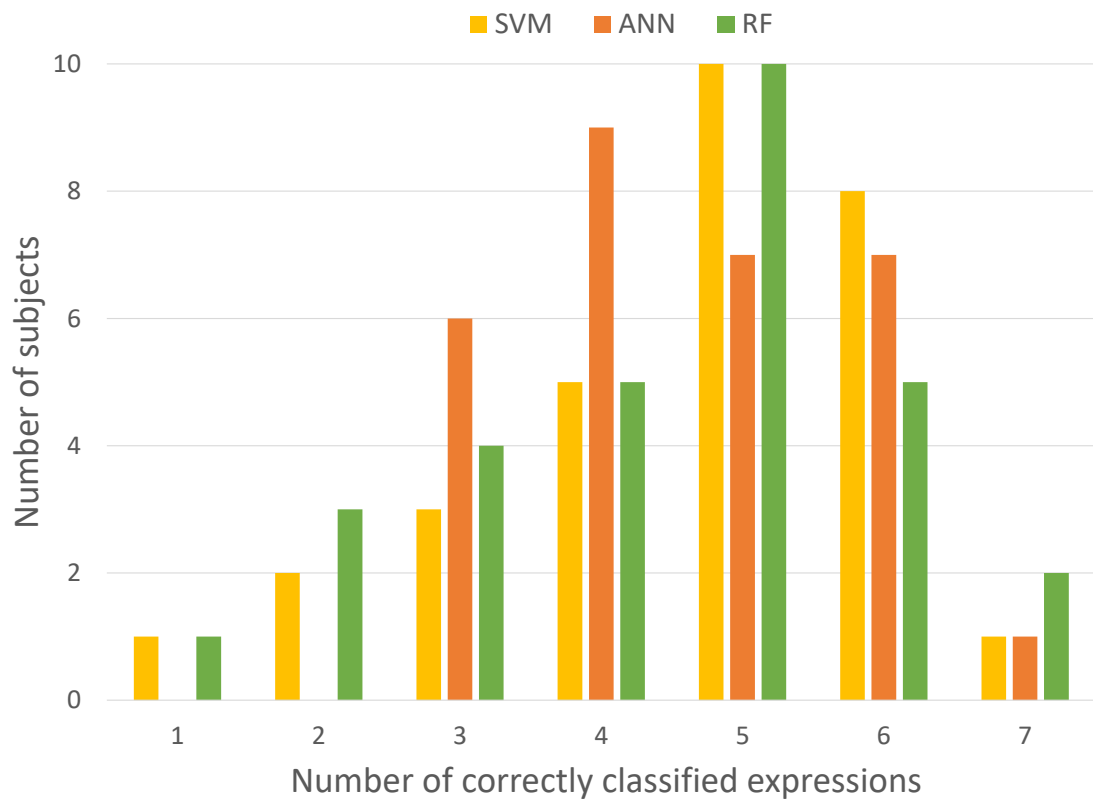


FIGURE 4.12: Test subject performance across machine learning techniques

consistent in regard to variation towards test subjects. Comparatively, the ANN is distributed more uniformly, with none of the subjects having a score of one or two out of seven. The majority of the scores for the ANN are distributed across three, four, five and six. Thus indicating that the spread of the ANN is much narrower than that of the SVM and RF. Therefore in this experiment the ANN has less variation and is more consistent across test subjects than the SVM and the RF.

#### 4.2.4 CK Test Set Results

The CK dataset was tested against the optimised models trained for each machine learning technique. It should be noted that the CK dataset does not contain the neutral class, but the models were trained to recognise this class. As mentioned previously using the CK dataset tests the generalising capability of the model and it must be remembered that this dataset has a non-constant number of images per class.

Table 4.15 summarises the number of correctly recognised frames for each machine learning technique on the CK dataset. Surprisingly the ANN increased in accuracy with the



TABLE 4.15: Performance of each machine learning technique on CK test set

Classifier	Total frames	Correct	Accuracy (%)
RF	307	180	58.63
ANN	307	214	69.71
SVM	307	198	64.50

highest accuracy of 69.71%. This is followed by the SVM with a high accuracy of 64.50% and the RF with 58.63%.

