University of Exeter

Department of Computer Science

Emotion Classification Using Combinations of Texture Descriptors

Huthaifa Ziad Abuhammad

November, 2018

Supervised by Professor Richard Everson & Dr Jacqueline Christmas

Submitted by Huthaifa Ziad Abuhammad to the University of Exeter as a thesis for the degree of Doctor of Philosophy in Computer Science

I certify that all material in this thesis which is not my own work has been identified and that no material has previously been submitted and approved for the award of a degree by this or any other University.

(signature) .....

Abstract

We present an automated new approach for facial expression recognition of seven emotions. The main objective of this thesis is building a model that can classify the spontaneous facial expressions, rather than the acted ones, and apply this model images and videos. Moreover, we will investigate if a combination of more than one image feature descriptor will improve the classification rate, and the efficacy of the texture descriptors on videos sequences.

Three types of texture features from static images were combined: Local Binary Patterns (LBP), Histogram of Oriented Gradients (HOG) and Dense Speeded Up Robust Features (D-SURF). The resulting features are classified using random forests. The use of random forests allows for the identification of the most important feature types and facial locations for emotion classification. Regions around the eyes, forehead, sides of the nose and mouth are found to be most significant. We classified the important features with random forest and Support Vector Machines. We also found that the classification performance became better than using all of the extracted facial features. We achieved better than state-of-the-art accuracies using multiple texture feature descriptors.

Current emotion recognition datasets comprise posed portraits of actors displaying emotions. To evaluate the recognition algorithms on spontaneous facial expressions, we introduced an unposed dataset called the "Emotional Faces in the Wild" (eLFW), a citizen-labelling of 1310 faces from the Labelled Faces in the Wild data. To collect this data, we built a website and asked citizens to label photos according to the emotion displayed. The citizens were also asked to label a selection of

KDEF faces. We evaluated the common misclassification of the faces, similar to what people do; machine algorithms perform worst regarding distinguishing between sad, angry and fearful expressions. We describe a new weighted voting algorithm for multi-calcification, in which the predictions of the classifiers trained on pairs of classes are combined with weights learned using an evolutionary algorithm. This method yields superior results, particularly for the hard-to-distinguish emotions. The method was applied to the DynEmo video database. We investigated some methods to smooth the classifier predictions in order to exploit temporal continuity emotions and therefore classification error. Several smoothing techniques were investigated and optimised, and we found that the simple moving average and linear fit Lowess smoothing performed best.

Acknowledgements

First and foremost, I express my heartfelt gratitude to my supervisor Professor Richard Everson for the support, encouragement and patient guidance that he provided throughout all stages. In particular, I am thankful to Dr Jacqueline Christmas and Dr Jonathan Fieldsend and all of the Computer Science Department for being an incredible people who always motivated and inspired my work in times of hardship, as Newton says, "If I have seen further, it is by standing on the shoulders of giants". I also would like to thank them, and all the people who helped us in the data collection conducted in chapter 3.

My gratitude goes to my family, whose help has been invaluable. If it were not for them, this project would not have been written. First, I would like to thank **my father, Ziad Abuhammad**, who has always been a source of encouragement and inspiration to me throughout my life. Without his sacrifices and generous full financial support of my research, my project could not have been completed. I know that whatever I say, it will be impossible to reciprocate. Secondly, I would thank **my mother, Fatema Alsied**, for all the confidence she displayed in me. Her continual support both emotionally and mentally with her prayers gave me the strength to pursue my project. Thirdly, I would like to thank **my wife, Rawaah Alzyod**, for her inspiring and enthusiastic advice during difficult moments. Her understanding played an essential role in enabling me to accomplish this work.

My thanks also go **to my children**, who have given me much happiness and keep me hoping for the future. Each one is deeply loved in a unique way. There is my beloved little man and my best friend, **Abdullah**; who I see myself through

his lovely eyes. To my sweet **daughter Lujain**, and my newborn baby, **Balqees**, I would like to express my thanks for being such a good girls always cheering me up. My son, **Khaled**, the little boy, who always try to do everything to make his presence felt. I hope I have been a good father and that I have not lost too much during the tenure of my study.

To my Grandmother soul, Helalah Abuhammad, who left me too early during my PhD, you waited to see me graduated, but die was faster. I hope that this work makes her soul proud, I will not forget you, you still alive in my heart.

To my grandfather, Awwad Abuhammad, I thank you for your continuing long-distance prayers and moral support.

I would like to thank my mother in law Nihad Alsied and my brother in law Zaid Alzyod for their continuous encouragement. I would like to mention as well my extended family, all my uncles, my aunts and all my cousins.

I also would like to thank my friend in Jordan Khaled Aljaber keeping in touch whole during my studying. I would like to thank my dear friends Rami Chehab and his wife Aya Al-Ghalayini, Yazeed Sammour and Omer Alsaif for their company and quality time. I would like to thank Mohammad Younis and his wife Luma Alsafar who had always encouraged me up at tough times especially during the lonely stages of my PhD.

A very special thanks go to my friends who I knew them in Exeter: Sohaib Al-Ramadhani and his wife Marwah Mohammed, Dr Hafez Alawad, Ali Nassr and his wife Sarah Mohammed, Hajji Shikh Mohammed and his wife Walaa Hemesho, Ali Alabdali Qusai Almudares, Saad Alkhalifa, Hussain Elahmad and Jalil Kwad.

Contents

1 Introduction **1**

1.1 Motivation 1

1.2 Challenges 3

1.3 Objectives and aims 5

1.4 Contributions 5

1.5 Publications 6

1.6 Structure of the thesis 7

2 Background **10**

2.1 Introduction 11

2.2 Review of the "state of the art" for facial expressions recognition methods 13

2.2.1 Active shape and active appearance models 13

2.2.2 Image texture-based methods 17

2.2.3 Classification 25

2.2.4 Support Vector Machine (SVMs) 27

2.2.5 Random forest 30
2.2.6 Deep learning and Convolutional Neural Networks 34
2.3 Facial expression datasets 35
2.4 Conclusion 38

3 Emotional faces in the wild database 40
3.1 Introduction 40
3.2 Data collection 41
3.3 Summary of citizens' classification 46
3.4 Conclusion 49

4 Combining texture features for emotion classification 50
4.1 Introduction 51
4.2 Image preprocessing 53
4.2.1 Face detection and alignment 54
4.2.2 Convert RGB image to greyscale. 55
4.3 Texture-feature extraction and combination 56
4.4 Baseline random forest classification 58
4.4.1 KDEF experiments 58
4.4.2 CK+ experiments 60
4.4.3 eLFW experiments 64
4.5 Feature importance mask 68
4.6 Random forest classification with importance mask 70
4.6.1 Masked KDEF experiments 70

4.6.2 Masked CK+ experiments . . . 71
4.6.3 Masked eLFW experiments . . . 72
4.7 Comparison with citizens' classification . . . 73
4.8 Support vector machine performance . . . 74
4.9 Pairwise classification . . . 77
4.9.1 Equally weighted pairwise classification . . . 78
4.9.2 Weighted pairwise classification . . . 83
4.10 Conclusion . . . 87

5 Facial expression recognition in video 89
5.1 Introduction . . . 90
5.2 DynEmo database preparation . . . 91
5.3 Video Classification Experiments . . . 94
5.4 Smoothing . . . 98
5.4.1 Smoothing techniques overview . . . 98
5.4.2 Smoothing optimisation . . . 100
5.5 Conclusion . . . 104

6 Conclusion and perspectives 106
6.1 Future Work . . . 108

List of Figures

2.1 Basic facial expression recognition system pipeline 12

2.2 Point Distribution Model (T.F. Cootes and Taylor, 1999) 14

2.3 HOG descriptor Visualization 17

2.4 An overview of the face recognition system with HOG (Chen et al., 2014) 20

2.5 An example of images with extracted SIFT features. The images represent the same object with different expressions and illumination (Zhang et al., 2008) 22

2.6 Simple decision tree 31

2.7 New subsets Randomisation by bagging method 32

2.8 The general pipeline of deep facial expression recognition systems (Li and Deng, 2018). 34

3.1 Examples of citizen labelling for LFW. 42

3.2 Screenshots two versions of the emotional faces website 43

3.3 Faces samples from KDEF database which posed expressions 45

3.4 Examples of faces for which citizens' votes differ from the KDEF labelling. KDEF labels are shown above each image with the citizens' consensus below. 48

4.1 Four points detection(the two eyes centres and two point on the mouth). 54

4.2 Face detection with the Viola Jones algorithm. 55

4.3 An example of facial alignment for a LFW image using the Huang et al. (2012) method 56

4.4 Block features (HOG, LBP and SURF) extraction and combination . 57

4.5 The out-of-bag error decreases with the number of grown trees for the combined features with KDEF images. 62

4.6 Schematic of steps used for determining the feature importance mask. This figure shows the steps of finding the importance mask by applying the Otsu method to the total of the estimated importance values. . . 67

4.7 Importance masking steps for a KDEF image. 68

4.8 HOG, LBP and SURF importance values comparison. Red, blue and green indicate which feature type (HOG, LBP and D-SURF respectively) was most important for each block. We use half of the face to make the mask symmetric. 69

4.9 Mask identifying the most important regions of the face for emotion classification derived by binariesing feature importance mask. 70

4.10 Min objective vs number of function evaluations 77

4.12 The out-of-bag error decreases with the number of grown trees fear vs surprise. 80

4.11 The out-of-bag error decreases with the number of grown trees for happy vs surprise. 81

5.1 Examples of the five facial expression in the database 93

5.2 Happy expression variation for the same person. 94

5.3 Classifier prediction behaviour for 2 happy-labelled videos 97

5.4 The overall accuracy vs. smoothing span, (25 frames per second.) . . 101

5.5 Smoothing result for video DVD79_5 with the optimal span. The top plot shows the scores before smoothing whereas the middle shows the scores after smoothing. The bottom plot illustrates the overall accuracy for all frames before a particular frame. 102

List of Tables

2.1 Databases summary 37

3.1 Voting data sample 46

3.2 Summary of statistics of eLFW database. 47

3.3 Confusion matrices for the citizens' performance on KDEF images. True classes are shown by rows, with assigned classes in columns. FE: fear; AN: anger; DI: disgust; HA: happiness; NE: neutral; SA: sadness; SU: surprise. 48

4.1 Confusion matrix for KDEF images with HOG features and a random forest classifier. The overall accuracy is 73.10%, precision is: 77.90%, recall is: 73.10% and the F1-score is: 71.40%. 59

4.2 Confusion matrix for the KDEF images with LBP features and random forest classifier. The overall accuracy is 80.10%, precision is: 85.00%, recall is: 80.10% and the F1-score is: 79.10%. 60

4.4 Confusion matrix for the KDEF images with the combined features and random forest classifier. The overall accuracy is: 82.20%, precision is: 83.40%, recall is: 82.20% and the F1-score is: 82.30%. 60

4.3 Confusion matrix for the KDEF photos with D-SURF features and random forest classifier. The overall accuracy is 70.50%, precision is: 75.30%, recall is: 70.50% and the F1-score is: 67.70%. 61

4.6 Confusion matrix for the CK+ images with LBP features and random forest classifier. The overall accuracy is: 71.60%, precision is: 66.40%, recall is: 68.10% and the F1-score is: 65.70%. 62

4.5 Confusion matrix for the CK+ images with HOG features and random forest classifier. The overall accuracy is: 67.70%, precision is: 61.70%, recall is: 63.30% and the F1-score is: 60.90%. 63

4.7 Confusion matrix for the CK+ images with D-SURF features and random forest classifier. The overall accuracy is: 60.50%, precision is: 54.70%, recall is: 56.80% and the F1-score is: 54.10%. 63

4.8 Confusion matrix for the CK+ images with the combined features and random forest classifier. The overall accuracy is: 78.30%, precision is: 74.80%, recall is: 75.80% and the F1-score is: 73.90%. 64

4.9 Confusion matrix for the eLFW images with HOG features and a random forest classifier. The overall accuracy is: 56.90%, precision is: 52.20%, recall is: 56.30% and the F1-score is: 52.60%. 65

4.10 Confusion matrix for the eLFW images with LBP features and random forest classifier. The overall accuracy is: 61.00%, precision is: 55.90%, recall is: 59.20% and the F1-score is: 56.10%. 65
4.11 Confusion matrix for the eLFW images with D-SURF features and random forest classifier. The overall accuracy is: 51.20%, precision is: 46.70%, recall is: 48.80% and the F1-score is: 46.50%. 66
4.12 Confusion matrix for the eLFW images with the combined features and random forest classifier. The overall accuracy is: 67.30%, precision is: 59.80%, recall is: 61.40% and the F1-score is: 59.60%. 66
4.13 Confusion matrix for the masked the KDEF images with the combined features and random forest classifier. The overall accuracy is: 89.80%, precision is: 90.80%, recall is: 89.80% and the F1-score is: 89.70%. . . 71
4.14 Confusion matrix for the masked CK+ images with the combined features and random forest classifier. The overall accuracy is: 82.20%, precision is: 79.10%, recall is: 81.30% and the F1-score is: 79.20%. . . 72
4.15 Confusion matrix for the random forest classification of the masked eLFW databases. The overall accuracy was 71.60%, precision is: 64.10%, recall is: 66.00% and the F1-score is: 64.00%. 73
4.16 Confusion matrix on the eLFW database using proposed method, trained with KDEF. Overall accuracy was 74.7%, precision is: 74.29%, recall is: 74.7% and the F1-score is: 74.01%. 74
4.17 Average entropy of citizens' voting distributions for correctly and incorrectly classified the eLFW images. 75

4.18 caption . . . 76
4.19 Confusion matrix for the SVM classification of the eLFW database with the importance mask shown in Figure 4.7. Accuracy 66.3%, precision is: 61.30%, recall is: 63.3% and the F1-score is: 60.60%. . . 78
4.20 caption . . . 79
4.21 Pairwise RF classifiers and pairwise SVM classifiers performance with KDEF. . . . 80
4.23 Confusion matrices for equally pairwise classification of the CK+ database. The overall accuracy is 89.40%, precision is: 88.20%, recall is: 88.80% and the F1-score is: 88.20%. . . . 81
4.22 caption . . . 82
4.24 Random forest pair-classifiers optimised weights for 7 classes . . . 83
4.25 caption . . . 84
4.26 Confusion matrices for weighted pairwise classification of the CK+ database. The overall accuracy is 91.30%, precision is: 90.70%, recall is: 89.40% and the F1-score is: 89.60%. . . . 85
4.27 Comparison of classification accuracy of random forest classification using masked LBP, HOG and D-SURF texture features with other recent techniques. Evaluation on the KDEF and CK+ databases. . . 86
4.28 The improvement progress summary for the proposed method. . . . 88

5.1 DynEmo database data type. . . . 91
5.2 General statistics about the new database . . . 94
5.3 caption . . . 95

5.4 caption 96
5.5 Five-Class random forest pair-classifiers with their optimised weights 97
5.6 Optimising of the smoothing span by Nelder–Mead (Staring point (0.5)) 100
5.7 caption 103
5.8 caption 103
5.9 Comparison of the classification accuracy of smoothed random forest pairwise classification using the masked LBP, HOG and D-SURF texture features with other recent techniques. Evaluation on the CK+ databases (for dynamic-based deep facial expression recognition). . . 104

Chapter 1

Introduction

Contents

1.1 Motivation 1

1.2 Challenges 3

1.3 Objectives and aims 5

1.4 Contributions 5

1.5 Publications 6

1.6 Structure of the thesis 7

1.1 Motivation

Computers have become an important part of our lives. They have moved from being just equipment for managing business and office tasks to being a real partner in social communication and interaction with humans. The prevalence in the past

two decades of small computers such as laptops, mobile phones and tablet devices has played a significant role in making computers a permanent companion of our lives and within our relationships with others.

Our emotions critically affect all aspects of our lives, from how we live, work, learn and play, to our decisions, big and small. Facial expressions have a significant role in our communication with others. Today, mobile phones and computers are a significant way to communicate with others, so machines have to perform lots of human tasks, and they need to have more human-like capabilities. Human emotional intelligence depends on our ability to recognise not only our own emotions but also those of other people. To this end, smart devices and advanced AI (artificial intelligence) systems should have the capacity to understand our emotions and to interact with humans emotionally. Human-computer intelligent interaction (HCII) is a growing field that aims to achieve that. Human faces are the most important part of the human body used to express feelings. People around the world use similar facial expressions to express emotions such as happiness, sadness, anger, disgust, surprise and fear. Facial expressions enable people to understand each other; sometimes without even a single word being spoken. In other words, facial expressions are a global language.

Most proposed automatic facial-expression recognition methods have been based on posed facial expressions, using databases that have been built from acted facial expressions. The creators of those databases have asked models or actors to express facial expressions and the problem here that the spontaneous emotions are different from those posed. This limits the ability of emotion classification systems trained on

these databases to generalise to naturally expressed emotions.

In recent times, facial emotional recognition systems have been used in several applications such as:

1. Detection and treatment of depression and anxiety (Ekman and Rosenberg, 1997).
2. The FaceReader (Den Uyl and Van Kuilenburg, 2005).
3. EmotiChat (Anderson and McOwan, 2006) .
4. Smart homes (Pantic et al., 2007).
5. Affective/social robots (Scherer et al., 2010).
6. Emotientv (Whitehill et al., 2013).
7. EmoVu (Arnold and Emerick, 2016).

For more extensive review see (Khan, 2013) and the "20+ Emotion Recognition APIs" website which keeps an updated compendium (Bill Doerrfeld, 2016).

From the consideration of the above, we need more to study facial expression recognition with spontaneous expressions to be more useful in all real-life applications because of the variation of facial expressions between people.

1.2 Challenges

Although many methods have been proposed for facial expression recognition, many challenges and difficulties are still faced in this field, especially with natural expres-

sions. This thesis particularly addresses the following challenges:

1. The similarity of facial expressions means that many people find it difficult to differentiate between certain expressions. For example, fear and surprise. We performed a study by asking people to identify a range of facial expressions and found a significant variation in people's ability to determine facial expressions particularly those representing emotions such as anger, fear, and surprise. Machines also face similar difficulty when recognising such expressions.

2. It is well known that there is a variation amongst human faces and thus their way of expressing their emotions. Some people exaggerate their expressions, while others try to hide them. Moreover, there are many basic facial expressions (fear, anger, disgust, happy, neutral, sad and surprise) and some secondary expressions. For example, pain, and this may change according to cultures and nation. This variation may divide the same expression into levels that would make it harder for machines to classify them. For example, a happy expression could include a small smile or a deep laugh.

3. This study was keen to use spontaneous facial expressions rather than posed ones. Most of the facial expression datasets contain posed facial expressions and have been built depending on acted facial expressions by models or actors. Consequently, these databases are not indicative of the way that people express their emotion in reality. Therefore, machines trained on these databases may not be able to classify spontaneous expressions accurately.

1.3 Objectives and aims

- The main objective of this thesis is building a model that can classify spontaneous facial expressions, rather than acted ones, and to apply this model to images and videos.
- We aim to investigate how texture descriptors effect facial emotion classification.
- We will investigate if a combination of more than one image features, the descriptor will improve the classification rate.
- We aim to investigate the efficacy of the texture descriptors on video sequences.

1.4 Contributions

The contributions in this field, as described in this thesis are as follows:

- We used the Labelled Faces in the Wild dataset to construct a new facial expression dataset emotional Labelled Faces in the Wild (eLFW) by using citizens to derive consensus labels for the emotions. The new dataset contains natural facial expressions in contrast to most of the current datasets, which depend on actors mimicking facial expressions. In total, 135 persons have voted the LFW, so the new eLFW dataset contains 1310 labelled images.
- We showed that a combination of three image feature extraction methods, Local Binary Patterns (LBP) (Ojala et al., 1996), the Histogram of Oriented

Gradients (HOG) (Dalal and Triggs, 2005), and Dense Speeded-up Robust Features (D-SURF) (Bay et al., 2006, 2008) provide a better image description than using only one of them. This improves the classification rate using SVM and random forests. Random forests give only 56.90% with HOG, 61.00% with LBP and 51.20% D-SURF separately when tested on the eLFW. For the combination of the three descriptors together, the accuracy jumps to 67.30%

- We propose a new method to identify the most relevant image regions for emotion classification. Applying the mask to the eLFW increases the random forest classification accuracy from 67.30% without masking to 71.60% with masking.

- We describe a new weighted voting algorithm, in which the weighted predictions of classifiers trained on pairs of classes are combined with the weights learned using an evolutionary algorithm. This method yields superior results, particularly for the hard-to-distinguish emotions. This algorithm results in 73.3% accuracy for the equal pair-wise classification and 76.60% for the weighted pair-wise classification when tested on eLFW.

- We used smoothing techniques to reduce video classification errors (noise), by relying on a set of sequential video frames.

1.5 Publications

The material presented in chapters 3 and 4 has been published in:

Abuhammad, H., & Everson, R. (2018, June). Emotional Faces in the Wild: Feature Descriptors for Emotion Classification. In *International Conference on Image Analysis and Recognition*, Pages 164-174. Springer.

1.6 Structure of the thesis

The rest of the thesis is structured as follows.

Chapter 2

We present a brief overview of some of the related feature extraction methods, together with the random forest and support vector machine classifiers. We present a brief overview of three optimisation methods: the Nelder-Mead simplex direct search, Bayesian optimisation and the Covariance Matrix Adaptation Evolution Strategy. In particular, we discuss some related methods of facial expression which depend on image-appearance techniques.

Chapter 3

In this chapter, we introduce the new "Emotional Labelled Faces in the Wild" dataset (eLFW), a citizen-labelling of 1310 faces from the Labelled Faces in the Wild data LFW (Huang et al., 2007a; Learned-Miller, 2014). To achieve that, we built a website to allow people to vote the facial expressions. The new data set enabled us to evaluate the proposed texture-based emotions classification on realistic data.

Chapter 4

In this chapter, we present an automated new approach for facial expression recognition of seven emotions. Three types of texture features (LBP, HOG and D-SURF) from static images are combined, and the resulting features are classified using random forests.

We achieve comparable accuracies with recent works using multiple texture feature descriptors. The use of random forests allows for the identification of the most important feature types and locations used for emotion classification. Regions around the eyes, forehead, sides of the nose and mouth were found to be the most significant.

We found a similarity between the machine and people's classification of the eLFW and the Karolinska Directed Emotional Faces (KDEF) Lundqvist et al. (1998) data obtained from actors, and poorest results are obtained in distinguishing the sad, angry and fearful emotions.

We describe a new weighted voting algorithm, in which the weighted predictions of classifiers trained on pairs of classes are combined with the weights learned using an evolutionary algorithm. This method yields superior classification accuracy, particularly for the hard-to-distinguish emotions.

Chapter 5

In this chapter, we apply the proposed method in this chapter to DynEmo video databases. We describe how smoothing positively affects the classifier scores by reducing errors.

Chapter 6

This chapter reviews a summary of the proposed algorithms and the associated results presented in this thesis, in addition directions for future work.

Chapter 2

Background

Contents

2.1 Introduction 11

2.2 Review of the "state of the art" for facial expressions recognition methods 13

2.2.1 Active shape and active appearance models 13

2.2.2 Image texture-based methods 17

2.2.3 Classification 25

2.2.4 Support Vector Machine (SVMs) 27

2.2.5 Random forest 30

2.2.6 Deep learning and Convolutional Neural Networks 34

2.3 Facial expression datasets 35

2.4 Conclusion 38

2.1 Introduction

In this chapter, we present the background necessary to appreciate our proposed method based on image texture and random forests. We present a generic overview of various main concepts, with the more specific details of different techniques in the relevant chapters.

The facial expression recognition field first started being discussed at the end of the last century (Suwa, 1978; Essa and Pentland, 1995). Since then, many researchers have proposed methods to improve the ability of machines to recognise human facial expressions.

Traditional pipeline facial expression recognition systems follow the same general steps, as illustrated in figure 2.1. The first is to find the Region Of Interest (ROI), which is the face. Unwanted areas may badly affect classification accuracy. Many facial detection methods have been proposed, such as the Viola-Jones algorithm (Viola and Jones, 2004, 2001). The next step is extracting the features from the detected face. The purpose of image feature extraction is to describe an image efficiently by extracting the essential values and reducing the data without losing any significant details. There are many ways to do that, with some depending on image texture itself, while other methods describe the image geometrically. The last step is to classify the extracted features. In this step, machine learning plays the primary role in finding the differences between the feature groups and making a decision as to what a group of values refers to.