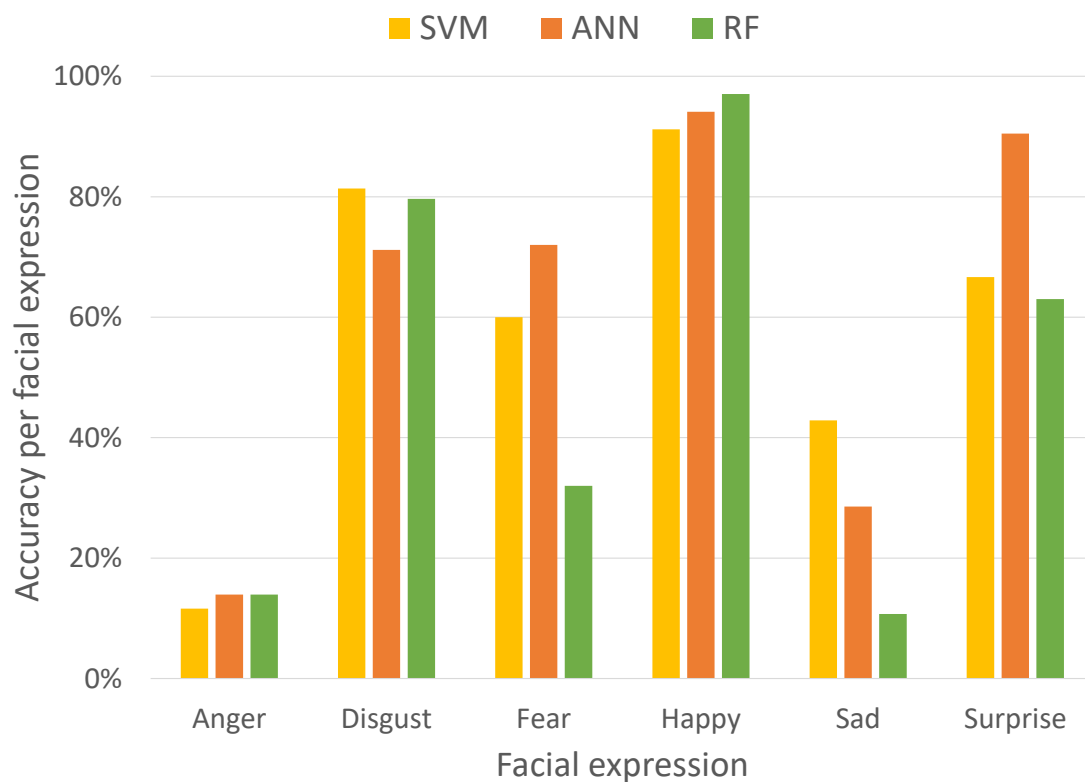


FIGURE 4.13: Comparison of facial expressions across machine learning techniques on the CK dataset

Figure 4.13 illustrates the per class analysis of the machine learning techniques on the CK dataset. Figure 4.13 illustrates that the RF classifier achieves accuracies below 30% for three of the six classes and achieves high accuracies of 80% for ‘Disgust’ and ‘Happy’. The performance of the RF on this dataset is acceptable, however it cannot compare with the ANN and SVM. The ANN performed exceptionally well, with the classes ‘Happy’ and ‘Surprise’ achieving accuracies of 90% and 94% respectively. ‘Disgust and ‘Fear’ also fared well, with accuracies above 70%. However, two of the classes score an accuracy of below 30%. Although the SVM and ANN are fairly comparable in terms of accuracy, the SVM only outperforms the ANN in two of the six classes whilst the ANN does so for

four classes. Thus it is clear that the ANN has excellent potential towards generalising on unseen data.

#### 4.2.5 Discussion and Choice of Machine Learning Technique

Several factors were considered when comparing the SVM, ANN, and RF which were: overall accuracy; accuracy across facial expressions; robustness towards test subjects; and ability to generalise. Table 4.16 summarises the results of the comparison of machine learning techniques.

TABLE 4.16: Summary of comparison of machine learning techniques

Dataset	Factor	SVM	ANN	RF
BU-3DFE	Overall Accuracy (%)	<b>66.20</b>	65.71	63.33
	Robust to Subjects	High	<b>Best</b>	High
	Robust to Classes	High	<b>Best</b>	High
CK	Overall Accuracy (%)	64.50	<b>69.71</b>	58.63
	Robust to Classes	High	<b>Best</b>	High
	Generalisation Capability	High	<b>Best</b>	High

The overall accuracies on the BU-3DFE test set revealed that the SVM, ANN and RF were comparable. Accuracies across facial expression classes were comparable as well, however the ANN proved to be slightly more consistent toward variation in classes. It is of note that similar trends in misclassification of classes were found. Cases exist whereby some images were misclassified identically across machine learning techniques. This confirms that some images, although labelled as one class, look more like another class. With regards to robustness towards test subjects, the ANN proved to be marginally better than the SVM and RF with all of the subjects getting scores of at least three and above out of seven. The SVM and the RF were comparable in terms of robustness towards test subjects.

The results on the CK test set revealed the generalisation capability of each machine learning technique. The ANN emerged as the outright winner followed by the SVM and lastly the RF. The RF performs considerably worse than the SVM and ANN. With regards to the spread of accuracies across classes the SVM and ANN are comparable, however the results favour the ANN.

These results indicate that the ANN tends to be a more robust machine learning technique as it outperforms the SVM and RF in terms of robustness towards test subjects. Furthermore, the SVM can generally be considered to be superior to the RF in terms of its ability to generalise.



## Chapter 5

# Conclusion

This research aimed to compare both feature extraction and machine learning techniques for facial expression recognition. A design science research artefact was built consisting of two phases. The first phase compared the feature descriptors: LBPs; CLBPs; and HOG using support vector machines. The second phase compared the machine learning techniques: SVMs; ANNs; and RFs using the best feature descriptor from phase one, namely, the HOG.

Four performance factors were considered, namely, classification accuracy, robustness towards variation in classes, robustness towards variation in subjects, and generalisation capability.

In response to the first research sub-question “How do the feature extraction methods local binary patterns, compound local binary patterns and histogram of oriented gradients compare in the context of facial expression recognition?”: It was concluded that in terms of classification accuracy, robustness towards variation in *classes* and robustness towards variation in *subjects*, the HOG descriptor proved to be marginally better than CLBP. CLBP outperforms LBP in the same cases. In terms of generalisation capability HOG performs the best, followed by LBP. Both HOG and LBP outperform CLBP which generalised poorly.

In response to the second research sub-question “How do the machine learning techniques support vector machines, artificial neural networks and random forests compare in the context of facial expression recognition?” It was concluded that in terms of classification accuracy the SVM was marginally better than the ANN. Both the ANN and SVM

outperforms the RF in this regard. However, the ANN is marginally better than the SVM in robustness towards variation in classes followed by the RF. With regard to robustness towards variation in subjects the ANN out-performs the SVM and is followed by the RF. In terms of generalisation capability the ANN once again outperforms the SVM which is followed by the RF.

Therefore in response to the main research question “Which feature extraction and machine learning techniques are best suited for facial expression recognition?”: It was concluded that the HOG feature descriptor and the ANN machine learning technique are best suited for facial expression recognition due to their consistency, overall accuracy, robustness toward changes in classes and subjects and generalisation capability.

## 5.1 Future Work

The HOG feature descriptor and the ANN machine learning technique have proven to be the best suited feature extraction method and machine learning technique in the context of facial expression recognition. In future, the HOG-ANN-based FER system should, therefore, be incorporated into the SASL gesture recognition system. Although the work has proved to be a good base for comparative research, it can be further extended to include: comparisons of each of the feature descriptors to each of the machine learning techniques; more comparative metrics such as ROC curves; more machine learning techniques for comparison such as convolutional neural networks; and more feature extraction for comparison such as other variations of LBPs.