Deep learning algorithms have been used in a wide range of fields, including

Figure 2.1: Basic facial expression recognition system pipeline

automatic speech recognition, image recognition, natural language processing, drug discovery, and facial expressions. One of the essential advantages of deep learning is that we do not need to extract the features from the image manually. Deep learning can learn to extract the features while training using its convolution kernels. The main disadvantages of deep learning are that it needs a large amount of data and a large amount of computational power. With the big revolution in the speed of computers and the emergence of Big Data, deep learning has attracted considerable attention by researchers in recent years.

Some facial expression recognition methods may contain further steps to improve accuracy. For example, applying pre-processing techniques to enhance the image before feature extraction, or by reducing the length of the extracted feature vector.

2.2 Review of the "state of the art" for facial expressions recognition methods

2.2.1 Active shape and active appearance models

An active appearance model (AAM) (Edwards et al., 1998; Cootes et al., 1998, 2001) is a computer vision algorithm which depends on statistically finding the values which fit with a grey image's texture values, and making a statistical link with an active shape model (Cootes and Taylor, 1992; Cootes et al., 1992, 1995). ASM was first proposed by Cootes and Taylor (1992) based on models created from sets of training examples called Point Distribution Models (PDM). These represent objects as sets of labelled points called landmark points. Figure 2.2 illustrates a PDM as an example of face points landmarking. AAM was first proposed by Edwards et al. (1998) for face analysis. Since then the method has been commonly used in computer vision applications such as face matching, tracking faces, medical image analysis and emotion recognition (Ratliff and Patterson, 2008; Ko and Sim, 2010; Setyati et al., 2012; Lozano-Monasor et al., 2014; Chen et al., 2013; Yu et al., 2013).

The training set is normally labelled manually. Each shape is represented as shown in equation 2.1.

$$\boldsymbol{x} = (x_0, y_0, x_1, y_1, ..., x_k, y_k, ..., x_{n-1}, y_{n-1})^T \tag{2.1}$$

where (x_k, y_k) is the position point k

The first step is computing the mean points for all the training set to find the mean shape $\bar{\boldsymbol{x}}$. AAM's idea is to combine a model of face shape variation with a model of

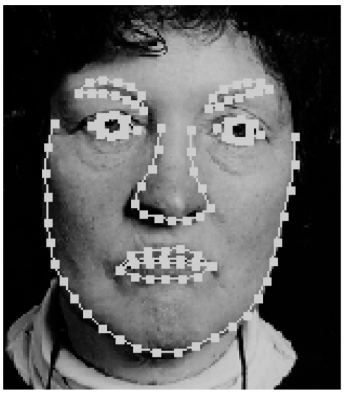


Figure 2.2: Point Distribution Model (T.F. Cootes and Taylor, 1999)

the appearance variations of a shape-normalised face. To create new shapes from the mean shape $\bar{\boldsymbol{x}}$, ASM generates new shapes and textures using Principal Component Analysis (PCA). PCA is applied to all training data to calculate the eigenvectors of the covariance matrix. The following equation is used to generate a new shape:

$$\boldsymbol{x} \approx \bar{\boldsymbol{x}} + \boldsymbol{P}_s \boldsymbol{b}_s \tag{2.2}$$

where $\bar{\boldsymbol{x}}$ the mean shape, $\boldsymbol{P} = (\boldsymbol{P}_1 \mid \boldsymbol{P}_2 \mid ... \mid \boldsymbol{P}_t)$ contains t eigenvectors of the covariance matrix and $\boldsymbol{b}$ is a t dimensional vector given by

$$\boldsymbol{b} = \boldsymbol{P}^T(\boldsymbol{x} - \bar{\boldsymbol{x}}) \tag{2.3}$$

All image shapes are normalised to the mean shape. Each image is then warped to its control points. The same pose (translation, scale and rotation) values are used in shape normalisation so then we can sample the grey level information $\boldsymbol{g}$ from a shape-normalised face patch. By applying PCA to this data we obtain this model:

$$\boldsymbol{g} \approx \bar{\boldsymbol{g}} + \boldsymbol{P}_g \boldsymbol{b}_g \tag{2.4}$$

A further PCA is applied to the correlated shape and grey-level variations, to obtain:

$$\boldsymbol{x} \approx \bar{\boldsymbol{x}} + \boldsymbol{Q}_s \boldsymbol{c} \tag{2.5}$$

$$\boldsymbol{g} \approx \bar{\boldsymbol{g}} + \boldsymbol{Q}_g \boldsymbol{c} \tag{2.6}$$

where $\bar{\boldsymbol{x}}$ is the ean shape, $\bar{\boldsymbol{g}}$ the mean texture, $\boldsymbol{Q_s}, \boldsymbol{Q_g}$ are the matrices describing the modes of variation derived from the training set and $\boldsymbol{c}$ is a vector of the appearance parameters controlling both the shape and grey-levels of the model.

By varying the elements of $\boldsymbol{c}$ in equation 2.5 and 2.4, new shapes and images will be generated. So c_i is the variance of the i_{th} parameter given by standard deviations λ_i. To generate similar shapes to the original training data, limits of $\pm 3\sqrt{\lambda_i}$ have been applied to the parameter b_i.

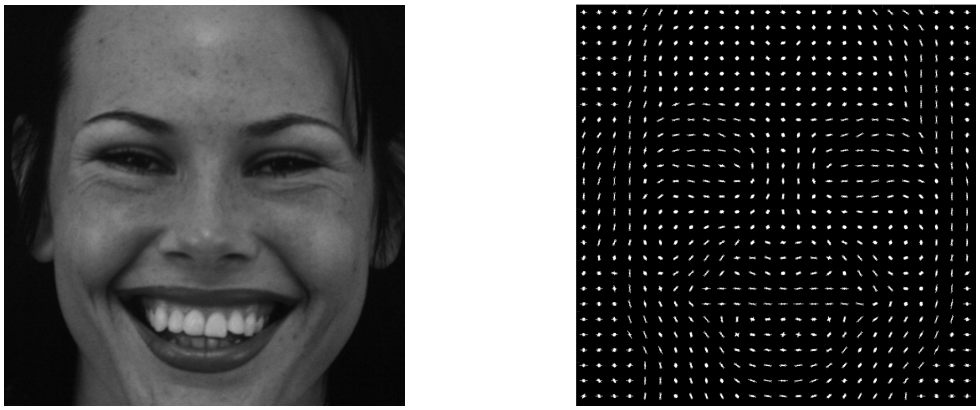
Now if we were given a new image $\boldsymbol{g_s}$, and we want to find the shape points which fit the image we need to vary $\boldsymbol{c}$ to generate new images by a set of model parameters $\boldsymbol{c}$. We can generate a hypothesis for the shape, x, and texture, $\boldsymbol{g}_m$, of a model instance, and then finding the most similar generated image for the appearance model to $\boldsymbol{g}_s$ by computing the difference, $\delta \boldsymbol{g} = \boldsymbol{g}_s - \boldsymbol{g}_m$. This is an optimisation problem to find the best $\boldsymbol{c}$ efficiently, which will generate shape landmarks which describe the face parts.

Since Edwards et al. (1998), many researches have used AAM in many applications to recognise facial expression in videos (Sung et al., 2006; Martin et al., 2008) and photos (Kanade et al., 2000; Wu et al., 2013) . Facial expressions recognition is one of the most discussed topics during the last three decades. Researchers have applied AAM to extract facial features face points to be classified using various classifiers. One of the ways that the AAM has been used in facial recognition system is

in the Action Units (AUs). The AUs system is described in the Facial Action Coding System (FACS) (Ekman, 1978), which was based on minimal muscular movements and which individually or in combinations represent all facial expressions (Cohn et al., 1998; Lucey et al., 2010). By finding the action units in a face, and by using one or a group of them together, facial expressions can be recognised. For instance, AU1 is Inner Brow Raiser, AU2 is Outer Brow Raiser, AU15 Lip Corner Depressor and AU28 Lip Suck. To determine anger expression, for example, AU23 and AU24 must be present in the AU combination, where AU1+4+15 or 11 must be present for sadness Lucey et al. (2010). To classify the extracted features to find the AUs trained classifiers have been used like neural networks trained with backpropagation (Van Kuilenburg et al., 2005) on Cohn-Kanade AU-Coded (CK+) Facial Expression Database (Kanade et al., 2000), or Support Vector Machine (SVM) in (Lucey et al., 2010) with the CK+ database as well.

On the other hand, AAM was used to detect facial expressions directly rather than AU detection. The AAM features have been classified using several classifiers such as the simple Euclidean-distance classification scheme (Ratliff and Patterson, 2008), K-nearest neighbours classification (Cheon and Kim, 2008) or SVM (Kotsia and Pitas, 2007).

The main advantage of ASM and AAM is that they have low dimension and simplicity. However, they are sensitive to error in image registration and motion discontinuities. In ASM and AAM, we depend on the facial landmarks to recognise the facial expression, so any error in representing the face by landmarks leads to a wrong recognition.



(a) A woman's face

(b) Visualization of extracted HOG

Figure 2.3: HOG descriptor Visualization

2.2.2 Image texture-based methods

Image texture-based or appearance-based methods have been widely used in computer vision applications for face recognition and facial expression recognition. The idea of image texture methods is using the image pixel values such as RGB and greyscale changes. Many methods have been proposed to describe the image and extract the image features such as Locally Binary Patterns (LBP), the Histogram of Oriented Gradients (HOG) Scale-Invariant Feature Transform (SIFT) and Speeded Up Robust Features (SURF).

2.2.2.1 Local histogram of oriented gradients (HOG)

The Local Histogram of Oriented Gradients (HOG) is a method proposed by Dalal and Triggs (2005). This method aims to describe an image with a set of local histograms. These histograms count the occurrences of gradient orientation in a local

part of the image. The HOG algorithm is similar to edge orientation histograms, scale-invariant feature transform descriptors and shape contexts, but the difference is that it is computed on a dense grid of uniformly spaced cells and that it uses overlapping local contrast normalisation for improved accuracy (Dalal and Triggs, 2005). Figure 2.3 shows a HOG feature visualisation for the face. By focusing on the image, we can see the essential features of the image.

There are several primary steps for extracting HOG features: the first is the Gamma/Colour Normalisation, where Dalal and Triggs (2005) found that gamma normalisation improves the facial expression classification rate. In fact, gamma correction is necessary as the block normalisation has the same effect. The second step is computing the image gradients by applying the 1-D central deferences. Dalal and Triggs found that this gives the best results in one or both of the horizontal and vertical directions, $[-1, 0, 1]$ for vertical and $[-1, 0, 1]^T$ for horizontal. At every pixel we calculate a value for the x-derivative and another value for the y-derivative for x and y gradient magnitudes respectively, let us call them Sx and Sy. The equations defining the gradients are, respectively being:

$$S_x(i, j) = \frac{\partial I}{\partial x}(i, j) \tag{2.7}$$

$$S_y(i, j) = \frac{\partial I}{\partial y}(i, j) \tag{2.8}$$

where I is an image, and (i, j) are the pixel coordinates. The gradient magnitude itself M is computed as the square root of the quadratic sum of each gradient com-

ponent, this is:

$$M(i,j) = \sqrt{S_x^2(i,j) + S_y^2(i,j)} \tag{2.9}$$

The gradient orientation angle is calculated by:

$$\theta(i,j) = \arctan\left(\frac{S_x(i,j)}{S_y(i,j)}\right) \tag{2.10}$$

The third step is called Orientation Binning, which aims to build a histogram of orientation for each cell (where the image was subdivided into little cells). Each pixel within the cell has a weighted vote for an orientation-based histogram channel based on the values found in the gradient results. These histograms represent the angles evenly spaced between 0° and 180° ("unsigned" gradient) or within 0° and 360° ("signed" gradient).

The final step is block-normalising histograms within each block of cells. Because of gradient strength variation as a result of local illumination variations and the foreground-background contrast, Dalal and Triggs (2005) found that some illumination normalisation must compensate for better accuracy. They explored different normalisation schemes to achieve that. Let us define first v as the vector containing all the histograms for a given block, $\|v\|_k$ the k-norm of v with $k \in 1, 2$ and let ϵ be a small constant. The normalisation schemes are:

$$L1 - norm : v \rightarrow \frac{v}{\|v\|_2 + \epsilon} \tag{2.11}$$

$$L1 - sqrt : v \rightarrow \sqrt{\frac{v}{\|v\|_2 + \epsilon}} \tag{2.12}$$

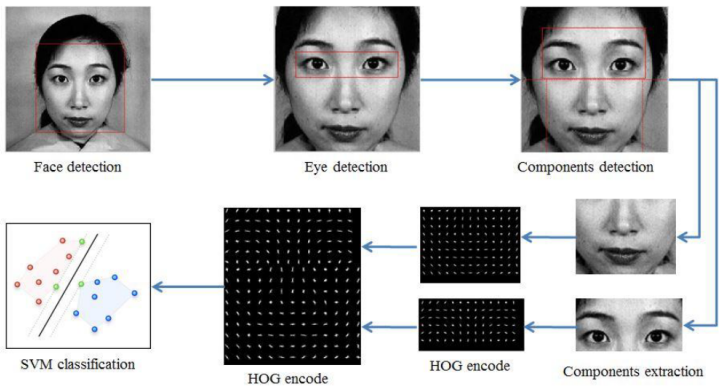


Figure 2.4: An overview of the face recognition system with HOG (Chen et al., 2014)

$$L2 - norm : v \rightarrow \frac{v}{\sqrt{\|v\|_2^2 + \epsilon^2}} \tag{2.13}$$

Dalal and Triggs' experiments found that L1-sqrt and L2-norm perform similarly, but L1-norm decreases performance by 5%. Not normalising reduces the performance enormously by around 27% (Dalal and Triggs, 2005).

The HOG method has been wildly used in computer vision to recognise objects (Dahmane and Meunier, 2011; Chen et al., 2014; Carcagnì et al., 2015). HOG features have used in facial expression recognition with multi-class RBF-SVM by extracting dense grid-based HOG features from images (Dahmane and Meunier, 2011), where they used a cropped region from the aligned face and divided it into (48) squares 8 rows and 6 columns. In (Dahmane and Meunier, 2011), the GEMEP-FERA dataset was used for training and testing 5 facial expressions: anger, fear, joy, relief and sadness. The face has been divided into its main parts and then the method extracts each part's HOG features, as shown in figure 2.4.

The HOG descriptor could be effectively exploited for facial expression recogni-

tion purposes. The configuration of HOG parameters can provide a robust image descriptor, which allows for a high classification performance for facial expressions (Carcagnì et al., 2015).

2.2.2.2 Scale-invariant feature transform (SIFT) and speeded up robust features (SURF)

SIFT has been proposed by Lowe (2004). SIFT contains keypoint localisation and construction of key-point descriptor. SIFT, particularly the SIFT descriptor, is a popular method used in computer vision and object recognition and it has been proven to be very effective (Berretti et al., 2010; Karami et al., 2017). The SIFT algorithm has 4 main steps. The first is to estimate a scale-space extremum using the Difference of Gaussian (DoG). Secondly, key point localisation must be calculated where the key point candidates are localised and refined by eliminating the low contrast points. Thirdly, the key point orientation assignment is based on the local image gradient, and lastly, a descriptor generator to compute the local image descriptor for each key point based on image gradient magnitude and orientation (Karami et al., 2017).

The main SIFT advantage is its stability for images in different resolutions, so it provides good performance in machine vision applications. In facial expression methods, SIFT features represent the same object with different expressions and illumination. Researchers have used SIFT descriptors with 2D and 3D images (Zhang et al., 2008; Berretti et al., 2010; Soyel and Demirel, 2012). Zhang et al. (2008) proposed a SIFT and SVM based method to investigate the robustness of SIFT

features for various training images on face recognition and used the ORL and the Yale database for experiments and the found the method managed to handle the expression problems better than other algorithms at that time. Figure 2.5 shows an example of images with extracted SIFT features.

Figure 2.5: An example of images with extracted SIFT features. The images represent the same object with different expressions and illumination (Zhang et al., 2008)

Speeded Up Robust Features (SURF) was first presented by Herbert Bay as a novel scale- and rotation-invariant interest point detectors and descriptors (Bay et al., 2006, 2008). SURF is similar to the SIFT descriptor (Lowe, 1999) properties. SURF is faster than SIFT and gives as good a performance as SIFT (Panchal et al., 2013). (D-SURF) is a local feature detector and descriptor. The D-SURF algorithm is based on the same principles and steps as SIFT, but the details in each step are different. The algorithm has two main parts: Firstly "interest point detection" was selected at important locations in the image, for instance, at T-junctions, corners and blobs. To achieve that, the algorithm uses a very basic Hessian matrix approximation because of its good performance in accuracy (Bay et al., 2008). The next step is to find the orientation of the point of interest to achieve rotational invariance. Haar-wavelet

responses within the interest-point circular neighbourhood of radius 6-scale around the interest-point.

2.2.2.3 Local Binary Patterns (LBP)

The Local Binary Patterns (LBP) method was first proposed by Ojala et al. (1996) as a texture descriptor depending on statistical analysis. Since then, it has been widely used for face analysis due to its classification performance (Zhou et al., 2013). The LPB operator compares each pixel in a 3x3 neighbourhood of the pixel to the central value and constructs a binary digit number from the result, thus computing the local texture characteristics. One of the most important advantages of LBP features is their tolerance against illumination variation (Shan et al., 2009). Let us, therefore, define texture T in a local neighbourhood of a greyscale image as the joint distribution of the grey levels of $P + 1$ $(P > 0)$ image pixels:

$$T = (I_c, I_0, ..., I_{P-1}) \tag{2.14}$$

where I_c corresponds to the grey value of the centre pixel of a local neighbourhood. $I_p(p = 0, ..., P - 1)$ is the grey values of P equally spaced pixels on a circle of radius R $(R > 0)$ that form a circularly symmetric set of neighbours.

To achieve invariance with respect to any monotonic transformation of the grayscale, only the signs of the differences are considered:

$$T = (s(I_0 - I_c), ..., s(I_{P-1} - I_c)) \tag{2.15}$$

where I_c corresponds to the grey value of the centre pixel (x_c, y_c), into the grey values

of the 8 surrounding pixels, and function $s(x)$ is defined as:

$$s(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases} \tag{2.16}$$

Finally the LBP describes the local image texture around (x_c, y_c):

$$LBP_{P,R}(x_c, y_c) = \sum_{p=0}^{P-1} s(g_p - g_c)2^p \tag{2.17}$$

The number of neighbours used to compute the basic LBP for each pixel in the input image is 8.

LBP has been used with many classifiers to recognise facial expression because of its advantages, for example, its tolerance of monotonic illumination changes and its computational simplicity (Huang et al., 2011). Shan et al. (2005) used simple Local Binary Patterns (LBP) with the Support Vector Machine (SVM). They have tested the extracted LBP features with linear, polynomial and RBF kernels SVM to classify 7 facial expressions. They used the Cohn Kanade Facial Expression Database, which was produced by Kanade et al. (2000). It contains faces of 100 university students from age 18 to 30 years. Shan et al. (2005) compared their results with Gabor wavelets, and they have found that LBP with SVM gave better classification accuracy than Gabor wavelets, and saved computational resources. They also proved that LBP gives good results with different resolutions, even with low-resolution images (Shan et al., 2005). In the same area, Shan et al. (2009) conducted a comprehensive study for facial expression recognition methods based on Local Binary Patterns. They found that the LBP features are effective and efficient for facial expression recognition and give good results with low-resolution images.

LBP has been used for both frontal faces, and angle pose faces. Moore and Bowden (2011) proposed a multi-view facial expression recognition method using some extensions including multi-scale local binary patterns (LBP^{ms}) and local Gabor binary patterns ($LGBP$). They have tested it on photos from multiple datasets (Gross et al., 2010) to see how head pose affects facial expression using SVM classifier.

During the last decade, researchers have proposed many methods based on LBP and its extensions. Some of them have tried to divide the face into equal blocks into grids (Moore and Bowden, 2011). Others have tried to divide the face into its main parts: eyes, nose and mouth as in Khan et al. (2013) who proposed a pyramidal local binary pattern (PLBP) operator to recognise six facial expressions. They have tested the extracted features using 4 classifiers: nearest neighbours (2NN), Random Forest (RF), SVM and decision tree. They used in experiments two datasets: CK+ and FG-NET. They found that the features extracted using PLBP have a strong discriminative ability as the recognition result for 6 expressions is not affected by choice of classifier.

2.2.3 Classification

In machine learning, classification is the problem of distinguishing to which set of classes a new observation belongs, based on a training set of data containing observations whose class membership is known.

Let us consider a given dataset $D = \{\boldsymbol{x}_n, y_n\}_{n=1}^{N}$, where each data point $\boldsymbol{x} \in \mathbb{R}^d$ is paired with a known discrete class label y_n where $y \in \{1, 2, ..., K\}$. The main goal of classification is to train a classifier to be able to classify any arbitrary d-dimensional

data point as one of the K discrete classes.

Classification has strong roots in probabilistic modelling. The idea is that we form a joint probability distribution $p(\boldsymbol{X}, Y)$ over the input $\boldsymbol{X}$ and label Y, and that we classify an arbitrary data point $\boldsymbol{x}$ with the class label that maximizes the joint probability:

$$\hat{y} = \underset{\mathbf{k}}{argmax}\, p(\boldsymbol{x}, Y = K) \tag{2.18}$$

.

If we aim to minimise the chance of predicting $\acute{\boldsymbol{x}}$ to the wrong class, then we should choose the class with the highest posterior probability. In this thesis, two classifiers have been considered, namely support vector machines and random forests. These will be discussed in detail in the following sections.

In this thesis we will use two popular data classification methods, random forests (RF) (Breiman, 1999, 2001) and support vector machines (SVM) (Cortes and Vapnik, 1995). SVMs aim to find a hyperplane (linear decision surface) which divides the data into two classes and it has the largest margin between the closest elements of the two classes to each other. This hyperplane is called the Optimal Separating Hyperplane (OSH) which minimises the misclassification. These elements are called vector machines. It is not always easy to find a linear decision surface, so the radial basis function (RBF) is used as a kernel after the data has been mapped into a higher dimensional space (Davison et al., 2014). An RF model uses the bootstrap method to build the n_{tree} decision randomly. Each tree is provided with randomly selected samples form the training input. The trees will vote together to give the

final decision by combining them in a forest.

2.2.4 Support Vector Machine (SVMs)

SVMs (Cortes and Vapnik, 1995) is a binary classification method, and it was proposed by Vladimir Vapnik in 1979 and first published in 1995. So if we have labelled training data $D = \{\boldsymbol{x}_n, y_n\}_{n=1}^{N}$ where $\boldsymbol{x} \in \mathbb{R}^d$ and $y \in \{1, -1\}$. The idea of SVMs is to find the Optimal Separating Hyperplane (OSH) which separates the two classes of d-dimensional data.

If the data is linearly separable, the two parallel hyperplanes are selected to separate the two classes of data so that the distance between them is as large as possible. The vectors that define the hyperplanes are called the support vectors.

The area bounded by these hyperplanes is called the margin. The maximum margin hyperplane is the maximum distance between the two hyperplanes and another hyperplane that lies in the middle between them. The distance between the two hyperplanes is $\frac{2}{\|\boldsymbol{w}\|}$. In order to maximise the distance between the planes we need to minimise $\|\boldsymbol{w}\|^2$. The hyperplanes can be described by the following:

$$\boldsymbol{w}^T\boldsymbol{x}_i + b \geq 1 \quad \text{for} \quad y_i = +1 \tag{2.19}$$

for anything on or above this boundary is of one class, with label 1.

$$\boldsymbol{w}^T\boldsymbol{x}_i + b \leq -1 \quad \text{for} \quad y_i = -1 \tag{2.20}$$

anything on or below this boundary is of the other class, with label -1, where $\boldsymbol{w}$ is the weight vector and b is the bias.

The optimisation problem can be solved using the Lagrange multiplier method. The objective function to be minimized in the Lagrangian form can be written as:

$$L(\boldsymbol{w}, b, \alpha) = \frac{1}{2} \parallel \boldsymbol{w} \parallel^2 \sum_{i=1}^{N} \alpha_i \big[y_i(\boldsymbol{w}^T \boldsymbol{x}_i + b) - 1 \big] \tag{2.21}$$

The Lagrange multipliers should be non-negative ($\alpha_i > 0$). In order to minimise the Lagrangian form, its partial derivatives are obtained with respect to $\boldsymbol{w}$ and b and are equated to zero.

$$\boldsymbol{w} = \sum_{i=1}^{N} \alpha_i y_i \boldsymbol{x}_i \tag{2.22}$$

$$\sum_{i=1}^{N} \alpha_i y_i = 0 \tag{2.23}$$

substituting these values back in 2.21, we obtain:

$$L(\boldsymbol{w}, b, \alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i \cdot \boldsymbol{x}_j \tag{2.24}$$

Training is therefore accomplished by maximising L. Training instances having $\alpha > 0$ are the 'support vectors'.