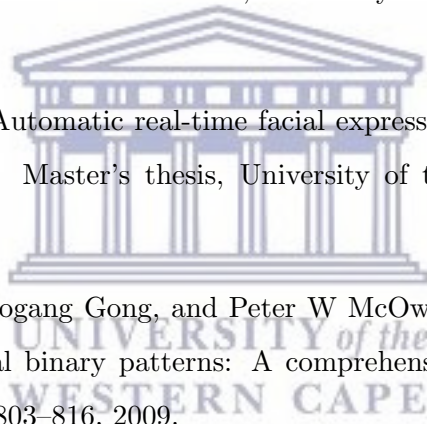
## 5.2 Concluding Remarks

The researcher has concluded that the course and processes followed have added great benefit to his life. It is hoped that this research can serve as a basis to compare machine learning techniques for other sign language parameters and for general classification problems.

# Bibliography

- [1] Zandile M Blose and Lavanithum N Joseph. The reality of every day communication for a deaf child using sign language in a developing country. *African Health Sciences*, 17(4):1149–1159, 2017.
- [2] Imraan Achmed. Upper body pose recognition and estimation towards the translation of South African Sign Language. Master’s thesis, University of the Western Cape, Computer Science, 2010.
- [3] Statistics South Africa. Profile of persons with disabilities in South Africa. In *Census 2011*. 2011.
- [4] Desmond van Wyk. Virtual human modelling and animation for sign language visualisation. Master’s thesis, University of the Western Cape, Computer Science, 2008.
- [5] Imraan Achmed. *Independent hand-tracking from a single two-dimensional view and its application to South African sign language recognition*. PhD thesis, University of the Western Cape, Computer Science, 2014.
- [6] William C Stokoe. Sign language structure: An outline of the visual communication systems of the american deaf. studies in linguistics: Occasional papers. Technical Report 8, Buffalo: Dept. of Anthropology and Linguistics, University of Buffalo, 1960.
- [7] Pei Li. Hand shape estimation for South African Sign Language. Master’s thesis, University of the Western Cape, Computer Science, 2010.
- [8] Roland G Foster. A comparison of machine learning techniques for hand shape recognition. Master’s thesis, University of the Western Cape, Computer Science, 2015.

- [9] Dane Brown. Upper body pose recognition and estimation towards the translation of South African Sign Language. Master's thesis, University of the Western Cape, Computer Science, 2013.
- [10] Imraan Achmed, Isabella M Venter, and Peter Eisert. A framework for independent hand tracking in unconstrained environments. In *Proc. of the Southern Africa Telecommunication Networks and Applications Conference 2012*, George, South Africa, 2012.
- [11] Ibraheem Frieslaar. Robust South African Sign Language gesture recognition using hand motion and shape. Master's thesis, University of the Western Cape, Computer Science, 2014.
- [12] Diego Mushfieldt. Robust facial expression recognition in the presence of rotation and partial occlusion. Master's thesis, University of the Western Cape, Computer Science, 2014.
- [13] Jacob Whitehill. Automatic real-time facial expression recognition for signed language translation. Master's thesis, University of the Western Cape, Computer Science, 2006.
- [14] Caifeng Shan, Shaogang Gong, and Peter W McOwan. Facial expression recognition based on local binary patterns: A comprehensive study. *Image and Vision Computing*, 27(6):803–816, 2009.
- [15] Rizwan Ahmed Khan, Alexandre Meyer, Hubert Konik, and Saida Bouakaz. Human vision inspired framework for facial expressions recognition. In *Image Processing (ICIP), 2012 19th IEEE International Conference on*, pages 2593–2596. IEEE, 2012.
- [16] Tommaso Gritti, Caifeng Shan, Vincent Jeanne, and Ralph Braspenning. Local features based facial expression recognition with face registration errors. In *8th IEEE International Conference on Automatic Face & Gesture Recognition, 2008. FG'08*, pages 1–8. IEEE, 2008.
- [17] Beat Fasel and Juergen Luetttin. Automatic facial expression analysis: a survey. *Pattern recognition*, 36(1):259–275, 2003.