These training instances are used to obtain the decision boundary parameters $\boldsymbol{w}$ and b. SVM outputs the following class output $\acute{y}_i$:

$$\acute{y}_i = f(\boldsymbol{x}_i) = sign\left[(\sum_{i=1}^{N} \alpha_i y_i (\boldsymbol{x}_i \cdot \boldsymbol{x}) + b) \right] \tag{2.25}$$

In some cases, it is better to misclassify some of the training errors in order to get a decision boundary plane with maximum margin. If we got a decision boundary with no training errors, but a smaller margin may lead not to classify unknown

samples correctly. A decision boundary with a larger margin and few training errors can classify the unknown samples more accurately. For this, we need a decision boundary between the margin and the number of training errors. This decision boundary is called a soft margin. Slack variables ξ is introduced to account for the soft margin. Also, a penalty for the training error C is be introduced order to balance the margin value and the number of training errors.

The objective function for the optimization problem will be minimization of

$$\frac{1}{2} \parallel \boldsymbol{w} \parallel^2 + C \sum_{i=1}^{N} \xi_i \tag{2.26}$$

So the hyperplanes can be described by the following:

$$\boldsymbol{w}^T \boldsymbol{x}_i + b \geq +1 - \xi_i \quad \text{for} \quad y_i = +1 \tag{2.27}$$

and

$$\boldsymbol{w}^T \boldsymbol{x}_i + b \leq -1 + \xi_i \quad \text{for} \quad y_i = -1 \tag{2.28}$$

Data is not always linear separatable, so researchers suggested to use essential kernel functions such as Radial Basis Function (RBF). The most often used kernel functions are the radial basis function (RBF):

$$K(\boldsymbol{x}_i, \boldsymbol{x}) = exp(-\gamma \parallel \boldsymbol{x}_i - \boldsymbol{x} \parallel^2), \gamma > 0 \tag{2.29}$$

where γ is a parameter that can be seen as the inverse of the radius of influence of samples selected by the model as support vectors. With low γ, the curvature of the decision boundary is small, and thus, the decision region is wide. When γ is huge, the curvature of the decision boundary is high, which creates islands of decision-boundaries around data points. When using a kernel function, the decision function

becomes:

$$\acute{y}_i = f(\boldsymbol{x}_i) = sign\left[(\sum_{i=1}^{N} \alpha_i y_i K(\boldsymbol{x}, \boldsymbol{x}_i) + b)\right] \tag{2.30}$$

In this thesis, we applied cross-validation and Bayesian optimisation method (Mockus, 2012) to find the optimal parameters of SVM with RBF kernel.

2.2.5 Random forest

Random forest is a popular method in machine learning because of its capacity to operate within large multi-class datasets and give high accuracy results (Fanelli et al., 2011). They have an excellent generalisation ability; they are very fast to train and parallelise (Breiman, 2001; Belle, 2008). RF classifier contains a combination of tree classifiers; each tree gives a unit vote for the most popular class to classify an input vector (Breiman, 1999).

To understand the random forest algorithm, we need to understand the basic idea of decision trees. Each tree is a collection of nodes and edges organised in a hierarchical structure as shown in figure 2.6. In a decision tree, the top node is called root connected with two children nodes. The nodes at the bottom are called leaves. Decision work according to an algorithm called Classification and Regression Tree (CART) algorithm has been proposed by Breiman et al. (1983).

In a decision tree, T the root node receives the entire training set, and each node asks a true/false question about one of the feature, and in response to this question, the data is split into two subsets. Theses subsets then become the input for two child nodes added to the tree. The goal for the question is to produce the purest possible distribution of the classes at each node. The trick to building an effective tree is

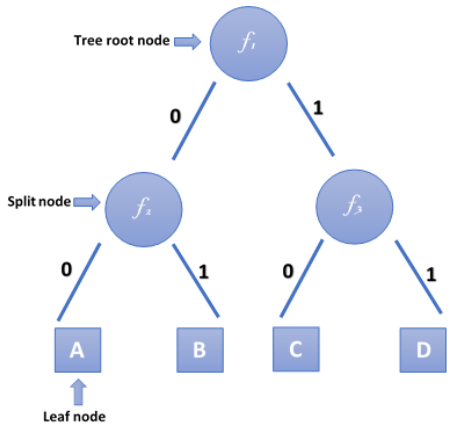


Figure 2.6: Simple decision tree

to understand which question to ask and when. To do that we need to quantify which question helps to unmix the classes, and to achieve that, the Gini impurity is used which finds the threshold value. To quantify how much the question reduces the uncertainty, it uses the information gain (Breiman et al., 1983). The data will continue dividing until there are no further question to ask, at which point a leaf node will be added.

The main idea of random forests (Breiman, 2001) $\mathcal{F}$ is to make a group (ensemble) of $\boldsymbol{F}$ decision trees vote together $\mathcal{F} = \{\boldsymbol{F}_1, ..., \boldsymbol{F}_t, ...\boldsymbol{F}_T\}$, where each tree node in the random forests classifier is a weak classifier, each tree gets a "vote" in classifying (Breiman, 1996, 1999, 2001). This combination of ensemble trees provides very good generalisation. A dataset, many random subset S_t can be generated to be processed by constructing decorrelated a tree $\boldsymbol{F}_t$ for each S_t. According to Breiman (2001) randomization method "Bagging", for a given dataset $D = \{\mathbf{x}_n, y_n\}_{n=1}^{N}$ is divided into random smaller subsets S_t. Each data subset S_t is called a bootstrap. By

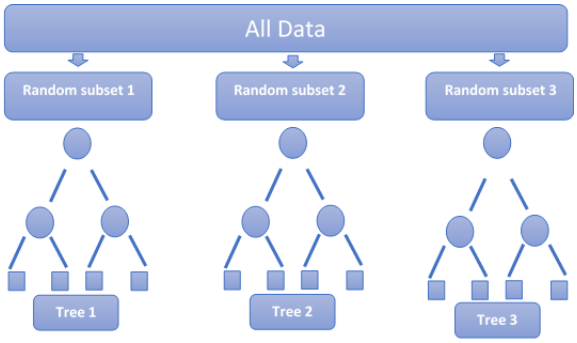


Figure 2.7: New subsets Randomisation by bagging method

growing a tree $\boldsymbol{F}_t$ for each bootstrap S_t, an ensemble of decision trees work together to vote on a new unseen observation $\acute{\boldsymbol{x}}$.

A new unseen observation $\acute{\boldsymbol{x}}$ should be predicted according to which class it refers to, by sending the unseen feature through all trees in the forest and combining the tree posteriors. The class prediction for a new observation is the class that yields the largest weighted average of the class posterior probabilities computed using the selected trees only.

For each class $y \in Y$, the prediction for new observation $\acute{\boldsymbol{x}}$ computes $p(y_i|\acute{\boldsymbol{x}})$ which is the estimated posterior probability of class y for the given observation $\acute{\boldsymbol{x}}$. Random forests apply the weight concept for to consider the impact of the results from any decision tree. Tree $\boldsymbol{F}_t$ with high error rate are given low weight value and vise versa. This would increases the decision impact of the trees with a low error rate. The prediction computes the weighted average of the class posterior probabilities over the selected trees as

$$p(y_i|\acute{\boldsymbol{x}}) = \frac{1}{\sum_{t=1}^{T} \Omega_t Z(t \in S)} \sum_{t=1}^{T} \Omega_t \hat{y}_t I(t \in S) \tag{2.31}$$

where $\hat{y}_t$ is the prediction from tree t in the ensemble, S is the set of indices of selected trees that comprise the prediction and Ω_t is the weight of the tree, $Z(t \in S)$ is 1 if t is in the set S, and 0 otherwise..

The RFs classifier has proficient power to gauge the importance of each features variable (predictor) (Breiman, 2001) by calculating how much a prediction error increases or decreases when the out-of-bag (OOB) error for that variable is permuted while all others are passed on unaltered. The computations are carried out tree by tree as the random forest is built (Breiman, 2001, 2002; Liaw and Wiener, 2002).

Breiman has proposed a method to evaluate the variable importance by measuring the Mean Decrease Accuracy (MDA) of the forest when the values of $\boldsymbol{x}_i$ are randomly permuted in the out-of-bag samples. For each tree, the prediction, error rate for the classification on the out-of-bag portion of the data is recorded. After that, the same is done after permuting each predictor variable. The differences between the two are then averaged over all of the trees. In other words, after each tree is built, the values of the i_{th} variable in the out-of-bag examples are randomly permuted, and the out-of-bag data is run down the corresponding tree. The classification is given for each $\boldsymbol{x}_i$ that is out of the bag is saved. At the end of the run, the plurality of out-of-bag class votes for $\boldsymbol{x}_i$ with the i_{th} variable noised up is compared with the true class label of $\boldsymbol{x}_i$ to give a misclassification rate. The output is the per cent increase in the misclassification rate as compared to the out-of-bag rate with all of the variables intact (Breiman, 2001; Louppe et al., 2013).

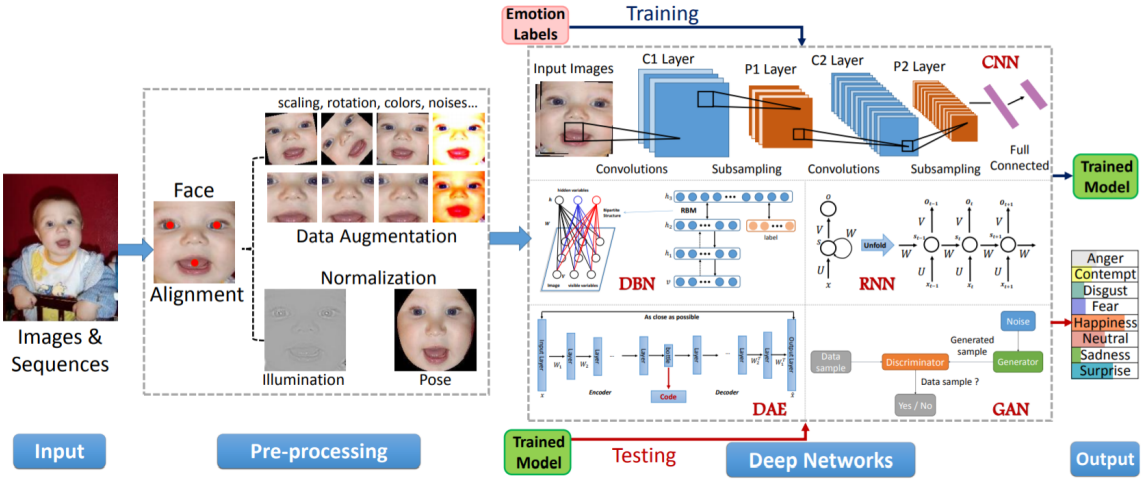


Figure 2.8: The general pipeline of deep facial expression recognition systems (Li and Deng, 2018).

2.2.6 Deep learning and Convolutional Neural Networks

Deep learning is a machine learning technique that learns the features and tasks directly from data. Deep learning tries to capture high-level abstractions and order them into hierarchical architectures of many non-linear transformations and representations (Li and Deng, 2018). Figure 2.8 illustrates the traditional architectures consisting of deep neural networks.

Deep learning architectures such as convolutional neural networks (CNN) (LeCun et al., 1998), deep belief networks (DBN) (Liu et al., 2014b), deep auto-encoders (DAE) Hinton and Salakhutdinov (2006), recurrent neural networks (RNN) (Caridakis et al., 2006) and generative adversarial network (GAN) (Goodfellow et al., 2014) have been applied to computer vision, speech recognition, natural language

processing, audio recognition, social network filtering, machine translation, bioinformatics and medical image analysis, and they have given good results. Deep learning includes multiple layers of non-linear processing units for feature extraction and transformation. Each successive layer uses the output from the previous layer as input. In image recognition applications like facial expression, the raw input is a matrix of pixels. The deep learning process can learn which features to place in which level on its own optimally.

Liu et al. (2014a) have used a combination of HOG, SIFT and CNN are extracted at each frame of the Acted Facial Expression in Wild (AFEW) (Dhall et al., 2012) videos In the classification stage, three types of classifiers are investigated for comparisons, kernel SVM, logistic regression, and partial least squares. Meng et al. (2017) proposed an identity-aware CNN (IACNN) with two identical sub-CNNs. They used the expression-sensitive contrastive loss to learn expression discriminative features, and the other stream used an identity-sensitive contrastive loss to learn identity-related features for identity-invariant.

Despite the widespread use of deep learning techniques, some limitations still face this field, like the need a large amount of computational power. It is costly to train due to complex data models involved. For this reason, we focused on traditional methods like random forest and SVMs.

2.3 Facial expression datasets

Within the past two decades, significant effort has been made to build databases for use in facial expression recognition systems. These databases have been used for

machine training and testing purposes. For example, MMI dataset (Pantic et al., 2005) was a major improvement. It contains nearly 1500 samples of static images and videos from 19 male and female subjects in both frontal and profile view, displaying several facial emotions, single AU and multiple AU activation, the MMI Facial Expression Database provides a large test-bed for research on automated facial expression analysis. As another example, considerable progress has been achieved with Affectiva-MIT Facial Expression Dataset (AM-FED) (McDuff et al., 2013) which presents a new dataset of labelled data recorded over the internet of people naturally viewing online media. The AM-FED contains 242 webcam videos recorded in real-world conditions, and 168,359 frames labelled for the presence of 10 symmetrical FACS action units. AM-FED is labelled frame by frame, and it has action units in addition to the location of 22 automatically detected landmark points. The 4D Database for Facial Expression Analysis and Biometric (4DFAB) (Cheng et al., 2017) includes videos of 180 subjects taken in 4 different sessions spanning over five years. It contains 4D videos of subjects representing both spontaneous and posed facial behaviours. 4DFAB contains large scale database of dynamic high-resolution 3D faces (over 1,800,000 3D meshes).

Some of the current databases that have mostly been used in the past two decades are the Karolinska Directed Emotional Faces (KDEF) Lundqvist et al. (1998), the AR database (Martinez, 1988), the Japanese Female Facial Expression Database (JAFFE) (Lyons et al., 1998), Cohn-Kanade facial expression database Cohn-Kanade (CK) (Kanade et al., 2000) and the extended Cohn-Kanade dataset (ck+) (Lucey et al., 2010), the MMI facial expression database (Pantic et al., 2005). Table 2.1 is

Table 2.1: Databases summary

Database	Samples	Facial expressions
AR (Martinez, 1988)	508 images	4 basic expressions plus contempt and neutral
JAFFE (Lyons et al., 1998)	213 images	6 basic expressions plus neutral
KDEF (Lundqvist et al., 1998)	4,900 images	6 basic expressions plus neutral
MMI (Pantic et al., 2005)	740 images and 2,900 videos	6 basic expressions plus neutral
BU-3DFE (Yin et al., 2006)	2,500 images	6 basic expressions plus neutral
CK+ (Lucey et al., 2010)	10780 image sequences	6 basic expressions plus contempt and neutral
RaFD (Langner et al., 2010)	1,608 images	6 basic expressions plus contempt and neutral
TFD (Susskind et al., 2010)	112,234 images	6 basic expressions plus neutral
Multi-PIE (Gross et al., 2010)	755,370 images	Smile, surprised, squint, disgust, scream and neutral
Oulu-CASIA (Zhao et al., 2011)	2,880 image sequences	6 basic expressions
AM-FED (McDuff et al., 2013)	168,359 image sequences and 242 videos	6 basic expressions
FER-2013 (Goodfellow et al., 2013)	35,887 images	6 basic expressions plus neutral
SFEW 2.0 (Dhall et al., 2015)	1,766 images	6 basic expressions plus neutral
EmotioNet (Fabian Benitez-Quiroz et al., 2016)	1,000,000 images	23 expressions or compound expressions
4DFAB (Cheng et al., 2017)	1,800,000 3D meshes	6 basic expressions
AFEW 7.0 (Dhall et al., 2017)	1,809 videos	6 basic expressions plus neutral
RAF-DB (Li et al., 2017)	29672 images	6 basic expressions plus neutral and 12 compound expressions
AffectNet (Mollahosseini et al., 2017)	450,000 images	6 basic expressions plus neutral
ExpW (Zhang et al., 2018)	91,793	6 basic expressions plus neutral

an overall summary for most of the facial expression databases.

Scientists over the last two decades have developed many databases in the field of facial expressions; the scientific community is moving towards databases containing more spontaneous emotions rather than acted ones.

2.4 Conclusion

In this chapter, we have reviewed some of the basic concepts regarding facial expression recognition and related topics such as features extraction and classification which is necessary to appreciate our proposed facial expression recognition method. We talked about the background of facial analysis, and we briefly introduced the current techniques that might be useful for our thesis.

In this thesis, we experiment with a combination of texture features, rather than using only one. Due to the high-dimensionality of this combination, it is important to remove any unnecessary features which may not offer any benefits in classification. Random forests provide a straightforward method for features importance estimation that will save a lot of time and effort when dealing only with the data that affects the classification.

Deep learning algorithm has changed the entire landscape over the past few years, but some limitations still face this field, like the need a large amount of computational power. It is costly to train due to complex data models. Moreover, deep learning requires expensive GPUs, which increases the cost to the users. A considerable drawback and difficulty in the use of deep learning is the need for large datasets, which are used as the input during the training procedure. Moreover, deep learning algorithms require a large amount of training data may take longer to train than simpler models. This requires a large number of hyperparameters such as the number of layers or the type of activation functions. This imitation pushes us to try to improve the traditional classifiers such as random forests.

Active shape models and Active appearance models only use shape constraints and do not take advantage of all the available information texture across the target object. For this reason, we take advantage of the image texture, which contains more information.

In the following chapters, we show how the combination of texture features may be used to classify the facial emotions with the random forests and SVMs, and how using of the important features improves the classification performance.

This work focuses on using machine learning methods and algorithms in order to evaluate the classification of spontaneous facial expressions. An experimental methodology will be adopted in this thesis; we will experiment with random forest and SVMs with image texture features.

Chapter 3

Emotional faces in the wild database

Contents

3.1 Introduction 40

3.2 Data collection 41

3.3 Summary of citizens' classification 46

3.4 Conclusion 49

3.1 Introduction

Training and testing the facial expression methods need databases containing human facial expressions. Researchers in the psychological field and machine learning have built various databases (Lundqvist et al., 1998; Kanade et al., 2000; Pantic et al.,

2005; Langner et al., 2010; Anitha et al., 2010; Lucey et al., 2010; El Ayadi et al., 2011). The main disadvantage of many of the available datasets is that they contain unnatural expressions because the people in the databases are actors expressing their emotions as they have been asked. In this thesis, we are keen to work with more natural facial expressions in order to investigate how the trained model can work with various people who expressed their emotions spontaneously.

The most commonly used method of constructing the databases is to ask some actors or models to show the required facial expressions. Most researchers have used relatively unnatural datasets because, in real life, natural facial expressions are different from those made by actors. The expressions of the human face are varied and show some differences between cultures and even from one person to another (Ekman, 1973). Just as human sometimes find it difficult to recognise some facial expressions, machines also face the same challenges. In this chapter, we introduce the "Emotional Labelled Faces in the Wild" dataset (eLFW), a citizen-labelling of 1310 faces from the Labelled Faces in the Wild dataset (Huang et al., 2007b). To collect this data, we built a website and asked citizens to label images from the LFW dataset according to the emotional expression displayed. This chapter presents the process of the new dataset collection and labelling and shows some of the summary statistics of the dataset.

3.2 Data collection

Our work began with building a new natural facial expression database using a current database called Labelled Faces in the Wild (LFW) (Huang et al., 2007a;

Figure 3.1: Examples of citizen labelling for LFW.

Learned-Miller, 2014).

The LFW is a database of facial photographs designed for analysing the problem of unconstrained face recognition. The dataset contains 13,233 images of individual faces collected from the web and aligned using deep funnelling (Huang et al., 2012), which is a combination of unsupervised joint alignment with unsupervised feature learning. Each face photo has been manually labelled with the name of the person pictured. A total 1680 people in the dataset appear in two or more different photos in the database. The only constraint regarding these faces is that they were discovered by the Viola-Jones face detector (Viola and Jones, 2001). Figure 3.1 shows some examples of LFW photos.

Since LFW photos were not labelled by emotion, our first goal was to build a website to collect some data and information from people around the world in order to label the facial expression. This website aimed to build an extensive database of

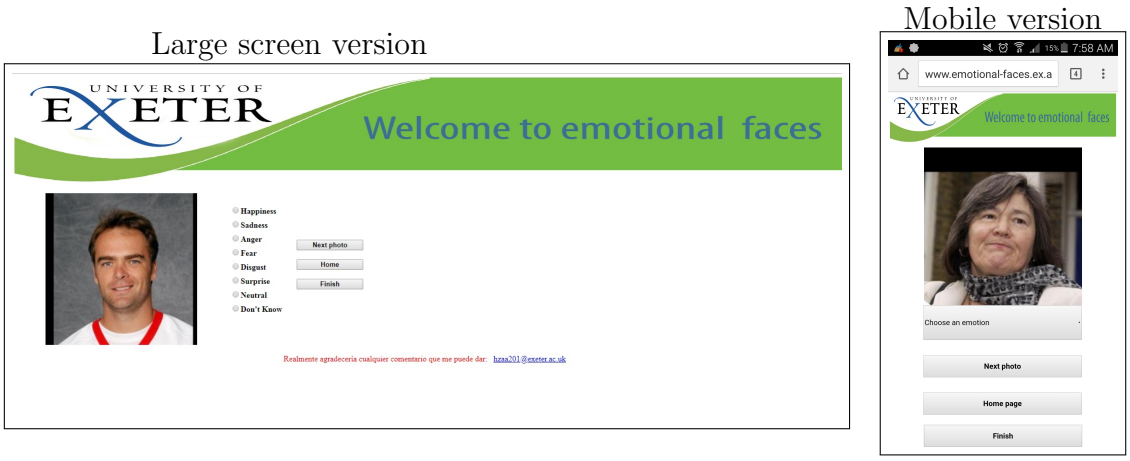


Figure 3.2: Screenshots two versions of the emotional faces website

real faces together with the emotion that they are expressing. Many facial photos were shown to the website visitors, and they were asked to choose the emotion that best matched the emotion being expressed by the face. For each face, people were asked to determine the emotion displayed from amongst the following: happiness, sadness, anger, fear, disgust, surprise, neutral, and don't know. They could label as many or as few as they wished. Since the labels assigned differed between the annotators and the images presented in a randomised order, images were retained in a pool of images to be labelled until they had been assigned labels by at least four different voters. The consensus emotion that is the modal classification was sufficiently unequivocal.

We used ASP.Net to build the website, and SQL Server to build the database. ASP.NET is for building web pages and websites with HTML, CSS, JavaScript and server scripting, while SQL Server is an efficient relational database management system (RDBMS) from Microsoft.

Two versions of the website were designed to be more usable for all users, those using PCs as shown in figure 3.2 (left), smartphones and tablet as shown in figure 3.2 (right). The website automatically redirects users to the appropriate version. This website has been tested many times to make sure that it is easy to use by users.

To evaluate the citizen labelling, we also used faces from a commonly use posed-dataset, the Karolinska Directed Emotional Faces (KDEF) (Lundqvist et al., 1998). Figure 3.3 shows some samples. KDEF contains a set of 4900 pictures of human facial expressions of emotion. The dataset contains 70 individuals, each displaying 7 different emotional expressions, with each expression being photographed (twice) from 5 different angles. In our experiments with KDEF, we used only frontal faces, which means 70 images for each facial expression, 490 images in total.

Website visitors were shown different photos chosen randomly from the two databases. The probability that displayed faces came from LFW was 0.9, so on average 1 in 10 of the faces for labelling came from KDEF data set.

Ambiguous classifications were avoided by calculating the entropy of the empirical distribution of classifications. Let p_n be the proportion of citizens' votes for the n^{th} emotion class ($n = 1, \ldots, 8$), then the entropy, $H = -\sum_n p_n \log_2 p_n$, measures the agreement between the annotators. The entropy is maximised when all classes are assigned in equal proportion and is minimised when images are assigned to only a single class. We, therefore, kept an image in the pool of images to be labelled until the entropy of the citizens' assignments was less than 1 bit, which means there is a consensus. Images that did not receive an unambiguous classification after 15 votes, and images for which the consensus was "don't know" were rejected. As mentioned

Figure 3.3: Faces samples from KDEF database which posed expressions

before, each face was classified by several citizens in order to obtain a consensus emotion. Each face image was classified at least 4 times and no more 15 anyway.

Table 3.1 illustrates the voting procedure for example images. It is clear that the best entropy value for the first photo, which was zero, that means the four voters have voted the same, which is the minimum entropy value and the best consensus. The second one, the entropy value was two, that means each voter differ from others; this case called the maximum entropy value. The third photo in table 3.1 shows that the entropy value was 1.5, which means that two votes have voted one selection, and the other two have voted in another two selections. Finally, the last photo shows shows 2 selections have been voted by fifty per cent for each, and that the entropy value is one. The new dataset involves useful data from each of the website visitors,

Table 3.1: Voting data sample

Photo Name	Happiness	Sadness	Anger	Fear	Disgust	Surprise	Contempt	Frustration	Neutral	Undefined	Entopy
Aaron_Peirsol_0001.jpg	1	0	0	0	0	0	0	0	0	0	0
Colin_Powell_0233.jpg	0	0	0.25	0	0.25	0.25	0	0	0.25	0	2
David_Anderson_0001.jpg	0.25	0	0.25	0	0	0	0	0	0.50	0	1.5
Chloe_Sevigny_0001.jpg	0.5	0	0	0	0	0	0.5	0	0	0	1

and all visitor votes will be classified as the table which shows four visitors have voted on four photos, and the entropy for each photo votes.

In addition to the LFW images, approximately 1 in 10 images presented to the citizens were a KDEF posed image. This allowed us to check the integrity of the individual annotators and, as discussed below, investigate the human performance on the KDEF data. Table 3.2 shows that the KDEF images were, unsurprisingly, easier than the LFW data for the citizens to classify, requiring fewer votes to reduce the entropy below the acceptance threshold.

3.3 Summary of citizens' classification

Table 3.2 shows summary statistics of the new data set collection; 135 visitors correctly voted (1588) photos from both datasets, KDEF and LFW. To achieve low entropy values, the citizens made approximately 7 votes on average for each LFW image and only five votes for KDEF. The average entropy of the votes for the accepted LFW photos was 0.905 bits, and for KDEF is 0.668 bits. It is clear that, unsurprisingly, the KDEF images were easier to classify than the LFW images.

Table 3.3 shows that the citizens classified the 278 KDEF photos as shown in table 3.3, which were distributed as: 49 fear, 35 anger, 42 disgust, 47 happy, 35

Table 3.2: Summary of statistics of eLFW database.

Number of users	135
Accepted LFW photos	1310
Accepted KDEF photos	278
Mean number of votes for accepted KDEF photos	4.989
Average votes entropy for accepted KDEF photos	0.668
Mean number of votes for accepted LFW photos	7.203
Average votes entropy for accepted LFW photos	0.905

neutral, 34 sad and 36 surprised. The overall agreement of voters with the KDEF labelling was 80.6%, because of the similarity between some expressions like fear and disgust. As the confusion matrix in table 3.3 shows, there was complete agreement with the KDEF labelling for happy and neutral facial expressions, but only 42.9% for fear (confused principally with disgust and surprise), 77.8% for surprise (confused principally with fear and anger), and 78.6% for disgust (confused principally with sadness and surprise). The mean number of votes and the average votes entropy for the accepted KDEF photos values were both less than those for eLFW, which mean that it was easier for the citizens to classify the KDEF images than the eLFW. Figure 3.4 shows examples of faces for which the citizen consensus differed from the KDEF labelling. We conclude that some facial expressions are similar, and even humans may be confused while determining what the expression is, so machines may face the same problems while recognising the expression.

Table 3.3: Confusion matrices for the citizens' performance on KDEF images. True classes are shown by rows, with assigned classes in columns. FE: fear; AN: anger; DI: disgust; HA: happiness; NE: neutral; SA: sadness; SU: surprise.

	FE	**AN**	**DI**	**HA**	**NE**	**SA**	**SU**
FE	**0.429**	0.020	0.347	0.000	0.000	0.061	0.143
AN	0.057	**0.800**	0.086	0.000	0.000	0.029	0.029
DI	0.000	0.024	**0.786**	0.000	0.000	0.048	0.143
HA	0.000	0.000	0.000	**1.000**	0.000	0.000	0.000
NE	0.000	0.000	0.000	0.000	**1.000**	0.000	0.000
SA	0.059	0.000	0.059	0.000	0.000	**0.853**	0.029
SU	0.139	0.056	0.000	0.000	0.000	0.028	**0.778**

Figure 3.4: Examples of faces for which citizens' votes differ from the KDEF labelling. KDEF labels are shown above each image with the citizens' consensus below.

3.4 Conclusion

The lack of spontaneous labelled data has hampered work on the machine recognition of emotional expressions. In this chapter, we have described the new emotional Labelled Faces in the Wild (eLFW) database, a citizen labelling of LFW faces. After labelling by citizens, the eLFW database comprises 190 fear images, 120 anger, 160 disgust, 330 happy, 240 neutral, 200 sad and 70 surprise images. The new data set enables us to go on to evaluate the proposed texture based emotion classification on realistic data.

Chapter 4

Combining texture features for emotion classification

Contents

4.1 Introduction 51

4.2 Image preprocessing 53

4.2.1 Face detection and alignment 54

4.2.2 Convert RGB image to greyscale. 55

4.3 Texture-feature extraction and combination 56

4.4 Baseline random forest classification 58

4.4.1 KDEF experiments 58

4.4.2 CK+ experiments 60

4.4.3 eLFW experiments 64

4.5 Feature importance mask 68

4.6 Random forest classification with importance mask . . . 70

4.6.1 Masked KDEF experiments 70

4.6.2 Masked CK+ experiments 71

4.6.3 Masked eLFW experiments 72

4.7 Comparison with citizens' classification 73

4.8 Support vector machine performance 74

4.9 Pairwise classification 77

4.9.1 Equally weighted pairwise classification 78

4.9.2 Weighted pairwise classification 83

4.10 Conclusion 87

4.1 Introduction

Facial expression recognition is a rapidly growing research topic due to an increased interest in applications of human-computer interaction. As discussed in chapter 2, it has been studied extensively over the past decade, with much of the research concentrating on geometric features. Appearance-based methods have become more prominent recently (Mishra and Dhole, 2015; Kumari et al., 2016; Yuqian and Bertram, 2016) and here we investigate the use of the combination of three feature descriptors, Histograms of Gradients (HOG) (Dalal and Triggs, 2005), Dense Speeded Up Robust Features (D-SURF) (Lowe, 2004; Uijlings et al., 2010) and Local Binary Patterns (LBP) (Ojala et al., 1996) to give more accurate classification. We show that

the combination gives a strong image descriptor. Classification with a random forest, which embodies natural feature selection, further allows us to find the face location of the most important image descriptors.

In our proposed system, there are four main steps involved in extracting and classify facial features: face detection, face alignment, facial texture feature extraction (LBP, HOG and D-SURF) and classification. We hypothesise that a combination of texture features is more effective than a single feature alone, thus yielding better classifications. In our experiments, we tested two state-of-the-art classifiers, random forest (Breiman, 2001) and support vector machines (SVM) (Cortes and Vapnik, 1995). Our proposed system has two training phases: the first one uses random forests to locate the important facial regions by estimating feature importance. The second training phase produces the final model by training with the important features only. The model automatically locates the important facial regions, which makes the classification faster and more accurate by excluding unnecessary and noisy face regions.

For evolution, three four types of measurement methods are used in this chapter, namely classification accuracy, precision, Recall and F1-score. These are widely used to evaluate the performance of classification. The accuracy is beneficial for being independent of class distribution and cost. Recall is a quality measure of completeness/quantity, which intuitively reflects the proportion of positive samples that are correctly identified. Precision refers to the percentage of your results which are relevant. F-score is a harmonic mean of precision and recall, which means the F1-score is the weighted average of precision and recall. They can distinguish the

results between classifiers when processing imbalanced data. (Zhu et al., 2018).

Image preparation and preprocessing steps are described in section 4.2. We describe the feature extraction and combination in section 4.3. Section 4.4 shows the baseline classification results for each one of the three feature extraction methods (HOG, LBP and D-SURF) separately with the two datasets (KDEF, CK+ and eLFW), and then examines the effects of combining the texture features. Section 4.5 discusses the feature size reduction by determining only the important features and image masking. After that, we show random forest classification results that apply the important feature regions. Evidence of the similarity between machine and human classification is provided in section 4.7. Facial expression classification using SVMs is discussed in section 4.8.

The difficulties encountered by people and machines when distinguishing between expressions displaying fear, anger and sadness leads us to consider alternative classifiers. In section 4.9, we describe a pairwise random forest classifier in which the pairwise classifiers have a weighted vote to determine the overall class. We show how to optimise the weights using an evolutionary algorithm and present the results showing the efficacy of the method. Finally, conclusions are drawn in section 4.10.

4.2 Image preprocessing

Image preprocessing is an important step to prepare images for feature extraction. Where the region-of-interest (ROI) is the face, we detect faces within the image to remove unwanted regions from images. Face alignment is then used to reduce the wide variety of face pose angles. Finally, all images are converted from RGB to

Figure 4.1: Four points detection(the two eyes centres and two point on the mouth).

greyscale because the image descriptors we used work with grey scale images.

4.2.1 Face detection and alignment

The first step is detecting the face. The face is detected using the Viola-Jones algorithm (Viola and Jones, 2004, 2001), which can find faces, mouths and eyes Figure 4.2 and shows an example of a detected face image after converting the RGB values to greyscale and then face detection. Having located the face, to achieve a more accurate localisation, we need to align all faces, so we need to estimate the main facial points from the centres of the eyes and at the two points on the mouth as shown in figure 4.1. Bounding boxes around the eyes and mouth were created by the Viola-Jones algorithm. We then calculate the centre points of the eyes and mouth similar to (Davison et al., 2014) these equations:

$$(Cl_x, Cl_y) = (\frac{W}{4} + x, \frac{H}{2} + y) \tag{4.1}$$

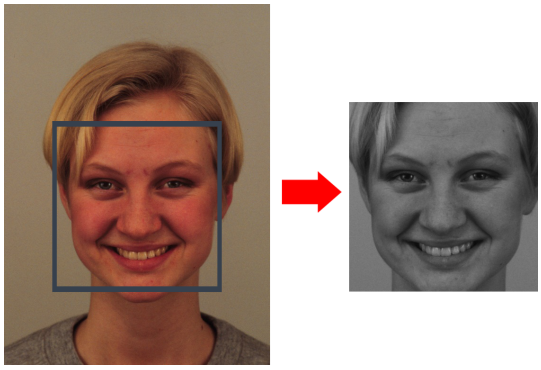


Figure 4.2: Face detection with the Viola Jones algorithm.

$$(Cr_x, Cr_y) = (\frac{3W}{4} + x, \frac{H}{2} + y) \tag{4.2}$$

where Cl is the centre of the left eye, Cr is the centre of the right eye, W is the width of the bounding box, H is the height, and x and y are the pixel locations of the top-left corner of the bounding box for the eyes. We applied the same equations 4.2 and 4.2 for the mouth (see figure 4.1). The estimated mean face points are calculated by applying Procrustes analysis (Kendall, 1989). All detected points for all image face were aligned to the mean shape by affine transformation (Hazewinkel, 2001), and each face was warped to its new aligned points, to achieve a more accurate localisation. It is important to note that all faces in the LFW dataset we used were aligned using the Huang et al. (2012) method, as shown in figure 4.3.

4.2.2 Convert RGB image to greyscale.

Our proposed method works with greyscale images because the LBP, HOG and D-SURF features were extracted from each grey scale images. All input images

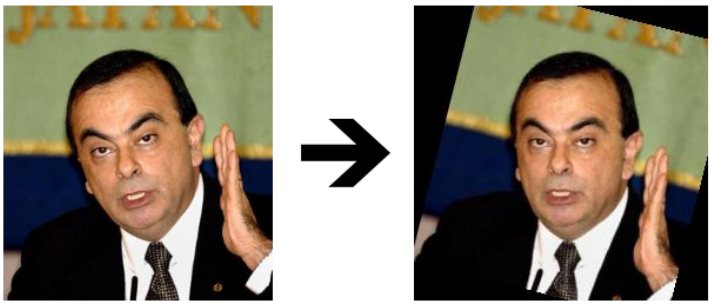


Figure 4.3: An example of facial alignment for a LFW image using the Huang et al. (2012) method

in the datasets we used are colour images. We convert RGB values to greyscale values by forming a weighted sum of the red R, green G and blue B components: $0.2989 \times R + 0.5870 \times G + 0.1140 \times B$ (Kanan and Cottrell, 2012).

4.3 Texture-feature extraction and combination

In machine learning and pattern recognition, feature extraction methods describe an image as a set of measured values called features. This data may be useful for further processing, such as in machine learning. In this thesis, we applied three commonly used image descriptors, Local Binary Pattern (LBP) (Ojala et al., 1996), Histogram of Oriented Gradients (HOG) (Dalal and Triggs, 2005) and Dense Speeded Up Robust Features (D-SURF) (Lowe, 2004; Uijlings et al., 2010), where we hypothesise that a combination of the three descriptors would give better image description than any single one. The HOG, LBP and D-SURF texture descriptors have been described in chapter 2.

As illustrated in figure 4.4, which shows an image **I** with size 400 by 400, we

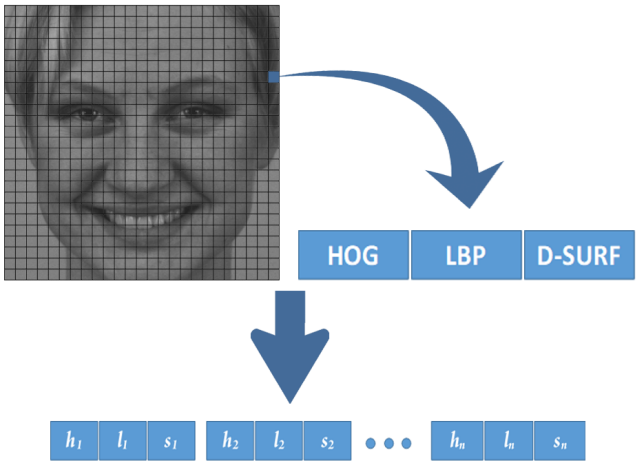


Figure 4.4: Block features (HOG, LBP and SURF) extraction and combination

divide each image into 25 by 25 non-overlapping blocks, each block size is 16 by 16 pixels, giving 625 blocks. Preliminary experiments showed that this block size of dividend gave the best accuracy and satisfying the three descriptors' constraints to be combined together. For each block the three feature descriptors: HOG $\mathbf{I_h}$, LBP $\mathbf{I_l}$ and D-SURF $\mathbf{I_s}$ are extracted and combined by concatenating each block's HOG, LBP and D-SURF descriptors in one vector $\mathbf{I_c}$, for $n = 625$ as:

$$\mathbf{I_c} = (\mathbf{I_{h_1}}, \mathbf{I_{l_1}}, \mathbf{I_{s_1}} \mathbf{I_{h_2}}, \mathbf{I_{l_2}}, \mathbf{I_{s_2}}, ..., \mathbf{I_{h_n}}, \mathbf{I_{l_n}}, \mathbf{I_{s_n}}) \tag{4.3}$$

The three features descriptors have been extracted for all the image blocks. The length of HOG for each block is 81, for LBP is 9 and for D-SURF is 64. So the length of the concatenated features is $81+64+9 = 154$ and for the image $154 \times 625 = 96250$ features in total.

4.4 Baseline random forest classification

Having a set of features describing the two datasets of images, we want to train a random forest to compare the effected of the combination of the LBP, HOG and D-SURF with using only one of them. In all our experimental evaluations, four types of performance measurements were used, the weighted F1-Score, precision, recall and accuracy. Accuracy is the number of correct predictions from all predictions made. Precision is the number of positive predictions divided by the total number of positive class values predicted. Recall is the number of positive predictions divided by the number of positive class values in the test data.F1-Score conveys the balance between the precision and the recall.

4.4.1 KDEF experiments

Initially, we tested the performance of each one of the three features (HOG, LBP and D-SURF) separately with a 5000-trees random forest classifier. 10-fold cross-validation was used to classify the 490 images in the KDEF dataset. The results are shown in tables 4.1, 4.2 and 4.3.

Table 4.1 shows the confusion matrix with only the HOG feature; the overall accuracy was 73.10%, precision is: 77.90%, recall is: 73.10% and the F1-score is: 71.40%.(true classes are shown by rows, with assigned classes in columns). Table 4.2 shows the confusion matrix result with the LBP features, where the overall accuracy was 80.10% precision is: 85.00%, recall is: 80.10% and the F1-score is: 79.10%. Finally, table 4.3 shows the D-SURF result with an overall accuracy of 70.50%;

Table 4.1: Confusion matrix for KDEF images with HOG features and a random forest classifier. The overall accuracy is 73.10%, precision is: 77.90%, recall is: 73.10% and the F1-score is: 71.40%.

	FE	**AN**	**DI**	**HA**	**NE**	**SA**	**SU**
FE	**0.300**	0.043	0.029	0.043	0.114	0.143	0.329
AN	0.000	**0.771**	0.114	0.029	0.071	0.014	0.000
DI	0.000	0.043	**0.771**	0.086	0.029	0.057	0.014
HA	0.014	0.014	0.000	**0.914**	0.057	0.000	0.00
NE	0.000	0.099	0.000	0.000	**0.843**	0.028	0.028
SA	0.043	0.100	0.043	0.014	0.229	**0.543**	0.029
SU	0.000	0.000	0.000	0.000	0.028	0.000	**0.971**

precision is: 75.30%, recall is: 70.50% and the F1-score is: 67.70%. It is noticeable that with the HOG, LBP and the D-SURF, there was a considerable misclassification with fear expression. The misclassification pattern in tables 4.1, 4.2 and 4.3 are very similar to table 3.3, and this is a clear indication of the similarity between the machine's performance and the humans.

The combined features were also tested with random forest and 10-fold cross-validation as well. Table 4.4 illustrates a visualisation of the performance. It is clear from the table that the combined model performance is better than using a single one of the three features, thus achieving an overall accuracy of 82.20%, precision is: 83.40%, recall is: 82.2% and the F1-score is: 82.3%. Figure 4.5 illustrates that the out-of-bag error decreases with the number of grown trees for the combined features with KDEF images.

Table 4.2: Confusion matrix for the KDEF images with LBP features and random forest classifier. The overall accuracy is 80.10%, precision is: 85.00%, recall is: 80.10% and the F1-score is: 79.10%.

	FE	AN	DI	HA	NE	SA	SU
FE	**0.457**	0.057	0.029	0.043	0.057	0.043	0.314
AN	0.000	**0.771**	0.114	0.029	0.071	0.000	0.014
DI	0.000	0.043	**0.914**	0.043	0.000	0.000	0.000
HA	0.000	0.000	0.014	**0.957**	0.029	0.000	0.000
NE	0.000	0.029	0.000	0.000	**0.943**	0.000	0.029
SA	0.028	0.029	0.043	0.014	0.229	**0.629**	0.029
SU	0.000	0.000	0.000	0.000	0.071	0.000	**0.929**

Table 4.4: Confusion matrix for the KDEF images with the combined features and random forest classifier. The overall accuracy is: 82.20%, precision is: 83.40%, recall is: 82.20% and the F1-score is: 82.30%.

	FE	AN	DI	HA	NE	SA	SU
FE	**0.657**	0.129	0.043	0.000	0.014	0.100	0.057
AN	0.000	**0.829**	0.100	0.000	0.000	0.057	0.014
DI	0.000	0.057	**0.871**	0.000	0.000	0.043	0.029
HA	0.014	0.014	0.014	**0.943**	0.000	0.014	0.000
NE	0.000	0.000	0.014	0.071	**0.886**	0.014	0.014
SA	0.071	0.043	0.057	0.000	0.000	**0.800**	0.029
SU	0.157	0.029	0.029	0.000	0.000	0.014	**0.771**

4.4.2 CK+ experiments

The Extended Cohn-Kanade database (CK+) (Lucey et al., 2010) is commonly used for evaluating facial expression recognition, especially with deep learning methods. It includes 593 video sequences obtained from 123 subjects labelled as 1 of 8 expressions,

Table 4.3: Confusion matrix for the KDEF photos with D-SURF features and random forest classifier. The overall accuracy is 70.50%, precision is: 75.30%, recall is: 70.50% and the F1-score is: 67.70%.

	FE	AN	DI	HA	NE	SA	SU
FE	**0.214**	0.085	0.057	0.057	0.100	0.171	0.314
AN	0.014	**0.714**	0.171	0.028	0.057	0.000	0.014
DI	0.000	0.043	**0.886**	0.057	0.000	0.014	0.000
HA	0.000	0.000	0.014	**0.957**	0.028	0.000	0.000
NE	0.014	0.071	0.000	0.000	**0.786**	0.000	0.129
SA	0.029	0.057	0.057	0.143	0.171	**0.443**	0.0100
SU	0.029	0.000	0.000	0.000	0.043	0.000	**0.929**

anger, contempt, disgust, fear, happiness, sadness, surprise and neutral. Only the last frame of each sequence is labelled. A general procedure, we use the last three frames of each sequence with the provided label, which results in 981 images. We tested the 981 CK+ images in the same way in the previous section. Tables 4.5, 4.6 and 4.7 show the results for the CK+ images with only one descriptor, HOG, LBP and D-SURF respectively. HOG achieved 67.70% overall accuracy and the F1-score 60.90%, and LBP achieved better results with 71.60% overall accuracy and 65.70% F1-Score. The combined features results as shown in table 4.8 are 78.30% for overall accuracy is, precision is: 74.80%, recall is: 75.80% and the F1-score is: 73.90%. Like the KDEF results, the D-SURF gave the lowest performance with only 60.50% overall accuracy and 54.10% F1-score. Like with the KDEF results, it is noticeable that with the HOG, LBP and the D-SURF there was a considerable misclassification with fear expression.

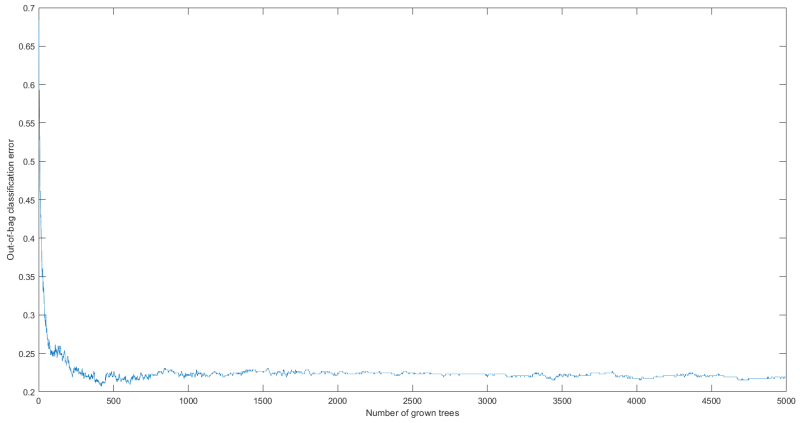


Figure 4.5: The out-of-bag error decreases with the number of grown trees for the combined features with KDEF images.

Table 4.6: Confusion matrix for the CK+ images with LBP features and random forest classifier. The overall accuracy is: 71.60%, precision is: 66.40%, recall is: 68.10% and the F1-score is: 65.70%.

	FE	AN	DI	**HA**	NE	SA	**SU**
FE	**0.333**	0.040	0.120	0.120	0.120	0.133	0.133
AN	0.030	**0.719**	0.081	0.037	0.015	0.059	0.059
DI	0.040	0.028	**0.723**	0.062	0.051	0.040	0.056
HA	0.014	0.014	0.010	**0.889**	0.029	0.019	0.024
NE	0.019	0.037	0.056	0.037	**0.796**	0.000	0.056
SA	0.048	0.048	0.048	0.060	0.119	**0.583**	0.095
SU	0.032	0.024	0.060	0.056	0.068	0.048	**0.711**

Table 4.5: Confusion matrix for the CK+ images with HOG features and random forest classifier. The overall accuracy is: 67.70%, precision is: 61.70%, recall is: 63.30% and the F1-score is: 60.90%.

	FE	AN	DI	**HA**	NE	SA	**SU**
FE	**0.267**	0.040	0.093	0.147	0.107	0.133	0.213
AN	0.000	**0.667**	0.126	0.030	0.022	0.074	0.081
DI	0.051	0.040	**0.678**	0.056	0.051	0.045	0.079
HA	0.019	0.010	0.005	**0.841**	0.068	0.019	0.039
NE	0.056	0.037	0.056	0.037	**0.704**	0.037	0.074
SA	0.071	0.048	0.083	0.036	0.119	**0.548**	0.095
SU	0.040	0.024	0.056	0.052	0.060	0.060	**0.707**

Table 4.7: Confusion matrix for the CK+ images with D-SURF features and random forest classifier. The overall accuracy is: 60.50%, precision is: 54.70%, recall is: 56.80% and the F1-score is: 54.10%.