- [18] Maja Pantic and Leon J. M. Rothkrantz. Automatic analysis of facial expressions: The state of the art. *IEEE Transactions on pattern analysis and machine intelligence*, 22(12):1424–1445, 2000.
- [19] Carl-Herman Hjortsjö. *Man's Face and Mimic Language*. Studen litteratur, 1969.
- [20] Paul Ekman and Erika L Rosenberg. *What the face reveals: Basic and applied studies of spontaneous expression using the Facial Action Coding System (FACS)*. Oxford University Press, USA, 1997.
- [21] Ekman Paul, Wallace V Friesen, and Joseph C Hager. Facial action coding system: The manual on CD ROM. *A Human Face, Salt Lake City*, 2002.
- [22] Paul Ekman. Facial expressions of emotion: an old controversy and new findings. *Phil. Trans. R. Soc. Lond. B*, 335(1273):63–69, 1992.
- [23] Takeo Kanade, Jeffrey F Cohn, and Yingli Tian. Comprehensive database for facial expression analysis. In *Proc. Fourth IEEE International Conference on Automatic Face and Gesture Recognition*, pages 46–53. IEEE, 2000.
- [24] Lijun Yin, Xiaozhou Wei, Yi Sun, Jun Wang, and Matthew J Rosato. A 3D facial expression database for facial behavior research. In *7th International Conference on Automatic Face and Gesture Recognition*, pages 211–216. IEEE, 2006.
- [25] Matti Pietikäinen, Guoying Zhao, Abdenour Hadid, and Timo Ahonen. *Computer Vision Using Local Binary Patterns*. Number 40 in Computational Imaging and Vision. Springer, 2011.
- [26] Timo Ojala, Matti Pietikäinen, and David Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition*, 29(1):51–59, 1996.
- [27] Xiaoyi Feng, M Pietikäinen, and Abdenour Hadid. Facial expression recognition with local binary patterns and linear programming. *Pattern Recognition and Image Analysis*, 15(2):546–548, 2005.
- [28] Ross Beveridge, David Bolme, Marcio Teixeira, and Bruce Draper. The CSU face identification evaluation system users guide: version 5.0. *Computer Science Department, Colorado State University*, 2(3):1–29, 2003.



- [29] Michael Lyons, Shigeru Akamatsu, Miyuki Kamachi, and Jiro Gyoba. Coding facial expressions with Gabor wavelets. In *Proc. Third IEEE International Conference on Automatic Face and Gesture Recognition*, pages 200–205, 1998.
- [30] Paul A Viola and Michael J Jones. Robust real-time object detection. *International Journal of Computer Vision*, 57(2):137–154, 2004.
- [31] Jalal Aldin Nasiri, Sara Khanchi, and Hamid Reza Pourreza. Eye detection algorithm on facial color images. In *Modeling & Simulation, 2008. AICMS 08. Second Asia International Conference on*, pages 344–349. IEEE, 2008.
- [32] Timo Ahonen, Abdenour Hadid, and Matti Pietikäinen. Face recognition with local binary patterns. In *Computer Vision-ECCV 2004*, pages 469–481. Springer, 2004.
- [33] Timo Ojala, Matti Pietikainen, and Topi Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, 2002.
- [34] Stefan Knerr, Léon Personnaz, and Gérard Dreyfus. Single-layer learning revisited: a stepwise procedure for building and training a neural network. *Neurocomputing: Algorithms, Architectures and Applications*, 68(41-50):71, 1990.
- [35] Xiaoyang Tan and Bill Triggs. Enhanced local texture feature sets for face recognition under difficult lighting conditions. In *International Workshop on Analysis and Modeling of Faces and Gestures*, pages 168–182. Springer, 2007.
- [36] Xiaosheng Wu, Junding Sun, Guoliang Fan, and Zhiheng Wang. Improved local ternary patterns for automatic target recognition in infrared imagery. *Sensors*, 15(3):6399–6418, 2015.
- [37] Faisal Ahmed, Hossain Bari, and Emam Hossain. Person-independent facial expression recognition based on compound local binary pattern (CLBP). *Int. Arab J. Inf. Technol.*, 11(2):195–203, 2014.
- [38] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893. IEEE, 2005.