	FE	AN	DI	**HA**	NE	SA	**SU**
FE	**0.187**	0.080	0.133	0.173	0.107	0.107	0.213
AN	0.037	**0.696**	0.104	0.067	0.030	0.030	0.037
DI	0.068	0.034	**0.650**	0.056	0.062	0.062	0.068
HA	0.029	0.029	0.039	**0.715**	0.077	0.043	0.068
NE	0.074	0.037	0.037	0.037	**0.630**	0.093	0.093
SA	0.083	0.048	0.107	0.060	0.131	**0.464**	0.107
SU	0.056	0.072	0.064	0.064	0.076	0.064	**0.602**

Table 4.8: Confusion matrix for the CK+ images with the combined features and random forest classifier. The overall accuracy is: 78.30%, precision is: 74.80%, recall is: 75.80% and the F1-score is: 73.90%.

	FE	AN	DI	**HA**	NE	SA	**SU**
FE	**0.413**	0.080	0.107	0.080	0.120	0.093	0.107
AN	0.059	**0.793**	0.007	0.052	0.030	0.015	0.044
DI	0.045	0.006	**0.802**	0.034	0.040	0.028	0.045
HA	0.014	0.019	0.005	**0.908**	0.029	0.019	0.005
NE	0.019	0.019	0.074	0.000	**0.833**	0.019	0.037
SA	0.012	0.060	0.036	0.060	0.060	**0.738**	0.036
SU	0.032	0.028	0.032	0.052	0.036	0.048	**0.771**

4.4.3 eLFW experiments

To test how the three texture features work with spontaneous rather than posed emotions, we repeated the experiments described in sections 4.4.1 and 4.4.2. As before we evaluated the performance using single feature types. 10-fold cross-validation was used with 5000 random forest trees for the 1310 eLFW faces. The results are shown in tables 4.9, 4.10 and 4.11. Table 4.9 shows the confusion matrix eLFW database, and with only HOG features; the overall accuracy was 56.9%, precision is: 52.20%, recall is: 56.30% and the F1-score is: 52.60%. Table 4.10 shows the confusion matrix result with LBP feature, where the overall accuracy was 60.9%, precision is: 55.90%, recall is: 59.20% and the F1-score is: 56.10%. Finally, table 4.11 shows the D-SURF result with an overall accuracy of 51.2%, precision is: 46.70%, recall is: 48.80% and the F1-score is: 46.50%.

Table 4.9: Confusion matrix for the eLFW images with HOG features and a random forest classifier. The overall accuracy is: 56.90%, precision is: 52.20%, recall is: 56.30% and the F1-score is: 52.60%.

	FE	**AN**	**DI**	**HA**	**NE**	**SA**	**SU**
FE	**0.137**	0.158	0.195	0.058	0.084	0.184	0.184
AN	0.092	**0.608**	0.067	0.025	0.042	0.042	0.125
DI	0.069	0.044	**0.681**	0.063	0.044	0.050	0.050
HA	0.036	0.039	0.036	**0.742**	0.070	0.036	0.048
NE	0.042	0.054	0.071	0.083	**0.638**	0.046	0.054
SA	0.150	0.130	0.085	0.030	0.035	**0.465**	0.105
SU	0.086	0.043	0.029	0.057	0.043	0.071	**0.671**

Table 4.10: Confusion matrix for the eLFW images with LBP features and random forest classifier. The overall accuracy is: 61.00%, precision is: 55.90%, recall is: 59.20% and the F1-score is: 56.10%.

	FE	**AN**	**DI**	**HA**	**NE**	**SA**	**SU**
FE	**0.184**	0.153	0.195	0.047	0.074	0.179	0.168
AN	0.092	**0.650**	0.058	0.017	0.033	0.033	0.117
DI	0.069	0.044	**0.719**	0.050	0.038	0.038	0.044
HA	0.027	0.033	0.036	**0.773**	0.045	0.036	0.048
NE	0.025	0.033	0.050	0.079	**0.729**	0.042	0.042
SA	0.150	0.125	0.075	0.020	0.025	**0.500**	0.105
SU	0.114	0.071	0.057	0.057	0.043	0.071	**0.586**

Table 4.11: Confusion matrix for the eLFW images with D-SURF features and random forest classifier. The overall accuracy is: 51.20%, precision is: 46.70%, recall is: 48.80% and the F1-score is: 46.50%.

	FE	**AN**	**DI**	**HA**	**NE**	**SA**	**SU**
FE	**0.168**	0.153	0.195	0.058	0.079	0.179	0.168
AN	0.100	**0.508**	0.067	0.092	0.050	0.058	0.125
DI	0.081	0.056	**0.656**	0.075	0.050	0.038	0.044
HA	0.042	0.039	0.045	**0.645**	0.100	0.052	0.076
NE	0.042	0.054	0.058	0.079	**0.650**	0.063	0.054
SA	0.155	0.145	0.085	0.060	0.040	**0.375**	0.140
SU	0.114	0.086	0.057	0.114	0.100	0.114	**0.414**

Table 4.12: Confusion matrix for the eLFW images with the combined features and random forest classifier. The overall accuracy is: 67.30%, precision is: 59.80%, recall is: 61.40% and the F1-score is: 59.60%.

	FE	**AN**	**DI**	**HA**	**NE**	**SA**	**SU**
FE	**0.232**	0.153	0.184	0.026	0.079	0.168	0.158
AN	0.092	**0.650**	0.058	0.025	0.017	0.042	0.117
DI	0.063	0.044	**0.738**	0.050	0.038	0.038	0.031
HA	0.012	0.006	0.021	**0.927**	0.024	0.003	0.006
NE	0.008	0.008	0.021	0.054	**0.879**	0.017	0.013
SA	0.150	0.135	0.075	0.025	0.020	**0.490**	0.105
SU	0.157	0.143	0.171	0.043	0.014	0.086	**0.386**

It is clear from this section that with LBP, HOG and D-SURF, it was hard to classify the fear expression whereas happy and neutral are more easily classified in both datasets and fear is most often misclassified. The combination of the three texture features improved the overall accuracy with both datasets. From table 4.12, it is clear that the combined model performance is better than using a single one of

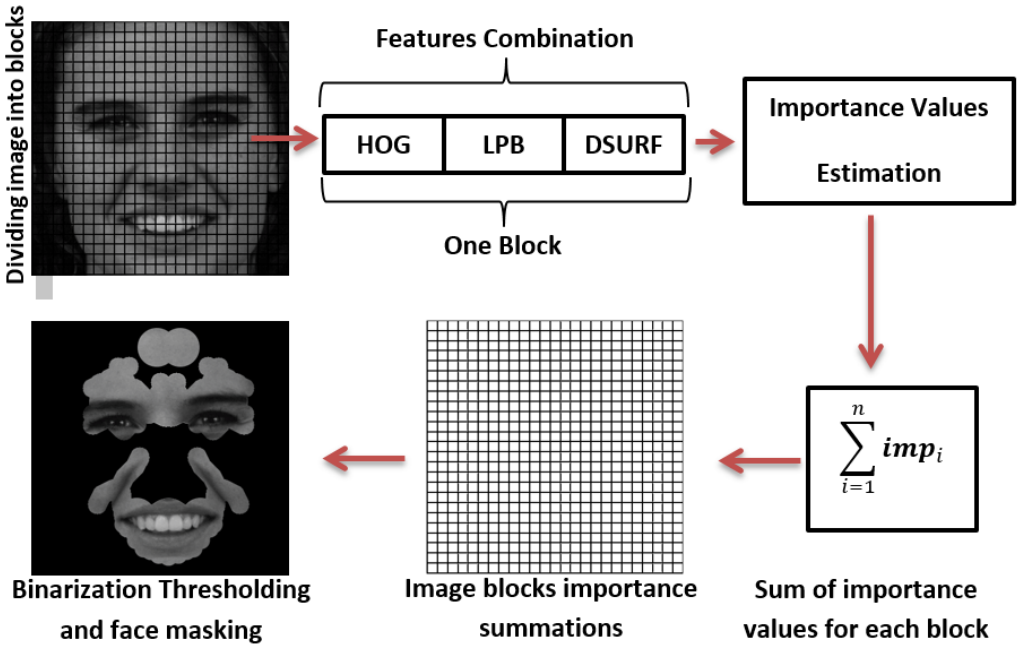


Figure 4.6: Schematic of steps used for determining the feature importance mask. This figure shows the steps of finding the importance mask by applying the Otsu method to the total of the estimated importance values.

the three features, achieving an overall accuracy of 67.3% precision is: 59.80%, recall is: 61.40% and the F1-score is: 59.60%.

The overall accuracy for eLFW is lower than CK+ and KDEF, but the pattern for the three types of misclassification is similar, so it is clear that the fear emotion, for instance, was the lowest accuracy where the happy emotion was the highest for both datasets.

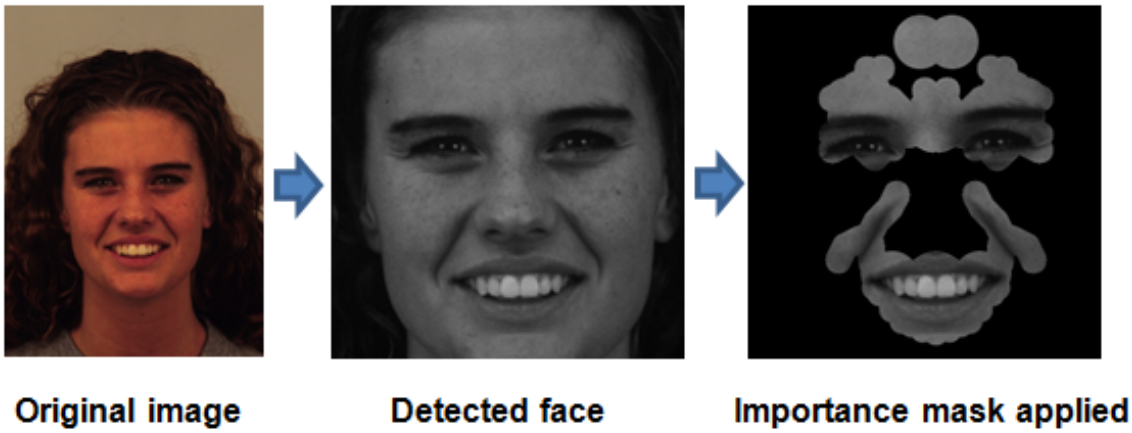


Figure 4.7: Importance masking steps for a KDEF image.

4.5 Feature importance mask

After the three types of features were extracted and combined, the random forest was trained by the features of the images to estimate the predictor importance for each value of the combined feature vector. Each image block is matched matched with its part of the combined vectors to decide where the important facial parts are. To locate the important face part, we sum the importance values for each face block to yield a matrix of 25 by 25 combined importance values. We applied the Otsu method (Otsu, 1975) to covert the importance values to binary. A mask of size 25 by 25 was produced. After getting the mask, by bicubic interpolation, we resized it to 400 by 400 to be the same as the size of the images. Figures 4.6 and 4.7 illustrate the main steps of mask creation.

Figure 4.8 illustrates a comparison between the random forest importance prediction values. Red colours show that the HOG features were the most important in

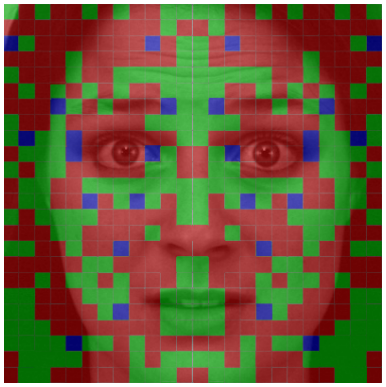
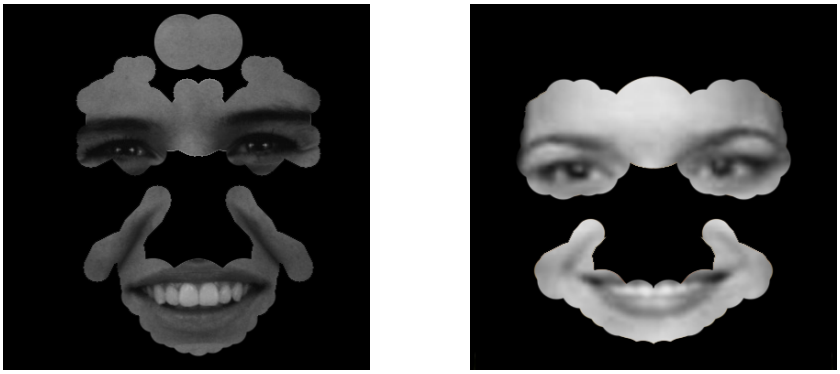


Figure 4.8: HOG, LBP and SURF importance values comparison. Red, blue and green indicate which feature type (HOG, LBP and D-SURF respectively) was most important for each block. We use half of the face to make the mask symmetric.

the location, while green colour refers to LBP and blue to D-SURF. It is clear that HOG and LBP occupy most of the facial region compared to D-SURF. As Figure 4.9 shows, this procedure identifies the eyes, mouth, the creases at the side of the nose and the forehead as the most informative. We point out that this is an empirically determined mask rather than one chosen a priori. We obtain slightly different masks for the acted and wild faces. It is well known that the human faces are symmetrical, so we made a horizontal-reverse the right half of the face mask, to ensure that we deal with both of face sides in the same way.

Importance masks are slightly different according to the training data images, so when we trained the system with the KDEF images, the mask was as shown in figure 4.7. The eLFW training produced a mask with some differences, as shown in figure 4.9.



(a) KDEF mask (b) eLFW mask

Figure 4.9: Mask identifying the most important regions of the face for emotion classification derived by binariesing feature importance mask.

4.6 Random forest classification with importance mask

After obtaining the masks from the two datasets KDEF and eLFW, we applied each mask to all images in the relevant dataset to remove the unwanted parts and to keep only those that were important. The resulting features were then classified using random forest (5000 trees). 10-fold cross-validation was used to evaluate the classification scheme.

4.6.1 Masked KDEF experiments

Table 4.13 shows the confusion matrix for masked KDEF images with the combined features and random forest classifier. The overall accuracy is 89.80%; precision is: 90.80%, recall is: 89.80% and the F1-score is: 89.70%. The happy expression had the highest classification rate, while the fear expression has the lowest. With these

Table 4.13: Confusion matrix for the masked the KDEF images with the combined features and random forest classifier. The overall accuracy is: 89.80%, precision is: 90.80%, recall is: 89.80% and the F1-score is: 89.70%.

	FE	AN	DI	HA	NE	SA	SU
FE	**0.671**	0.143	0.043	0.000	0.000	0.143	0.000
AN	0.000	**0.857**	0.071	0.000	0.000	0.071	0.000
DI	0.000	0.043	**0.929**	0.000	0.000	0.029	0.000
HA	0.000	0.000	0.000	**0.986**	0.014	0.000	0.000
NE	0.000	0.000	0.000	0.014	**0.986**	0.000	0.000
SA	0.043	0.029	0.043	0.000	0.000	**0.886**	0.000
SU	0.014	0.000	0.014	0.000	0.000	0.000	**0.971**

two expressions, there is no significant improvement between masking and without. Neutral and surprise rose significantly by applying the mask, 88.6% to 98.6% for neutral, and 77.1% to 97.1% for the surprise expression.

4.6.2 Masked CK+ experiments

Table 4.13 shows the confusion matrix for the masked KDEF images with the combined features and random forest classifier. The overall accuracy is 82.20%; precision is: 79.10%, recall is: 81.30% and the F1-score is: 79.20%. Similar the achievements with KDEF, the happy expression had the highest classification rate, while the fear expression has the lowest. With these two expressions, there is no significant improvement between masking and without.

Table 4.14: Confusion matrix for the masked CK+ images with the combined features and random forest classifier. The overall accuracy is: 82.20%, precision is: 79.10%, recall is: 81.30% and the F1-score is: 79.20%.

	FE	AN	DI	**HA**	NE	SA	**SU**
FE	**0.667**	0.053	0.027	0.027	0.053	0.067	0.107
AN	0.059	**0.748**	0.037	0.044	0.037	0.022	0.052
DI	0.040	0.006	**0.825**	0.040	0.056	0.011	0.023
HA	0.014	0.014	0.005	**0.913**	0.014	0.029	0.010
NE	0.000	0.037	0.019	0.000	**0.889**	0.019	0.037
SA	0.024	0.095	0.024	0.012	0.012	**0.798**	0.036
SU	0.020	0.024	0.020	0.024	0.036	0.056	**0.819**

4.6.3 Masked eLFW experiments

Table 4.15 shows the confusion matrix for the masked eLFW images. It is clear that applying the mask increases the accuracy from 67.3% without masking to 71.6% with masking (see table 4.12). As more results i table 4.15 the precision is: 64.10%, recall is: 66.00% and the F1-score is: 64.00%. The happy and neutral expressions are most easily classified in both datasets and fear is most often misclassified, particularly in the eLFW data, where it is confused with all the other classes except happy and neutral.

Table 4.16 shows a confusion matrix for testing the eLFW database after training our proposed system with the KDEF database. KDEF contains 70 faces for each expression. We chose 70 random faces from the eLFW for testing. It is clear from the table that gives good results for most of the facial expressions. The overall accuracy was 74.7%.

Table 4.15: Confusion matrix for the random forest classification of the masked eLFW databases. The overall accuracy was 71.60%, precision is: 64.10%, recall is: 66.00% and the F1-score is: 64.00%.

	FE	**AN**	**DI**	**HA**	**NE**	**SA**	**SU**
FE	**0.263**	0.153	0.200	0.016	0.042	0.168	0.158
AN	0.083	**0.725**	0.042	0.000	0.000	0.042	0.108
DI	0.063	0.031	**0.775**	0.038	0.031	0.031	0.031
HA	0.000	0.006	0.009	**0.973**	0.012	0.000	0.000
NE	0.004	0.004	0.017	0.054	**0.908**	0.008	0.004
SA	0.150	0.130	0.065	0.010	0.010	**0.535**	0.100
SU	0.157	0.129	0.157	0.029	0.000	0.086	**0.443**

4.7 Comparison with citizens' classification

As mentioned in chapter 3, to avoid ambiguous classifications by the annotators, we calculate the entropy of the empirical distribution of classifications. The entropy, $H = -\sum_n p_n \log_2 p_n$, measures the agreement between the annotators, so the entropy is maximised when all classes are assigned in equal proportion and it is minimised when images are assigned to only a single class.

The patterns of misclassification for RF and the annotators are similar. Table 4.17 shows the average entropy of the distributions of the citizens' votes for images that were correctly and incorrectly classified. For each of the emotion classes, the average entropy for the misclassified images is greater than or equal to the average entropy of the correctly classified images displaying the same emotion. This indicates that there was more disagreement about the emotion displayed that there was about correctly classified images.

Table 4.16: Confusion matrix on the eLFW database using proposed method, trained with KDEF. Overall accuracy was 74.7%, precision is: 74.29%, recall is: 74.7% and the F1-score is: 74.01%.

	FE	**AN**	**DI**	**HA**	**NE**	**SA**	**SU**
FE	**0.371**	0.086	0.114	0.043	0.043	0.071	0.271
AN	0.057	**0.643**	0.157	0.000	0.014	0.057	0.071
DI	0.43	0.029	**0.743**	0.000	0.071	0.014	0.100
HA	0.029	0.014	0.000	**0.929**	0.014	0.014	0.000
NE	0.014	0.014	0.014	0.029	**0.900**	0.014	0.014
SA	0.029	0.014	0.043	0.000	0.029	**0.871**	0.014
SU	0.157	0.014	0.043	0.000	0.000	0.014	**0.771**

It is evident in table 4.17 that the correctly classified happy-labelled images are lower than those has been missed classified. That means the lowest entropy images were easier for both machine and annotators and vice versa, so this can be determined as a consensus indicator between machine and annotators. From the same table, we can notice that the images were labelled happy and natural were the easiest voting by annotators. Fear images were the hardest for machine and annotators. The missed classified images are 0.981 and those were correctly classified were 0.962.

4.8 Support vector machine performance

Random forest and SVMs have been used extensively for classification tasks recently Datta et al. (2017); Munasinghe (2018), so in this section, we compare the performance of SVMs with the performance of random forests. We used 1-vs-all SVM with a radial bias function (RBF) kernel. There are two parameters that affect the SVM performance. The first is the regularisation parameter C, which controls the

Table 4.17: Average entropy of citizens' voting distributions for correctly and incorrectly classified the eLFW images.

	Entropy (bits)		
Emotion	Correct		Misclassified
Fear	0.962	$<$	0.981
Anger	0.956	$<$	0.982
Disgust	0.983	$=$	0.983
Happy	0.777	$<$	0.902
Neutral	0.869	$<$	0.943
Sad	0.956	$<$	0.963
Surprised	0.937	$<$	0.976
Overall	0.920	$<$	0.961

trade-off between achieving a low error on the training data and minimising the norm of the weights. The second parameter is γ, which can be seen as the inverse of the radius of influence of samples selected by the model as support vectors. With low γ, the curvature of the decision boundary is small, and thus, the decision region is wide. When γ is large, the curvature of the decision boundary is high, which creates islands of decision-boundaries around data points. To optimise the SVM, we need to find the best SVM parameters, γ and C. We used Bayesian optimisation, which is a powerful method to optimise functions (Mockus, 2012; Brochu et al., 2010). Figure 4.10 shows the surface plot of the Bayesian optimisation search for the error function minimisation, This figure illustrates the estimated location of the minimum error,

Table 4.18: Confusion matrices for the SVM classification of the KDEF database.

Without the mask. Accuracy 72%, precision is: 73.90%, recall is: 72.4% and the F1-score is: 72.40%.

	FE	AN	DI	HA	NE	SA	SU
FE	**0.643**	0.014	0.057	0.100	0.029	0.071	0.086
AN	0.043	**0.671**	0.071	0.000	0.043	0.086	0.086
DI	0.086	0.057	**0.714**	0.000	0.000	0.071	0.071
HA	0.000	0.000	0.000	**0.900**	0.057	0.014	0.029
NE	0.000	0.000	0.014	0.071	**0.843**	0.057	0.014
SA	0.068	0.043	0.071	0.000	0.029	**0.657**	0.114
SU	0.157	0.057	0.043	0.014	0.014	0.071	**0.643**

With the mask.Accuracy 81.00%, precision is: 82.00%, recall is: 81.00% and the F1-score is: 80.70%.

	FE	AN	DI	HA	NE	SA	SU
FE	**0.714**	0.057	0.043	0.000	0.000	0.100	0.086
AN	0.043	**0.686**	0.071	0.000	0.029	0.086	0.086
DI	0.043	0.057	**0.786**	0.000	0.000	0.057	0.057
HA	0.000	0.000	0.000	**0.957**	0.029	0.000	0.014
NE	0.000	0.000	0.000	0.057	**0.900**	0.029	0.014
SA	0.057	0.057	0.043	0.000	0.000	**0.771**	0.071
SU	0.071	0.029	0.014	0.014	0.000	0.029	**0.843**

and the location of the next proposed point to evaluate. An example of a Bayesian optimisation search. We applied 10-fold cross-validation, so the optimisation aims to minimise the loss function of the partitioned classification model, which measure the predictive inaccuracy of classification models.

This cross-validation can be computationally expensive but it has a significant advantage by not wasting too much data as when fixing an arbitrary test set (Chih-Wei Hsu, 2003).

Table 4.18 (left) illustrates confusion matrices for the KDEF photos with SVMs for whole facial features without applying the importance mask, the overall accuracy was 72%. After applying the KDEF mask, the SVM overall classification rate rose to 81% as shown in table 4.18 (right). The optimum parameters were $\gamma \approx 4.3 \times 10^4$ and $C \approx 10^{19}$, indicating that the classifier is quite non-linear.

By comparing the result presented in table 4.18 (left) with those shown in 4.4

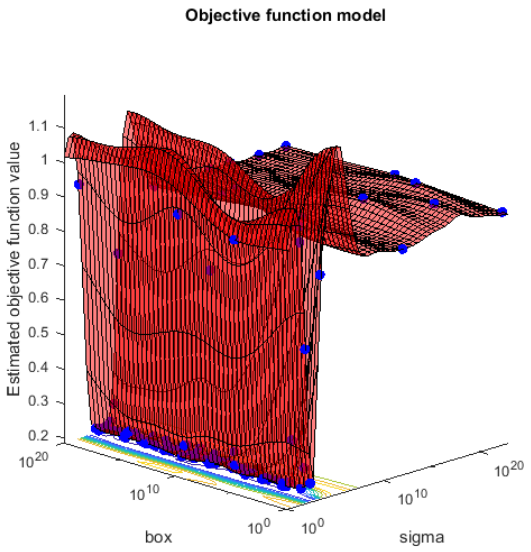


Figure 4.10: Min objective vs number of function evaluations

and 4.13, we can see that the random forest gives better results for this classification problem.

Table 4.19 illustrates a confusion matrix of the masked eLFW by SVMs. Compared to the results in table 4.12, the random forest still gives slightly better accuracy compared to the SVM with the importance masked combined features. According to table 4.19, the overall classification accuracy for SVMs was 66.3% where the overall classification in the table 4.12 was 67.3%.

4.9 Pairwise classification

Random forest classifiers combine decision trees which are naturally capable of multi-class classification, as opposed to dichotomous classifiers, such as SVMs, for which

Table 4.19: Confusion matrix for the SVM classification of the eLFW database with the importance mask shown in Figure 4.7. Accuracy 66.3%, precision is: 61.30%, recall is: 63.3% and the F1-score is: 60.60%.

	FE	AN	DI	HA	NE	SA	SU
FE	**0.237**	0.137	0.184	0.042	0.058	0.153	0.189
AN	0.058	**0.733**	0.025	0.000	0.008	0.075	0.100
DI	0.056	0.025	**0.750**	0.025	0.038	0.025	0.081
HA	0.018	0.021	0.15	**0.845**	0.033	0.018	0.048
NE	0.025	0.013	0.038	0.063	**0.829**	0.021	0.013
SA	0.175	0.125	0.060	0.010	0.025	**0.495**	0.110
SU	0.157	0.114	0.143	0.029	0.000	0.014	**0.543**

strategies such as one-versus-all or pairwise voting must be employed. In an effort to reduce the misclassifications of fear, anger, disgust and surprise, we also investigated the use of pairwise classifiers for classifying emotions. In this framework, a single classifier is trained to discriminate between a pair of classes. The item is then assigned to the class which receives the majority of votes from the pairwise classifiers.

4.9.1 Equally weighted pairwise classification

For n-Class pairwise classification, the number of the classifiers is $c = \frac{n(n-1)}{2}$, so to classify the $n = 7$ emotions, there are 21 pairs of classifiers. Each pair of facial expression classes was used to train one random forest classifier. At the testing stage, each image was tested with all 21 classifiers. For each facial expression, we calculate all votes from each classifier to get the total vote. The maximum vote for the seven expressions is the final decision. As example, table 4.20 shows the posterior probabilities of 21 random forest classifiers for two different images whose

Table 4.20: Posterior probabilities for two different fear images tested by the 21 random classifiers, the left has been wrongly classified and the right correctly classified.

Wrong classification decision

	Fear	Anger	Disgust	Happy	Neutral	Sad	Surprise	Total
Fear	x	0.87	0.88	0.85	0.49	0.62	0.38	**4.10**
Anger	0.13	x	0.80	0.73	0.12	0.22	0.12	**2.11**
Disgust	0.12	0.21	x	0.56	0.08	0.10	0.07	**1.13**
Happy	0.15	0.27	0.44	x	0.07	0.11	0.08	**1.12**
Neutral	0.51	0.88	0.92	0.93	x	0.64	0.48	**4.36**
Sad	0.38	0.78	0.90	0.89	0.36	x	0.31	**3.63**
Surprise	0.62	0.88	0.93	0.92	0.51	0.69	x	**4.55**

Correct classification decision

	Fear	Anger	Disgust	Happy	Neutral	Sad	Surprise	Total
Fear	x	0.94	0.93	0.86	0.85	0.69	0.48	**4.76**
Anger	0.06	x	0.78	0.77	0.27	0.16	0.10	**2.13**
Disgust	0.07	0.23	x	0.57	0.16	0.12	0.12	**1.27**
Happy	0.14	0.23	0.43	x	0.18	0.28	0.13	**1.40**
Neutral	0.15	0.73	0.84	0.81	x	0.19	0.31	**3.04**
Sad	0.31	0.84	0.88	0.72	0.81	x	0.31	**3.86**
Surprise	0.52	0.90	0.88	0.86	0.69	0.70	x	**4.54**

correct class was fear. Each cell in the tables is the posterior probability (classifier score) with the expression in the same row vs the expression in the same column. In the left table, the image has been wrongly classified as surprise (maximum total 4.55). Table 4.20 (right) shows another fear image which has been correctly classified (maximum total 4.76).

Table 4.21 shows the accuracy of the RF and SVM classifiers on the KDEF data, using the usual 10-fold cross-validation. We notice that the performance of both classifiers is very similar, and that there is no significant difference between the overall scores: the RF's average accuracy was 0.963 and SVMs was 0.962. There is one advantage of SVMs: all the 21 classifiers gave an accuracy of over 90%, whereas RF gave two results lower than 90% with fear vs surprise, and neutral vs sad.

Figure 4.11 and 4.12 show two examples of the out-of-bag error decreases with the number of grown trees. We can see how is the classification between happy and surprise gives excellent accuracy, and that we do not need to build lots of trees to get high accuracy. Fear and surprise are the most difficult to classify, but the random forest achieved 88.3% accuracy, and this is an excellent result.

Table 4.21: Pairwise RF classifiers and pairwise SVM classifiers performance with KDEF.

Classifier	**RF**	**SVM**	**Classifier**	**RF**	**SVM**
Fear vs Anger	0.956	0.935	Disgust vs Happy	0.976	0.978
Fear vs Disgust	0.968	0.978	Disgust vs Neutral	0.999	0.993
Fear vs Happy	0.976	0.978	Disgust vs Sad	0.960	0.942
Fear vs Neutral	0.940	0.957	Disgust vs Surprise	0.988	1.000
Fear vs Sad	0.920	0.900	Happy vs Neutral	0.999	0.993
Fear vs Surprise	0.883	0.900	Happy vs Sad	0.987	0.971
Anger vs Disgust	0.935	0.900	Happy vs Surprise	0.999	0.993
Anger vs Happy	0.993	0.985	Neutral vs Sad	0.895	0.940
Anger vs Neutral	0.952	0.950	Neutral vs Surprise	0.984	0.985
Anger vs Sad	0.936	0.950	Sad vs Surprise	0.976	0.978
Anger vs Surprise	0.994	1.000	**Overall**	**0.963**	**0.962**

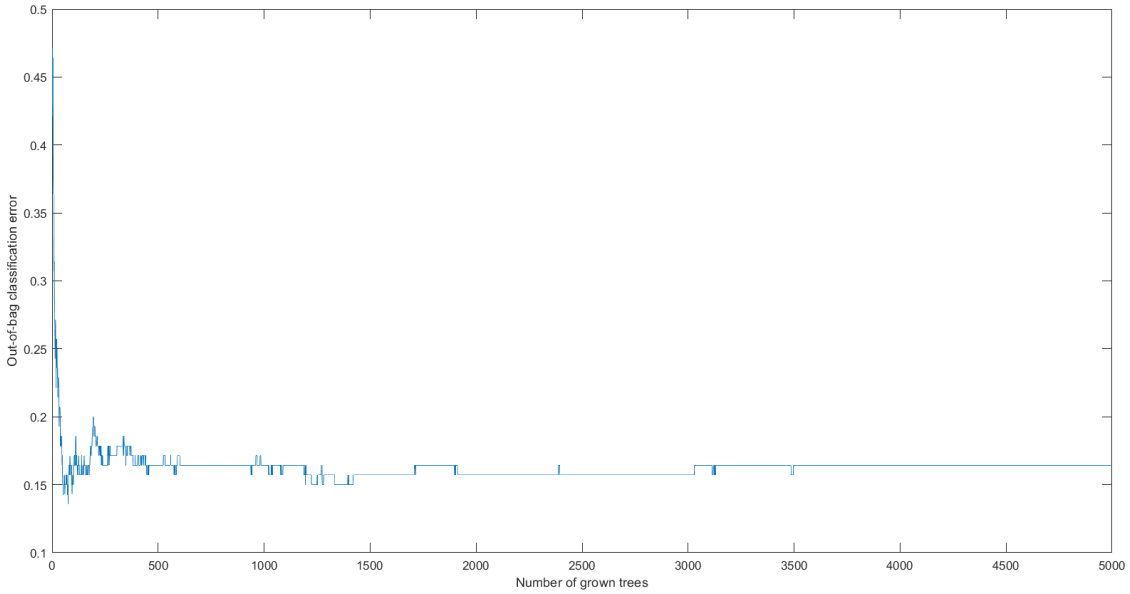


Figure 4.12: The out-of-bag error decreases with the number of grown trees fear vs surprise.

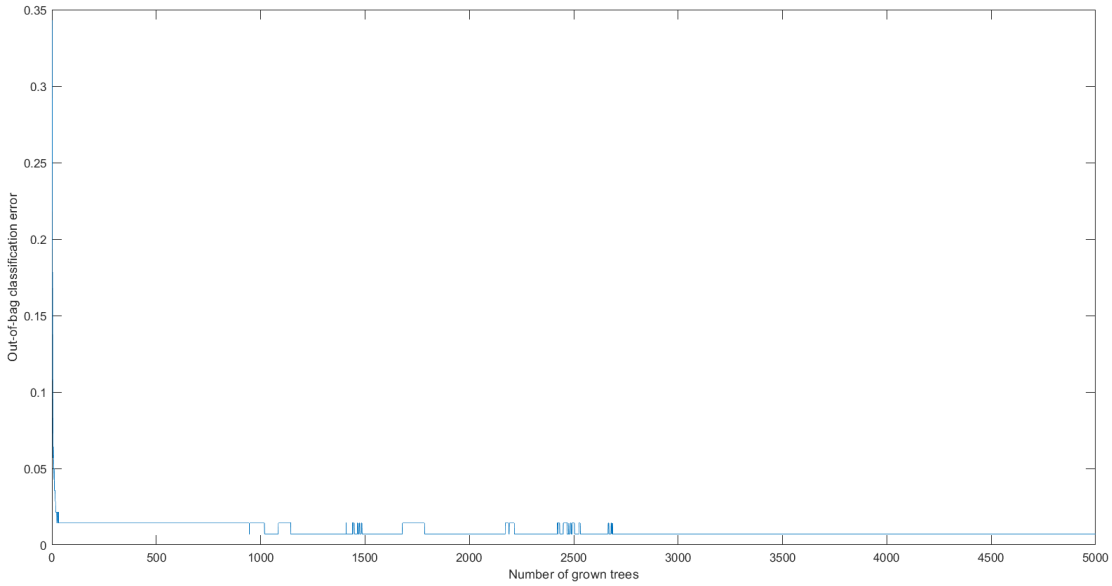


Figure 4.11: The out-of-bag error decreases with the number of grown trees for happy vs surprise.

Table 4.23: Confusion matrices for equally pairwise classification of the CK+ database. The overall accuracy is 89.40%, precision is: 88.20%, recall is: 88.80% and the F1-score is: 88.20%.

	FE	**AN**	**DI**	**HA**	**NE**	**SA**	**SU**
FE	**0.773**	0.040	0.027	0.000	0.000	0.027	0.133
AN	0.022	**0.822**	0.089	0.007	0.000	0.007	0.052
DI	0.023	0.023	**0.898**	0.017	0.000	0.006	0.034
HA	0.010	0.014	0.000	**0.937**	0.019	0.014	0.005
NE	0.000	0.019	0.000	0.019	**0.926**	0.000	0.037
SA	0.000	0.048	0.024	0.012	0.000	**0.905**	0.012
SU	0.040	0.004	0.004	0.008	0.016	0.008	**0.920**

Table 4.22 and 4.23 show the RF pairwise classification results of the KDEF,

Table 4.22: Confusion matrices for equally pairwise classification of the KDEF and eLFW databases. The overall accuracy is 92.2%., precision is: 93.20%, recall is: 92.2% and the F1-score is: 92.10% for KDEF. The overall accuracy is 73.3%, precision is: 67.40%, recall is: 68.7% and the F1-score is: 66.9% for eLFW.

KDEF

	FE	**AN**	**DI**	**HA**	**NE**	**SA**	**SU**
FE	**0.814**	0.029	0.043	0.000	0.000	0.029	0.086
AN	0.000	**0.886**	0.043	0.000	0.000	0.071	0.000
DI	0.000	0.043	**0.929**	0.000	0.000	0.029	0.000
HA	0.000	0.000	0.000	**0.986**	0.014	0.000	0.000
NE	0.000	0.000	0.000	0.014	**0.986**	0.000	0.000
Sad	0.029	0.029	0.043	0.000	0.000	**0.900**	0.000
SU	0.029	0.000	0.014	0.000	0.000	0.000	**0.957**

eLFW

	FE	**AN**	**DI**	**HA**	**NE**	**SA**	**SU**
FE	**0.463**	0.095	0.142	0.016	0.042	0.079	0.163
AN	0.033	**0.750**	0.058	0.025	0.000	0.067	0.067
DI	0.056	0.019	**0.781**	0.044	0.031	0.025	0.044
HA	0.003	0.027	0.006	**0.927**	0.018	0.009	0.009
NE	0.008	0.017	0.013	0.017	**0.917**	0.029	0.000
SA	0.165	0.120	0.065	0.030	0.025	**0.485**	0.110
SU	0.100	0.071	0.086	0.029	0.114	0.114	**0.486**

eLFW and CK+ data respectively. As we can see in the tables, the overall accuracy is 92.2%., precision is: 93.20%, recall is: 92.2% and the F1-score is: 92.10% for KDEF. The overall accuracy is 73.3%, precision is: 67.40%, recall is: 68.7% and the F1-score is: 66.9% for eLFW. The overall accuracy is 89.40%, precision is: 88.20%, recall is: 88.80% and the F1-score is: 88.20%. By comparing these results with the results in table 4.13, 4.14 and 4.15, it is clear that pairwise classification has enhanced the overall accuracy from 89% to be 92.2% for KDEF, and from 71.7% to 73.3% for eLFW, and from 82.20% to 89.40% for CK+. We note that the fear expression classification rate has been significantly improved.

Table 4.24: Random forest pair-classifiers optimised weights for 7 classes

	Fear	**Anger**	**Disgust**	**Happy**	**Neutral**	**Sad**	**Surprise**
Fear	x	0.198	0.202	0.159	0.106	0.159	0.176
Anger	0.226	x	0.247	0.232	0.045	0.127	0.122
Disgust	0.205	0.187	x	0.175	0.114	0.155	0.164
Happy	0.210	0.187	0.184	x	0.114	0.150	0.156
Neutral	0.216	0.187	0.174	0.107	x	0.157	0.159
Sad	0.206	0.195	0.172	0.106	0.158	x	0.164
Surprise	0.204	0.190	0.171	0.116	0.151	0.169	x

4.9.2 Weighted pairwise classification

In most pairwise architectures each of the constituent classifiers has an equal vote. Here we weigh the votes from each classifier and learn appropriate weights by optimising the classification accuracy of a validation set.

More specifically, suppose $y_{ij}(\mathbf{x}_n) \in [0,1]$ is the output of the classifier discriminating between classes i and j for image features $\mathbf{x}_n$. Here we use random forest for each dichotomous classifier, so that $y_{ij}(\mathbf{x}_n)$ is the proportion of decision trees in the (i,j)-th forest that voted for class i. Then the overall score for class i is

$$Y_i(\mathbf{x}_n) = \sum_{j \neq i} \lambda_{ij} y_{ij}(\mathbf{x}_n) \tag{4.4}$$

where the weights are λ_{ij}, and the image is assigned to the class with the largest overall score: $\operatorname{argmax}_i Y_i(\mathbf{x}_n)$. The weights are constrained to be non-negative, $\lambda_{ij} \geq 0$ for all i and j, and we demand that $\sum_j \lambda_{ij} = 1$ for all i.

Training takes place in two phases. First, the constituent classifiers are independently trained on the pairs of classes. Secondly, the accuracy of the overall classifier

Table 4.25: Confusion matrices for the weighted pairwise classification of the KDEF and eLFW databases. The overall accuracy is 95.1%, precision is: 95.70%, recall is: 95.1% and the F1-score is: 95.0% for KDEF. The overall accuracy is 76.6%, precision is: 71.10%, recall is: 72.10% and the F1-score is: 70.6% for eLFW.

KDEF

	FE	**AN**	**DI**	**HA**	**NE**	**SA**	**SU**
FE	**0.914**	0.029	0.000	0.000	0.000	0.000	0.057
AN	0.000	**0.943**	0.014	0.000	0.000	0.043	0.000
DI	0.000	0.014	**0.971**	0.000	0.000	0.014	0.000
HA	0.000	0.000	0.000	**0.986**	0.014	0.000	0.000
NE	0.000	0.000	0.000	0.014	**0.986**	0.000	0.000
SA	0.043	0.029	0.043	0.000	0.000	**0.886**	0.000
SU	0.014	0.000	0.014	0.000	0.000	0.000	**0.971**

eLFW

	FE	**AN**	**DI**	**HA**	**NE**	**SA**	**SU**
FE	**0.495**	0.089	0.142	0.011	0.042	0.074	0.147
AN	0.025	**0.792**	0.050	0.000	0.000	0.083	0.050
DI	0.056	0.019	**0.788**	0.044	0.031	0.025	0.038
HA	0.003	0.012	0.003	**0.970**	0.009	0.003	0.000
NE	0.004	0.013	0.000	0.017	**0.938**	0.029	0.000
SA	0.150	0.130	0.065	0.010	0.010	**0.535**	0.100
SU	0.086	0.086	0.071	0.014	0.129	0.086	**0.529**

on a second training data set is maximised by optimising the voting weights using an evolutionary optimiser. Here we used CMA-ES (Hansen, 2006), a popular and effective evolutionary optimiser. Constraints were enforced by working in terms of variables θ_{ij} with

$$\lambda_{ij} = \frac{\theta_{ij}}{\sum_k \theta_{ik}}. \tag{4.5}$$

We note that the procedure is efficient because once the pairwise classifiers have been trained, classification scores $y_{ij}(\mathbf{x}_n)$ need only be calculated once before optimisation of the weights.

Table 4.26: Confusion matrices for weighted pairwise classification of the CK+ database. The overall accuracy is 91.30%, precision is: 90.70%, recall is: 89.40% and the F1-score is: 89.60%.

	FE	**AN**	**DI**	**HA**	**NE**	**SA**	**SU**
FE	**0.733**	0.000	0.027	0.013	0.000	0.013	0.213
AN	0.037	**0.896**	0.030	0.007	0.000	0.000	0.030
DI	0.017	0.011	**0.932**	0.006	0.000	0.006	0.028
HA	0.000	0.000	0.000	**0.976**	0.024	0.000	0.000
NE	0.000	0.019	0.000	0.019	**0.926**	0.000	0.037
SA	0.024	0.048	0.012	0.036	0.000	**0.810**	0.071
SU	0.040	0.004	0.004	0.004	0.004	0.004	**0.940**

Table 4.24 shows the weights λ_{ij} obtained by the optimisation processes. This weights can show which pairwise-classier is more important to make a decision than others. In the first raw the fear-vs-disgust classifier has the largest weight, where fear-vs-neutral is the lowest. Moreover, in the disgust raw the fear-vs-disgust was the most important.

Table 4.25 shows the confusion matrix obtained with the optimised pairwise classification using 10-fold cross-validation testing. The accuracies for the data sets have increased to 95.1% for KDEF and 76.6% for eLFW. As can be seen from the confusion matrix, classification accuracies of fear, anger, surprise and disgust have increased substantially, although for the eLFW data there is still considerable misclassification of fear (confused with disgust and surprise), sadness (confused with fear and anger) and surprise (confused with neutral, fear, anger and disgust). We remark that these emotions are all often expressed through a grimacing expression which may account for the difficulty in distinguishing them. Furthermore, these are the emotions about

which the citizens showed the most disagreement (see section 4.7).

Table 4.27: Comparison of classification accuracy of random forest classification using masked LBP, HOG and D-SURF texture features with other recent techniques. Evaluation on the KDEF and CK+ databases.

Method	Database	Accuracy
SURF with AdaBoost Rao et al. (2015)	KDEF	74.05%
LDBP with SVM Santra and Mukherjee (2016a)	KDEF	83.51%
LSiBP with SVM Santra and Mukherjee (2016b)	KDEF	84.07%
Our proposed method	KDEF	95.10%
CNN Meng et al. (2017)	CK+	95.37%
CNN Cai et al. (2018)	CK+	94.39%
GAN (cGAN) Yang et al. (2018)	CK+	97.30%
CNN Zhang et al. (2018) *(6 classes)*	CK+	98.00%
Our proposed method	CK+	91.30%

Table 4.27 shows a comparison between our proposed system with some of recent works of the state-of-the-art methods. It is seen by comparing our proposed method with some of the recent works Rao et al. (2015); Santra and Mukherjee (2016a,b) which used images descriptors; our method gave the highest overall accuracy 95.10% with KDEF database. In the table, we show some of recent works that have used the KDEF database, with image descriptors such as SURF, LDBP, LSiBP. In the last few years, deep learning algorithms have seen great attention from researchers, which

has achieved amazing and promising results. The CK+ database is widely used by researchers to evaluate their proposed deep learning systems. For this reason, we tested our method with CK+ to compare our method with deep learning. In table 4.27 we show four recent works by Meng et al. (2017); Cai et al. (2018); Yang et al. (2018); Zhang et al. (2018). We use the same number of CK+ images (981) as the same as Meng et al. (2017); Cai et al. (2018); Yang et al. (2018) and 10-fold cross-validation. Our method gave 91.30% overall accuracy whereas the GAN (cGAN) Yang et al. (2018) gave 6% higher accuracy than our method. Zhang et al. (2018) method achieved 98.00% but with only 6 facial expressions: angry, happy, surprise, sad, disgust and fear. In other words, our method, which based is on a combination of image descriptors and weighted pairwise random forest classifiers , achieved better results rather than using only one. Deep learning still gives better results than traditional image features.

4.10 Conclusion

Table 4.28 summarises the improvement progress with the two databases, KDEF and eLFW, after combining the three texture descriptors, rather than using only one. Moreover, the table shows how the masking gives better accuracy than without. It is clear from the table that the random weighted-pairwise classification gives much better accuracy.

Rather than using a single type of texture descriptor for the appearance-based classification of emotions, we showed that a combination of LBP, HOG and D-SURF significantly increases classification accuracy. Furthermore, the feature selection was

Table 4.28: The improvement progress summary for the proposed method.

	Accuracy			Precision			Recall			F1 Score		
Method	**KDEF**	**CK+**	**eLFW**	**KDEF**	**CK+**	**eLFW**	**KDEF**	**CK+**	**eLFW**	**KDEF**	**CK+**	**eLFW**
HOG	73.10%	67.70%	56.90%	77.90%	61.70%	52.20%	73.10%	63.30%	56.30%	71.40%	60.90%	52.60%
LBP	80.10%	71.60%	60.90%	85.00%	66.40%	55.90%	80.10%	68.10%	59.20%	79.10%	65.70%	56.10%
D-SURF	70.50%	60.50%	51.20%	75.30%	54.00%	46.70%	70.50%	56.80%	48.80%	67.70%	54.10%	46.50%
Combined feature	82.20%	78.30%	67.30%	83.40%	74.80%	59.80%	82.20%	75.80%	61.40%	82.30%	73.90%	59.60%
Masked imeages	89.90%	82.20%	71.60%	90.80%	79.10%	64.10%	89.80%	81.30%	66.00%	89.70%	79.20%	64.00%
Equally pair-wise classification	92.20%	89.40%	73.30%	93.20%	88.20%	67.40%	92.20%	88.80%	68.70%	92.10%	88.20%	66.90%
Weighted pair-wise classification	95.10%	91.13%	76.6%	95.70%	90.70%	71.10%	95.10%	89.40%	72.10%	95.00%	89.60%	70.60%

used to empirically identify the important regions of the faces for classifying emotion. As might be expected these are mainly around the eyes, mouth, the creases on either side of the nose and the forehead. Use of our empirically defined importance mask enhances the classification accuracy.

Further improvements to classification accuracy were obtained by pairwise weighted voting between dichotomous classifiers, and we showed how to learn optimal weights using an evolutionary algorithm. The resulting accuracies are significantly better than the current published state of the art results on the posed KDEF data. Nonetheless, particularly for spontaneous data, classification of fear, anger, disgust, sadness and surprise remains imperfect, and we obtain overall classification accuracies of about 77%. We observe that these are the emotions that humans find more challenging to classify from static images in eLFW data.

Chapter 5

Facial expression recognition in video

Contents

5.1 Introduction 90

5.2 DynEmo database preparation 91

5.3 Video Classification Experiments 94

5.4 Smoothing 98

5.4.1 Smoothing techniques overview 98

5.4.2 Smoothing optimisation 100

5.5 Conclusion 104

5.1 Introduction

In this chapter, we apply the method proposed in chapter 4 to the DynEmo (Tcherkassof et al., 2013) video database. We investigate methods of smoothing the classifier predictions to exploit the temporal continuity of emotions and therefore, the classification error. Several smoothing techniques are investigated and optimised.