- [39] Ying-li Tian. Evaluation of face resolution for expression analysis. In *Computer Vision and Pattern Recognition Workshop, 2004. CVPRW'04. Conference on*, pages 82–82. IEEE, 2004.
- [40] Judith MS Prewitt. Object enhancement and extraction. *Picture Processing and Psychopictorics*, 10(1):15–19, 1970.
- [41] Junkai Chen, Zenghai Chen, Zheru Chi, and Hong Fu. Facial expression recognition based on facial components detection and HOG features. In *International Workshops on Electrical and Computer Engineering Subfields*, pages 884–888, 2014.
- [42] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.
- [43] Caifeng Shan, Shaogang Gong, and Peter W McOwan. Robust facial expression recognition using local binary patterns. In *IEEE International Conference on Image Processing 2005*, volume 2, pages II–370. IEEE, 2005.
- [44] Sunanda P Khandait, Ravindra C Thool, and Prabhakar D Khandait. Automatic facial feature extraction and expression recognition based on neural network. *International Journal of Advanced Computer Science and Applications*, 2(1):113–118, 2011.
- [45] Mauricio Hess and Geovanni Martinez. Facial feature extraction based on the smallest univalue segment assimilating nucleus (susan) algorithm. In *Proceedings of Picture Coding Symposium*, volume 1, pages 261–266, 2004.
- [46] Javier G Rázuri, David Sundgren, Rahim Rahmani, and Antonio Moran Cardenas. Automatic emotion recognition through facial expression analysis in merged images based on an artificial neural network. In *Proc. 12th Mexican International Conference on Artificial Intelligence*, pages 85–96. IEEE, 2013.
- [47] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.
- [48] Arnaud Dapogny, Kevin Bailly, and Séverine Dubuisson. Pairwise conditional random forests for facial expression recognition. In *Proc. of the IEEE International Conference on Computer Vision*, pages 3783–3791, 2015.
- [49] Naval Bajpai. *Business Research Methods*. Pearson Education India, 2011.

- [50] Michael Crotty. *The Foundations of Social Research: Meaning and Perspective in the Research Process*. Sage, 1998.
- [51] Salvatore T March and Gerald F Smith. Design and natural science research on information technology. *Decision Support Systems*, 15(4):251–266, 1995.
- [52] R Hevner Von Alan, Salvatore T March, Jinsoo Park, and Sudha Ram. Design science in information systems research. *MIS Quarterly*, 28(1):75–105, 2004.
- [53] Alan Hevner and Samir Chatterjee. *Design Research in Information Systems: Theory and Practice*, volume 22. Springer Science & Business Media, 2010.
- [54] Ken Peffers, Tuure Tuunanen, Charles E Gengler, Matti Rossi, Wendy Hui, Ville Virtanen, and Johanna Bragge. The design science research process: a model for producing and presenting information systems research. In *Proc. of the First International Conference on Design Science Research in Information Systems and Technology (DESRIST 2006)*, pages 83–106. ME Sharpe, Inc., 2006.
- [55] Matti Rossi and Maung K Sein. *Design Research Workshop: A Proactive Research Approach*. IRIS Association, Haikko, 2003.
- [56] Hideaki Takeda, Paul Veerkamp, and Hiroyuki Yoshikawa. Modeling design processes. *AI Magazine*, 11(4):37, 1990.
- [57] Waleed Deaney, Isabella Venter, Mehrdad Ghaziasgar, and Reg Dodds. A comparison of facial feature representation methods for automatic facial expression recognition. In *Proc. of the South African Institute of Computer Scientists and Information Technologists, SAICSIT '17*, pages 10:1–10, Thaba 'Nchu, South Africa, 2017.
- [58] David G Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.
- [59] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, pages 144–152. ACM, 1992.
- [60] Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin. A practical guide to support vector classification. Technical report, National Taiwan University, 2003.

- [61] Ulrich H-G Kreßel. Pairwise classification and support vector machines. In *Advances in Kernel Methods*, pages 255–268. MIT press, 1999.
- [62] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [63] Kishan Mehrotra, Chilukuri K Mohan, and Sanjay Ranka. *Elements of Artificial Neural Networks*. MIT Press, 1997.
- [64] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer, 2009.