The field of video automatic facial expression analysis has grown in recent years. Nevertheless, most researchers still depend on databases that contain acted emotions from models or actors (Pantic et al., 2005; O'Toole et al., 2005; Lucey et al., 2010). It is clear that the way that facial expressions in real life are different from those posed in many databases. A recent exciting database showing spontaneous facial expression is the DynEmo database which was created by a group of psychological researchers, computer scientists, statisticians, experimentation and instrumentation specialists, and a legal professional (Tcherkassof et al., 2013). The DynEmo database contains dynamic and natural emotional facial expressions filmed in natural but standardised conditions.

To compare our proposed method with the state-of-the-art methods, we use all sequences of The Extended Cohn Kanade (CK+) database which labelled as the seven basic facial expressions, fear, anger, disgust, happy, neutral, sad and surprised. CK+ database is the most widely used laboratory-controlled database for evaluating facial expression recognition systems. CK+ includes 593 video sequences from 123 subjects. The sequences differ in duration between 10 to 60 frames and show a shift

Table 5.1: DynEmo database data type.

C_INDUC	SEX_Sujet	SEXE_Juge	C_Juge	C_Video	E_Detectee	TC_Debut	TC_Fin
EM	F	N	237	DVD3_5.mpg	Surprise	7181	10183
EM	F	N	237	DVD3_5.mpg	Surprise	23438	29080
EM	F	N	237	DVD3_5.mpg	Surprise	39798	50305

from a neutral facial expression to the peak expression.

In this chapter, section 5.2 will show how the DynEmo database has been prepared to be usable in our work. In section 5.3, we show the result of applying the proposed method in the previous chapter to the new video database. Section 5.4 shows how smoothing the classifier scores improves accuracy. Finally, conclusions are drawn in section 5.5.

5.2 DynEmo database preparation

The DynEmo database has been labelled over time, not frame by frame. In our experiments, we need labelled frames to be used for training and testing. The researchers in the DynEmo database worked on the following facial expression expressions in the DynEmo database: curiosity, happy, Surprise, boredom, disgust, fright (fear), shame, annoyance (anger), disappointment, humiliation and other marginal expressions. This was in an attempt to extract as much as possible of the expressions. For the data collection, the researchers have recorded films using hidden cameras while people (called encoders) were sitting on a chair at a small table facing the wall where a PC was projected on the wall using a video projector. During the video playback,

the experimenters were sitting in an adjacent room watching the encoder's reactions. There recorded videos properties are width: 768, height: 576, frame rate: 25 frames per second, bits per pixel: 24 and video format: RGB24.

Table 5.1 shows a sample of the original data layout in the DynEmo database. The most important columns for us in the data are C_Video, which means the video name, E_Detectee which is the labelled emotion and TC_Debut and TC_Fin are the start and end time in milliseconds. The DynEmo database contains 358 videos, but not all of them were available to download. The first objective for us, was to make the labelling based on frames rather than time. To ensure that there was a consensus on the judgements on each frame. Work started by preparing the database to be usable in training and testing. From the available videos on their website, we extracted the 44 sequences for 5 facial expressions, and these 44 sequences were consensus voted.

Similar to the work in chapter 3, we need to ensure that there was a consensus on the facial expression on each frame. To achieve this, each frame must be assigned the same emotion and at least four different judges. To qualify the quality of the consensus, we calculate the entropy value for the votes. The entropy must be less than 1 to accept the consensus vote. The entropy for n probabilities $(p_1, p_2, ..., p_n)$ is calculated by using the following equation:

$$H = -\sum_{i=1}^{n} p_i \log_2(p_i) \tag{5.1}$$

where p; is the fraction of judges voting for class i.

We adapted the labelling way to be based in every single frame rather than the original method, which was based on time sectors. The 44 sequences include 14543

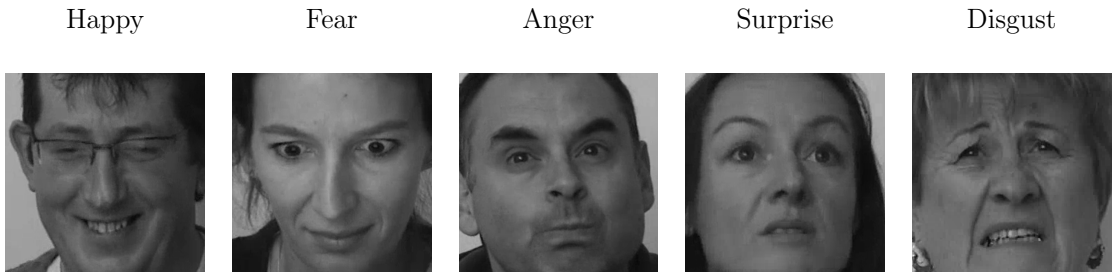


Figure 5.1: Examples of the five facial expression in the database

frames that had been voted with consensus ($H < 1$). We removed some the short parts of the 44 videos which have no consensus ($H > 1$). Some videos have more than one expression during the video, so we cut out the short parts from those videos which did not get a consensus, and we kept the sequences of frames with a consensus vote. Only 5 facial emotions were remaining after calculating the entropy: fear, anger, disgust, happy and surprise, as shown in figure 5.1.

Facial expressions in real life vary even for the same person. For example, the happy expression has many different levels. The same person may express a small smile or a wide smile. People sometimes try to hide their expressions, so this causes facial expression variation. This variation is a big challenge for natural facial expressions recognition. Figures 5.2a, 5.2b and 5.2c show happy expression variation for the same person from the DynEmo database. In most unnatural expression databases, the actors express a very similar way of showing the same facial expression.

The DynEmo preparation process produced 44 videos labelled based on frames rather than time. Table 5.2 illustrates the new dataset, which contains 14543 frames distributed over 5 facial expressions as 8902 frames labelled as happy; 313 as fear,

Table 5.2: General statistics about the new database

Number of videos	Happy frames	Fear frames	Anger frames	Surprise frames	Disgust frames
44	8902	313	2192	2665	471

(a) Low smile. (b) Average smile. (c) Wide smile (laugh).

Figure 5.2: Happy expression variation for the same person.

2192 as angry, 2665 as surprise and 471 as disgust. Figure 5.1 shows examples of newly labelled frames. We used these data in the experiments described in section 5.3.

5.3 Video Classification Experiments

In our experiments, we used five facial expressions: happy, fear, anger, surprise and disgust. We trained our new system on videos and photos containing the 5 facial expressions. Because the videos are sequences of frames, we do not need to train the system using all frames in the same video. We chose only one frame from each 25 frames for: happy, anger and surprise and all fear and disgust frames. We add to each class of the training data 70 images from KDEF and 70 images from eLFW. It is important to know that in the testing, we have never used any video for testing and

Table 5.3: Confusion matrix on the DynEmo database using the proposed method, the left is by 5-class RF classifier, and the right 10-pairwise random forest classifiers

Overall accuracy was 79.6%

	FE	**AN**	**DI**	**HA**	**SU**
FE	**0.390**	0.093	0.163	0.118	0.236
AN	0.045	**0.728**	0.175	0.018	0.034
DI	0.085	0.176	**0.609**	0.011	0.119
HA	0.044	0.043	0.022	**0.850**	0.040
SU	0.134	0.026	0.072	0.014	**0.753**

Overall accuracy was 83.4%

	FE	**AN**	**DI**	**HA**	**SU**
FE	**0.422**	0.093	0.150	0.118	0.217
AN	0.038	**0.776**	0.141	0.019	0.026
DI	0.064	0.174	**0.631**	0.011	0.121
HA	0.029	0.030	0.018	**0.890**	0.034
SU	0.114	0.024	0.068	0.014	**0.781**

training at the same time. This training and testing were repeated 5 times (5-fold cross-validation), with each time leaving different complete videos out for testing, and training with the remaining videos mixed with KDEF and eLFW images.

A trained 5000 trees random forest model has been used to predict the testing videos, and this returns scores for each training class. The scores (posterior probability) generated by each tree have been represented as a matrix with one row per predicted frame and one column per class. Figure 5.3 shows the prediction behaviour of random forest classifiers with two happy-labelled videos (DVD31_1 and DVD14_). Random forest returned voting values (scores) for each frame referring to the training classes.

Table 5.3 right shows two confusion matrices, the left for 1 RF classifier and the right shows a confusion matrix for the 10 pairwise classifiers. In the left table, the happy expression was the best rate 85% accuracy, and fear like with the static images was the lowest accuracy with only 39%. The overall accuracy was 79.6%. The

Table 5.4: Confusion matrix on the CK+ database using the proposed method, the left is by 5-class RF classifier, and the right 10-pairwise random forest classifiers

Overall accuracy was 81.2%

	FE	AN	DI	HA	NE	SA	SU
FE	**0.639**	0.033	0.093	0.022	0.082	0.064	0.066
AN	0.022	**0.810**	0.023	0.005	0.103	0.017	0.022
DI	0.021	0.013	**0.801**	0.015	0.104	0.028	0.020
HA	0.008	0.017	0.010	**0.860**	0.080	0.018	0.008
NE	0.013	0.013	0.026	0.034	**0.863**	0.039	0.013
SA	0.007	0.022	0.013	0.018	0.122	**0.808**	0.009
SU	0.013	0.015	0.011	0.010	0.093	0.020	**0.838**

Overall accuracy was 89.0%

	FE	AN	DI	HA	NE	SA	SU
FE	**0.828**	0.018	0.048	0.020	0.035	0.016	0.035
AN	0.008	**0.889**	0.011	0.003	0.069	0.006	0.014
DI	0.015	0.006	**0.872**	0.013	0.071	0.012	0.012
HA	0.005	0.008	0.007	**0.924**	0.044	0.008	0.005
NE	0.004	0.009	0.026	0.021	**0.914**	0.013	0.013
SA	0.007	0.018	0.009	0.015	0.044	**0.901**	0.005
SU	0.011	0.011	0.008	0.008	0.065	0.014	**0.883**

overall rise by pairwise classification in the right table to 83.4%, which is an increase of nearly 4%, similar to the static images. Table 5.4 right shows two confusion matrices, the left for one RF classifier, and the right shows a confusion matrix for 10 pairwise classifiers. In the left table, the happy expression was the best rate 86% accuracy, and fear like the static images was the lowest accuracy with only 63.9%. The overall accuracy was 81.2%. The overall rise by pairwise classification in the right table to 89.0%, which is an increase of nearly 8%, similar to the static images. Like with the static images, our proposed system gives good results with the videos as shown in table 5.3 and 5.4.

The weights are shown in table 5.5. We note that the weights have a similar weights-pattern to table 4.24, so for example, fear-vs-disgust is the largest weight in the fear row as well. In the next step in section 5.4, we investigate improving the performance of the classifiers by smoothing their scores, and then we impose that the short misclassification for few sequence frames should be fixed depending on the

Table 5.5: Five-Class random forest pair-classifiers with their optimised weights

	Fear	**Anger**	**Disgust**	**Happy**	**Surprise**
Fear	x	0.281	0.295	0.174	0.249
Anger	0.257	x	0.301	0.283	0.158
Disgust	0.272	0.274	x	0.248	0.205
Happy	0.268	0.251	0.259	x	0.22
Surprise	0.281	0.273	0.257	0.187	x

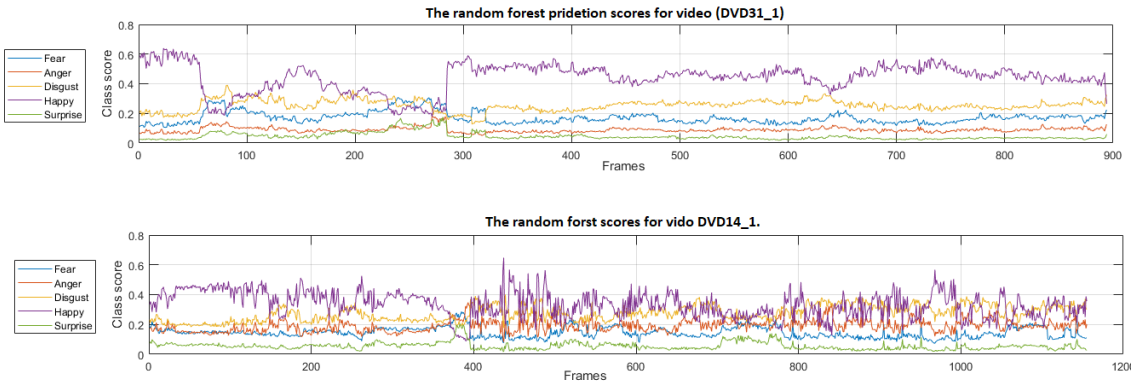


Figure 5.3: Classifier prediction behaviour for 2 happy-labelled videos

nearby frames.

It may be expected that the neighbouring frames in a video mostly contain the same facial expression. In other words, we assume that if the classifier has classified a frame as fear where the adjacent frames were happy, then it is likely to be a misclassification. To solve this problem, we suppose that smoothing the posterior probabilities of the classifier may reduce this misclassification and enhance the overall accuracy.

5.4 Smoothing

In smoothing, the individual data points that are higher than the adjacent points are reduced, and those that are lower than the adjacent points are increased (Simonoff, 2012). In this work, the smoothing techniques are applied to the classifier's scores in order to reduce the misclassification according to a group of adjacent frames called span.

5.4.1 Smoothing techniques overview

To smooth classifiers' scores, several smoothing techniques have been tested: Moving average smoothing, Locally weighted scatterplot smoothing (LOWESS and LOESS) (Cleveland, 1979; Cleveland and Devlin, 1988) and their robust versions, the Savitzky-Golay filter (Savitzky and Golay, 1964).

One of the simplest ways to smooth fluctuating data is by a moving average. The moving average filtering smoothed value is determined by the neighbouring data points within a span. Moving average filtering smooths data by displacing each data point with the average of data points within the span (Smith, 1997). This method is described by the following equation:

$$y_s(i) = \frac{1}{2U+1} \sum_{u=-U}^{u=+U} y(i+u) \tag{5.2}$$

where $y_s(i)$ is the smoothed value for the i_{th} data point, $U \geqslant 0$ is integer represents the number of neighbouring data points on either side of $y_s(i)$, and $2U+1$ is the span.

Another method we used in this chapter is called the Savitzky-Golay filter (Savitzky and Golay, 1964). It aims to remove high-frequency noise without causing distortion. The method is based on least-squares error approximation, by applying a smooth polynomial line on the data points in the neighbourhood of a sample and adjusting the latter's amplitude to the fitted lines. This method is known as a weighted moving average with weighting given as a polynomial of specific degree (Savitzky and Golay, 1964). The coefficients of a Savitzky-Golay smoothing method, when applied to data, fit a polynomial of the degree k to $U = U_r + U_l + 1$ points of the signal, where U is the span. U_r and U_l are signal points in the right and data points in the left of a current data point to the span, respectively; $U_r = U_l$ here.

Locally weighted scatterplot smoothing LOWES (Cleveland, 1979) and locally estimated scatterplot smoothing LOESS (Cleveland and Devlin, 1988) are also known as moving regression. These methods depend on a weighting function with the effect that the influence of a neighbouring value on the smoothed value at a particular location decreases with their distance to that location. The difference between LOWESS and LOESS is that LOWESS allows for only 1 predictor, whereas LOESS can be used to smooth multivariate data into a kind of surface.

The methods start with computing the regression weights for each data point in the data window. Unlike the moving average filter, it is possible that the number of points to the left and right may differ.

In the next step, a weighted linear least-squares regression is performed. For LOWESS, the regression uses the first-degree polynomial. For LOESS, the regression uses a second-degree polynomial. Finally, the smoothed value is given by the

Table 5.6: Optimising of the smoothing span by Nelder–Mead (Staring point (0.5))

Smoothing method	**Optimal smoothing (span)**	**Overall accuracy**
Savitzky-Golay	69	0.833
Moving average	**73**	**0.871**
Lowess	**87**	**0.869**
Loess	64	0.794
Robust Lowess	88	0.845
Robust Loess	81	0.834

weighted regression at the predictor value of interest. LOWESS and LOESS methods have robust versions that include an additional calculation of robust weights, which is resistant to outliers values in the span.

5.4.2 Smoothing optimisation

To smooth the classifier scores, we need to find the optimal span size to get the best results. To achieve that, we use the Nelder–Mead method (Nelder and Mead, 1965) which is a popular numerical method to find the maximum or minimum of an objective function. It provides improvements in the first few iterations and quickly produces satisfactory results (Barati, 2011). By applying Nelder–Mead to the classifier scores, we aim to find the optimal span that minimises the error. Table 5.6 shows the optimisation results for the 6 smoothing methods described above. The table shows the best window size (span), i.e. that gives the best accuracy for each smoothing method. We found that the moving average and Lowess methods give the

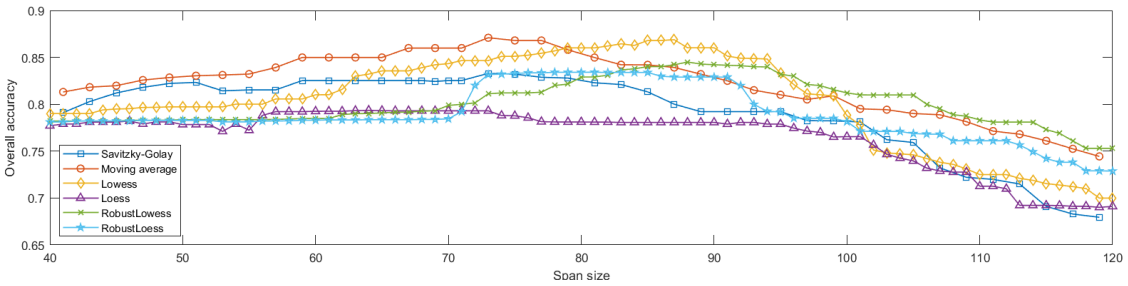


Figure 5.4: The overall accuracy vs. smoothing span, (25 frames per second.)

best smoothing accuracy. The Moving average smoothing result was 87.1%, where Lowess was 86.9%.

Figure 5.4 illustrates the relationship between changing the smoothing span and the overall accuracy. We chose a range of span size from 40 to 120 to show this figure, and these correspond to 1.6 and 4.8 seconds of video since the video rate is 25 frames per second. All the optimal span sizes are in this size range. All smoothing methods give the best accuracy between 65 and 90 (2.6 to 3.6 seconds). LOESS is consistently poorer, whereas the moving average filter and the LOWESS are the best.

Figure 5.5 shows an example of a smoothing video (DVD79 _5). The top plot shows the classifier scores before the smoothing, followed by the smoothing results obtained by applying the moving average to the optimal span. This classifier gave 100% accuracy until frame number 20, which was misclassified as happy, and the smoothing fixed this misclassification. From figure 5.5 (top) a noticeable disorder between frame 96 to 144 can be seen. This disorder has been fixed, as shown in the middle plot, and the overall accuracy rose as illustrated in the bottom of the figure. Between frame 426 and 485, we can see another disorder has been fixed.

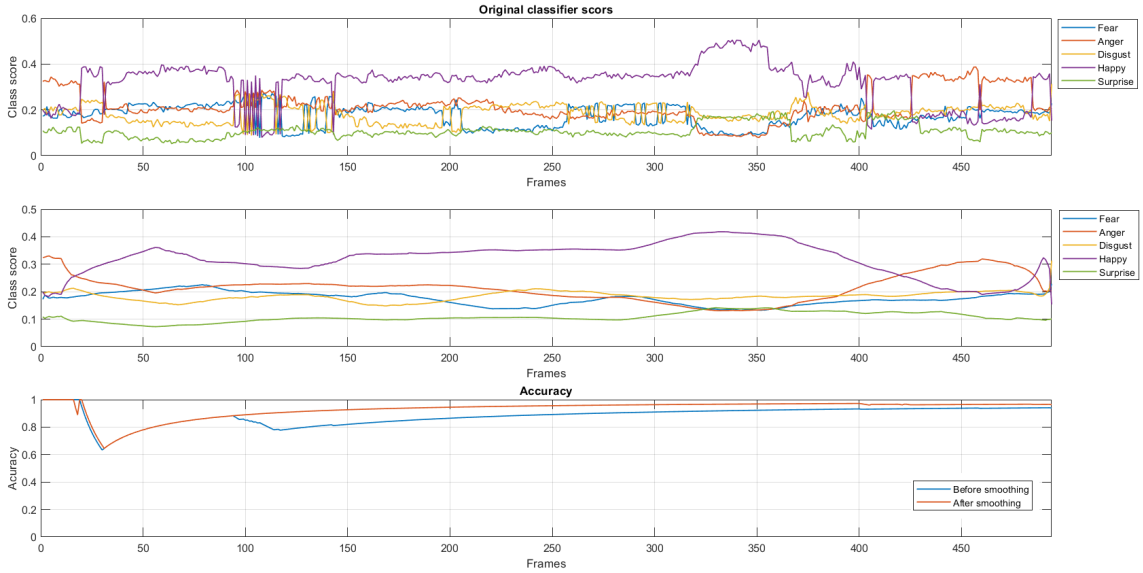


Figure 5.5: Smoothing result for video DVD79_5 with the optimal span. The top plot shows the scores before smoothing whereas the middle shows the scores after smoothing. The bottom plot illustrates the overall accuracy for all frames before a particular frame.

Table 5.7 shows two confusion matrices for the DynEmo database and by applying 5-fold cross-validation after applying the smoothing method. The right confusion matrix shows the 1 RF classifier and the left shows the 10-pairwise classifiers. We can see that something improved both methods with marked increase, from 79.6% to 87.1% for the one classifier, which is an increase of nearly 8%, and from 83.4% to 88.3% for the one classifier which is an increase of nearly 5%. Table 5.8 shows two confusion matrices for the same the CK+ database sequences after applying the smoothing method. The right confusion matrix show the 1 RF classifier and the left shows the 10-pairwise classifiers. We can see that something improved both methods

Table 5.7: Confusion matrices for the DynEmo database using the proposed method after smoothing, the left is by 1 RF classifier, and the right 10-pairwise classifiers

Overall accuracy was 87.1% (Normal RF classifier)

	FE	**AN**	**DI**	**HA**	**SU**
FE	**0.457**	0.045	0.198	0.099	0.201
AN	0.043	**0.766**	0.145	0.012	0.034
DI	0.085	0.117	**0.679**	0.000	0.119
HA	0.014	0.009	0.018	**0.938**	0.021
SU	0.083	0.009	0.071	0.022	**0.815**

Overall accuracy was 88.3% (10-pairwise classifiers).

	FE	**AN**	**DI**	**HA**	**SU**
FE	**0.543**	0.058	0.157	0.054	0.188
AN	0.031	**0.826**	0.124	0.010	0.010
DI	0.064	0.174	**0.641**	0.000	0.121
HA	0.013	0.011	0.009	**0.952**	0.016
SU	0.115	0.026	0.065	0.010	**0.784**

with marked increase, from 79.6% to 84.4% for the one classifier, which is an increase of nearly 5%, and from 89.0% to 93.% for the one classifier which is an increase of nearly 5%.

Table 5.8: Confusion matrices for the CK+ database using the proposed method after smoothing, the left is by one RF classifier, and the right 10-pairwise classifiers

Overall accuracy was 84.4% (Normal RF classifier)

	FE	**AN**	**DI**	**HA**	**NE**	**SA**	**SU**
FE	**0.749**	0.033	0.081	0.009	0.077	0.031	0.020
AN	0.017	**0.837**	0.016	0.002	0.095	0.014	0.021
DI	0.018	0.009	**0.827**	0.009	0.096	0.029	0.012
HA	0.004	0.016	0.010	**0.876**	0.081	0.008	0.006
NE	0.013	0.013	0.017	0.030	**0.884**	0.039	0.004
SA	0.002	0.005	0.004	0.015	0.124	**0.848**	0.002
SU	0.008	0.010	0.008	0.004	0.087	0.026	**0.857**

Overall accuracy was 93.2% (10-pairwise classifiers).

	FE	**AN**	**DI**	**HA**	**NE**	**SA**	**SU**
FE	**0.874**	0.022	0.044	0.005	0.026	0.009	0.020
AN	0.002	**0.932**	0.003	0.000	0.059	0.000	0.005
DI	0.010	0.000	**0.924**	0.000	0.058	0.002	0.006
HA	0.001	0.008	0.004	**0.952**	0.032	0.002	0.003
NE	0.000	0.000	0.017	0.009	**0.970**	0.004	0.000
SA	0.004	0.000	0.005	0.000	0.042	**0.949**	0.000
SU	0.001	0.004	0.000	0.000	0.053	0.014	**0.929**

Table 5.9: Comparison of the classification accuracy of smoothed random forest pairwise classification using the masked LBP, HOG and D-SURF texture features with other recent techniques. Evaluation on the CK+ databases (for dynamic-based deep facial expression recognition).

Method	Accuracy
Kim et al. (2017)	95.22%
Kim et al. (2017)	97.93%
Sun et al. (2017)	97.28%
Kuo et al. (2018)	98.47%
Our proposed method(smoothed pairwise classifiers)	93.20%

Table 5.9 shows a comparison with some of the state-of-the-art methods which are based on deep learning algorithms. All of the shown methods in table 5.9 have been applied to the CK+ sequences. Kuo et al. (2018) have achieved excellent performance, with overall accuracy reached to 98.47. It is clear that the deep learning methods give better results than our method, but our result is still comparable. Our proposed method can be complementary to many methods which work with dynamic problems, like the sequences frames in videos.

5.5 Conclusion

In this chapter, we used an existing psychological facial emotion video dataset called DynEmo and prepared it to be used for machine training and testing purposes. We

prepared 44 videos to include 14543 frames distributed on 5 facial expressions happy, fear, anger, surprise and disgust. We use the method proposed in chapter 4 to train the random forest model with data from the DynEmo dataset mixed with some images from masked KDEF and eLFW. In order to compare our results with the state-of-the-art, we used 7 facial expression from the CK+ database sequences. We tested some smoothing techniques to reduce the misclassification by smoothing the classifier scores (posterior probability). To find the optimal smoothing span, we used the Nelder–Mead method to minimise the error.

As a result, like static images, our proposed system gives good results with videos. We found that applying smoothing methods with an optimal span value improved the performance of the classifiers by smoothing their scores. As we have imposed, the small misclassification should be fixed depending on the nearby frames. The best span size is between 65 and 90, which is 2.8 to 3.6 seconds. This effects on the ability of our proposed system to work in real-time applications because it needs about 3 seconds to give the most accurate results.

Chapter 6

Conclusion and perspectives

With the advancement in human-computer interaction, machines are becoming a more critical part of our lives. Facial expressions are an essential language to understand more about humans. One important factor that should be considered in developing a spontaneous facial expression recognition system is the availability of a useful database that does not contain posed portraits of actors displaying emotions. Our work started with the building of a website to construct a new natural facial expression database based on a current database called Labelled Faces in the Wild (LFW) (Huang et al., 2007b).This was done by asking citizens to vote what emotions they see in a selected group of images. Another current emotionally-labelled database, the Karolinska Directed Emotional Faces (KDEF) (Lundqvist et al., 1998) was used to evaluate the citizens' performance. The new database is called the Emotional Labelled Faces in the Wild (eLFW).

We presented an automated new approach for facial expression recognition of 7 emotions. Three types of texture features from static images were combined:

Local Binary Patterns (LBP) Ojala et al. (1996), Histogram of Oriented Gradients (HOG) (Dalal and Triggs, 2005) and Dense Speeded Up Robust Features (D-SURF) (Lowe, 2004; Uijlings et al., 2010), then the resulting features were classified using random forests. The use of random forests (Breiman, 2001) allows identification of the most important feature types and facial locations for emotion classification. Regions around the eyes, forehead, sides of the nose and the mouth were found to be the most significant. We classified the important features with random forest and support vector machines (Cortes and Vapnik, 1995), and we found that the classification performance became better than using all extracted facial features. We achieved better than some state-of-the-art accuracies using multiple texture feature descriptors.

Further improvements to classification accuracy were obtained by pairwise weighted voting between dichotomous classifiers, and we showed how to learn optimal weights using an evolutionary algorithm. The resulting accuracies were significantly better than the current published state of the art results on the posed KDEF data. Nonetheless, particularly for unposed data, classification of fear, anger, disgust, sadness and surprise remains imperfect, and we obtain classification accuracies of about 77%. We observe that these were the emotions that humans find more difficult to classify from static images in the eLFW data.

We use the method which was proposed in chapter 4 to train random forests model with data from the DynEmo dataset mixed with some images from KDEF and eLFW. We tested some smoothing techniques to reduce the misclassification by smoothing the classifiers scores. To find the optimal smoothing span, we used the

Nelder–Mead method to minimise the error.

As a result, like static images, our proposed system gives good results with videos. We found that applying smoothing methods with an optimal span value improved the performance of the classifiers by smoothing their scores. As we have imposed, that the small misclassification should be fixed depending on the nearby frames.

6.1 Future Work

Our work has examined 7 principle types of emotions, so more research effort is required focused on recognising more complicated facial expressions or on getting more information from expressions like stress (Giannakakis et al., 2017), pain, and mental states such as agreeing, disagreeing, lying, frustration and thinking as they have numerous application areas. Moreover, the differences in emotion recognition between males and females need more investigation to see how theses differences effect machine recognition ability (Wright et al., 2018).

In this thesis, we have considered spontaneous facial emotions recognition. For this mission, we used existing data contains spontaneous emotion, and we labelled it or modified the labelling method. However, our bodies are effectively contributing to our faces to show our emotion (Burgoon et al., 2016). It would be important in the future to use both humand and objects estimation techniques (Zhou et al., 2016; Kiforenko and Kraft, 2016; Mehta et al., 2017). For this, we may create a new database for video, and static images contain labelled data for both face and body language. 3D Models became powerful statistical models, which describes both identity and expression, with an “in-the-wild” texture model (James Booth, 2017).

3D models describe faces and bodies and strongly employable in the face and body analysis (Booth et al., 2016, 2018; Zafeiriou et al., 2018).

Our proposed method in this thesis works with frontal face images and in a lighting condition. However, in real life, we need to consider the facial pose and suboptimal lighting conditions. 3D models offer a promising solution to the problems facing 2D images such as pose and lighting (Liu and Ward, 2005). The recent success of convolutional neural networks (CNNs) (LeCun et al., 1998) in computer vision classification tasks, it has been extended to facial expression recognition problems (Fan et al., 2018). Detection using CNNs is a robust method against changes in shape due to camera lenses, different lighting conditions, different facial poses, the presence of partial occlusions and both horizontal and vertical shifts (Hijazi et al., 2015).

Real-time performance is a significant factor in many applications, as we saw in chapter 5, smoothing needs 2.6 to 3.6 seconds (the smoothing span size) to get the result. The time that was taken for image preprocessing and extracting the features is computationally expensive (Davison et al., 2018a), so we need to investigate how can we solve this problem in our proposed system. A possible solution that may resolve this issue is to explore Micro-Expression (Ekman and Rosenberg, 1997), which has taken great interest in recent times (Yap et al., 2018; Liong et al., 2018). Micro-expression recognition should be investigated and work with datasets based on the Action Unit with deep learning techniques (Merghani et al., 2018). Samm is a spontaneous micro-facial movement dataset (Davison et al., 2018b) is the highest resolution available and it includes a very diverse demographic of the micro-movement datasets

currently available which to be used in the future Micro-Expression works. This data set has a variety of emotions, which simulate spontaneous emotion expressions.

Bibliography

Anderson, K. and McOwan, P. W. (2006). A real-time automated system for the recognition of human facial expressions. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 36(1):96–105.

Anitha, C., Venkatesha, M., and Adiga, B. S. (2010). A survey on facial expression databases. *International Journal of Engineering Science and Technology*, 2(10):5158–5174.

Arnold, J. and Emerick, M. (2016). Emotional evaluation through facial recognition. Technical report, Penn State Wilkes-Barre University.

Barati, R. (2011). Parameter estimation of nonlinear muskingum models using Nelder-Mead simplex algorithm. *Journal of Hydrologic Engineering*, 16(11):946–954.

Bay, H., Ess, A., Tuytelaars, T., and Van Gool, L. (2008). Speeded-up robust features (SURF). *Computer Vision and Image Understanding*, 110(3):346–359.

Bay, H., Tuytelaars, T., and Van Gool, L. (2006). SURF: Speeded up robust features. *Computer Vision–ECCV 2006*, pages 404–417.

Belle, V. (2008). Detection and recognition of human faces using random forests for a mobile robot. *Master of Science Thesis, RWTH Aachen University*.

Berretti, S., Del Bimbo, A., Pala, P., Amor, B. B., and Daoudi, M. (2010). A set of selected SIFT features for 3D facial expression recognition. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 4125–4128. IEEE.

Bill Doerrfeld (2016). 20+ Emotion Recognition APIs That Will Leave You Impressed, and Concerned. https://nordicapis.com/20-emotion-recognition-apis-that-will-leave-you-impressed-and-concerned/. Last checked on August 11, 2018.

Booth, J., Roussos, A., Ververas, E., Antonakos, E., Poumpis, S., Panagakis, Y., and Zafeiriou, S. P. (2018). 3d reconstruction of "in-the-wild" faces in images and videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 2638–2652.

Booth, J., Roussos, A., Zafeiriou, S., Ponniah, A., and Dunaway, D. (2016). A 3d morphable model learnt from 10,000 faces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5543–5552.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24(2):123–140.

Breiman, L. (1999). Random forests—random features. Technical report, Statistics-Department, University of California.

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.

Breiman, L. (2002). Manual on setting up, using, and understanding random forests v3. 1. Technical report, Statistics Department University of California Berkeley, CA, USA.

Breiman, L. F., Friedman, J., and Olshen, S. (1983). Cj, 1984. classification and regression trees. *Pacific Grove, Kalifornien*.

Brochu, E., Cora, V. M., and De Freitas, N. (2010). A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599*.

Burgoon, J. K., Guerrero, L. K., and Floyd, K. (2016). *Nonverbal communication*. Routledge.

Cai, J., Meng, Z., Khan, A. S., Li, Z., O'Reilly, J., and Tong, Y. (2018). Island loss for learning discriminative features in facial expression recognition. In *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pages 302–309. IEEE.

Carcagnì, P., Del Coco, M., Leo, M., and Distante, C. (2015). Facial expression recognition and histograms of oriented gradients: a comprehensive study. *SpringerPlus*, 4(1):1–25.

Caridakis, G., Malatesta, L., Kessous, L., Amir, N., Raouzaiou, A., and Karpouzis, K. (2006). Modeling naturalistic affective states via facial and vocal expressions recognition. In *Proceedings of the 8th international conference on Multimodal interfaces*, pages 146–154. ACM.

Chen, J., Chen, Z., Chi, Z., and Fu, H. (2014). Facial expression recognition based on facial components detection and hog features. In *International Workshops on Electrical and Computer Engineering Subfields*, pages 884–888.

Chen, X., Udupa, J. K., Alavi, A., and Torigian, D. A. (2013). GC-ASM: Synergistic integration of graph-cut and active shape model strategies for medical image segmentation. *Computer Vision and Image Understanding*, 117(5):513–524.

Cheng, S., Kotsia, I., Pantic, M., and Zafeiriou, S. (2017). 4dfab: A large scale 4d facial expression database for biometric applications. *arXiv preprint arXiv:1712.01443*.

Cheon, Y. and Kim, D. (2008). A natural facial expression recognition using differential-AAM and k-NNS. In *Multimedia, 2008. ISM 2008. Tenth IEEE International Symposium on*, pages 220–227. IEEE.

Chih-Wei Hsu, Chih-Chung Chang, C.-J. L. (2003). A practical guide to support vector classification. Technical report, Taipei, Taiwan: National Taiwan University.

Cleveland, W. S. (1979). Robust locally weighted regression and smoothing scatterplots. *Journal of the American Statistical Association*, 74(368):829–836.

Cleveland, W. S. and Devlin, S. J. (1988). Locally weighted regression: an approach to regression analysis by local fitting. *Journal of the American Statistical Association*, 83(403):596–610.

Cohn, J. F., Zlochower, A. J., Lien, J. J., and Kanade, T. (1998). Feature-point tracking by optical flow discriminates subtle differences in facial expression. In *Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference On*, pages 396–401. IEEE.

Cootes, T. F., Edwards, G. J., and Taylor, C. J. (1998). Active appearance models. In *ECCV98*, volume 2, pages 484–498.

Cootes, T. F., Edwards, G. J., and Taylor, C. J. (2001). Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):681–685.

Cootes, T. F. and Taylor, C. J. (1992). Active shape models-'smart snakes'. In *BMVC*, volume 92, pages 266–275.

Cootes, T. F., Taylor, C. J., Cooper, D. H., and Graham, J. (1992). Training models of shape from sets of examples. In *BMVC92*, pages 9–18. Springer.

Cootes, T. F., Taylor, C. J., Cooper, D. H., and Graham, J. (1995). Active shape models-their training and application. *Computer Vision and Image Understanding*, 61(1):38–59.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3):273–297.

Dahmane, M. and Meunier, J. (2011). Emotion recognition using dynamic grid-based HOG features. In *Automatic Face & Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on*, pages 884–888. IEEE.

Dalal, N. and Triggs, B. (2005). 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.

Datta, S., Sen, D., and Balasubramanian, R. (2017). Integrating geometric and textural features for facial emotion classification using SVM frameworks. In *Proceedings of International Conference on Computer Vision and Image Processing*, pages 619–628. Springer.

Davison, A., Merghani, W., and Yap, M. (2018a). Objective classes for micro-facial expression recognition. *Journal of Imaging*, 4(10):119.

Davison, A. K., Lansley, C., Costen, N., Tan, K., and Yap, M. H. (2018b). Samm: A spontaneous micro-facial movement dataset. *IEEE Transactions on Affective Computing*, 9(1):116–129.

Davison, A. K., Yap, M. H., Costen, N., Tan, K., Lansley, C., and Leightley, D. (2014). Micro-facial movements: an investigation on spatio-temporal descriptors. In *European Conference on Computer Vision*, pages 111–123. Springer.

Den Uyl, M. and Van Kuilenburg, H. (2005). The facereader: Online facial expression recognition. In *Proceedings of Measuring Behavior*, volume 30, pages 589–590. Citeseer.

Dhall, A., Goecke, R., Ghosh, S., Joshi, J., Hoey, J., and Gedeon, T. (2017). From individual to group-level emotion recognition: Emotiw 5.0. In *Proceedings of the*

19th ACM International Conference on Multimodal Interaction, pages 524–528. ACM.

Dhall, A., Goecke, R., Lucey, S., Gedeon, T., et al. (2012). Collecting large, richly annotated facial-expression databases from movies. *IEEE multimedia*, 19(3):34–41.

Dhall, A., Ramana Murthy, O., Goecke, R., Joshi, J., and Gedeon, T. (2015). Video and image based emotion recognition challenges in the wild: Emotiw 2015. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, pages 423–426. ACM.

Edwards, G. J., Taylor, C. J., and Cootes, T. F. (1998). Interpreting face images using active appearance models. In *Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on*, pages 300–305. IEEE.

Ekman, P. (1973). *Cross-cultural studies of facial expression*. New York: Academic Press.

Ekman, P. (1978). Facial expression. *Nonverbal Behavior and Communication*, pages 97–116.

Ekman, P. and Rosenberg, E. L. (1997). *What the face reveals: Basic and applied studies of spontaneous expression using the Facial Action Coding System (FACS)*. Oxford University Press, USA.

El Ayadi, M., Kamel, M. S., and Karray, F. (2011). Survey on speech emotion recognition: Features, classification schemes, and databases. *Pattern Recognition*, 44(3):572–587.

Essa, I. A. and Pentland, A. P. (1995). Facial expression recognition using a dynamic model and motion energy. In *Computer Vision, 1995. Proceedings., Fifth International Conference on*, pages 360–367. IEEE.

Fabian Benitez-Quiroz, C., Srinivasan, R., and Martinez, A. M. (2016). Emotionet: An accurate, real-time algorithm for the automatic annotation of a million facial expressions in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5562–5570.

Fan, Y., Lam, J. C., and Li, V. O. (2018). Multi-Region Ensemble Convolutional Neural Network for Facial Expression Recognition. In *International Conference on Artificial Neural Networks*, pages 84–94. Springer.

Fanelli, G., Gall, J., and Van Gool, L. (2011). Real time head pose estimation with random regression forests. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 617–624. IEEE.

Giannakakis, G., Pediaditis, M., Manousos, D., Kazantzaki, E., Chiarugi, F., Simos, P. G., Marias, K., and Tsiknakis, M. (2017). Stress and anxiety detection using facial cues from videos. *Biomedical Signal Processing and Control*, 31:89–101.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. in advances in neural information processing systems. pages 2672–2680.

Goodfellow, I. J., Erhan, D., Carrier, P. L., Courville, A., Mirza, M., Hamner, B., Cukierski, W., Tang, Y., Thaler, D., Lee, D.-H., et al. (2013). Challenges in repre-

sentation learning: A report on three machine learning contests. In *International Conference on Neural Information Processing*, pages 117–124. Springer.

Gross, R., Matthews, I., Cohn, J., Kanade, T., and Baker, S. (2010). Multi-pie. *Image and Vision Computing*, 28(5):807–813.

Hansen, N. (2006). The CMA evolution strategy: a comparing review. In Lozano, J., Larranaga, P., Inza, I., and Bengoetxea, E., editors, *Towards a new evolutionary computation. Advances on estimation of distribution algorithms*, pages 75–102. Springer.

Hazewinkel, M. (2001). Affine transformation. *Encyclopedia of Mathematics, Springer*.

Hijazi, S., Kumar, R., and Rowen, C. (2015). Using convolutional neural networks for image recognition. Technical report. Also available as http://www.multimediadocs.com/assets/cadence_emea/documents/using_convolutional_neural_networks_for_image_recognition.pdf.

Hinton, G. E. and Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507.

Huang, D., Shan, C., Ardabilian, M., Wang, Y., and Chen, L. (2011). Local binary patterns and its application to facial image analysis: a survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 41(6):765–781.

Huang, G., Mattar, M., Lee, H., and Learned-Miller, E. G. (2012). Learning to align from scratch. In *Advances in Neural Information Processing Systems*, pages 764–772.

Huang, G. B., Ramesh, M., Berg, T., and Learned-Miller, E. (2007a). Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst.

Huang, G. B., Ramesh, M., Berg, T., and Learned-Miller, E. (2007b). Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical report, University of Massachusetts, Amherst.

James Booth, Epameinondas Antonakos, S. P. G. T. Y. P. S. Z. (2017). 3d face morphable models "in-the-wild". In *Proceedings of the IEEE Conference on ComputerVision and Pattern Recognition*, volume 1, pages 48–57.

Kanade, T., Cohn, J. F., and Tian, Y. (2000). Comprehensive database for facial expression analysis. In *Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on*, pages 46–53. IEEE.

Kanan, C. and Cottrell, G. W. (2012). Color-to-grayscale: does the method matter in image recognition? *PloS One*, 7(1):e29740.

Karami, E., Prasad, S., and Shehata, M. (2017). Image matching using SIFT, SURF, BRIEF and ORB: Performance comparison for distorted images. *Newfoundland Electrical and Computer Engineering Conference*.

Kendall, D. G. (1989). A survey of the statistical theory of shape. *Statistical Science*, pages 87–99.

Khan, R. A. (2013). *Detection of emotions from video in non-controlled environment.* PhD thesis, Université Claude Bernard-Lyon I.

Khan, R. A., Meyer, A., Konik, H., and Bouakaz, S. (2013). Framework for reliable, real-time facial expression recognition for low resolution images. *Pattern Recognition Letters*, 34(10):1159–1168.

Kiforenko, L. and Kraft, D. (2016). Emotion Recognition Through Body Language Using RGB-D Sensor. In *11th International Conference on Computer Vision Theory and ApplicationsComputer Vision Theory and Applications*, pages 398–405. SCITEPRESS Digital Library.

Kim, Y., Yoo, B., Kwak, Y., Choi, C., and Kim, J. (2017). Deep generative-contrastive networks for facial expression recognition. *arXiv preprint arXiv:1703.07140*.

Ko, K.-E. and Sim, K.-B. (2010). Development of a facial emotion recognition method based on combining AAM with DBN. In *Cyberworlds (CW), 2010 International Conference on*, pages 87–91. IEEE.

Kotsia, I. and Pitas, I. (2007). Facial expression recognition in image sequences using geometric deformation features and support vector machines. *IEEE Transactions on Image Processing*, 16(1):172–187.

Kumari, J., Rajesh, R., and Kumar, A. (2016). Fusion of features for the effective facial expression recognition. In *Communication and Signal Processing (ICCSP), 2016 International Conference on*, pages 0457–0461. IEEE.

Kuo, C.-M., Lai, S.-H., and Sarkis, M. (2018). A compact deep learning model for robust facial expression recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 2121–2129.

Langner, O., Dotsch, R., Bijlstra, G., Wigboldus, D. H., Hawk, S. T., and Van Knippenberg, A. (2010). Presentation and validation of the radboud faces database. *Cognition and emotion*, 24(8):1377–1388.

Learned-Miller, G. B. H. E. (2014). Labeled faces in the wild: Updates and new reporting procedures. Technical Report UM-CS-2014-003, University of Massachusetts, Amherst.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Li, S. and Deng, W. (2018). Deep facial expression recognition: A survey. *arXiv preprint arXiv:1804.08348*.

Li, S., Deng, W., and Du, J. (2017). Reliable crowdsourcing and deep locality-preserving learning for expression recognition in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2852–2861.

Liaw, A. and Wiener, M. (2002). Classification and regression by randomforest. *R news*, 2(3):18–22.

Liong, S.-T., See, J., Wong, K., and Phan, R. C.-W. (2018). Less is more: Micro-expression recognition from video using apex frame. *Signal Processing: Image Communication*, 62:82–92.

Liu, C. H. and Ward, J. (2005). Advantages of 3d methods for face recognition research in humans. In *International Workshop on Analysis and Modeling of Faces and Gestures*, pages 244–254. Springer.

Liu, M., Wang, R., Li, S., Shan, S., Huang, Z., and Chen, X. (2014a). Combining multiple kernel methods on riemannian manifold for emotion recognition in the wild. In *Proceedings of the 16th International Conference on multimodal interaction*, pages 494–501. ACM.

Liu, P., Han, S., Meng, Z., and Tong, Y. (2014b). Facial expression recognition via a boosted deep belief network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1805–1812.

Louppe, G., Wehenkel, L., Sutera, A., and Geurts, P. (2013). Understanding variable importances in forests of randomized trees. In *Advances in neural information processing systems*, pages 431–439.

Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110.

Lozano-Monasor, E., López, M. T., Fernández-Caballero, A., and Vigo-Bustos, F. (2014). Facial expression recognition from webcam based on active shape models and support vector machines. In *International Workshop on Ambient Assisted Living*, pages 147–154. Springer.

Lucey, P., Cohn, J. F., Kanade, T., Saragih, J., Ambadar, Z., and Matthews, I. (2010). The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, pages 94–101. IEEE.

Lundqvist, D., Flykt, A., and Öhman, A. (1998). The karolinska directed emotional faces (KDEF). *CD ROM from Department of Clinical Neuroscience, Psychology section, Karolinska Institutet*, 91:630.

Lyons, M., Akamatsu, S., Kamachi, M., and Gyoba, J. (1998). Coding facial expressions with gabor wavelets. In *Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on*, pages 200–205. IEEE.

Martin, C., Werner, U., and Gross, H.-M. (2008). A real-time facial expression recognition system based on active appearance models using gray images and edge images. In *Automatic Face & Gesture Recognition, 2008. FG'08. 8th IEEE International Conference on*, pages 1–6. IEEE.

Martinez, A. M. (1988). The AR face database. Technical report, Universitat Autònoma de Barcelona.

McDuff, D., Kaliouby, R., Senechal, T., Amr, M., Cohn, J., and Picard, R. (2013). Affectiva-mit facial expression dataset (am-fed): Naturalistic and spontaneous facial expressions collected. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 881–888.

Mehta, D., Sridhar, S., Sotnychenko, O., Rhodin, H., Shafiei, M., Seidel, H.-P., Xu, W., Casas, D., and Theobalt, C. (2017). Vnect: Real-time 3d human pose estimation with a single rgb camera. *ACM Transactions on Graphics (TOG)*, 36(4):44.

Meng, Z., Liu, P., Cai, J., Han, S., and Tong, Y. (2017). Identity-aware convolutional neural network for facial expression recognition. In *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*, pages 558–565. IEEE.

Merghani, W., Davison, A. K., and Yap, M. H. (2018). A review on facial micro-expressions analysis: Datasets, features and metrics. *arXiv preprint arXiv:1805.02397*.

Mishra, S. and Dhole, A. (2015). A survey on facial expression recognition techniques. *International Journal of Science and Research*, 4(4):1247–1250.

Mockus, J. (2012). *Bayesian approach to global optimization: theory and applications*, volume 37. Springer Science & Business Media.

Mollahosseini, A., Hasani, B., and Mahoor, M. H. (2017). Affectnet: A database

for facial expression, valence, and arousal computing in the wild. *arXiv preprint arXiv:1708.03985*.

Moore, S. and Bowden, R. (2011). Local binary patterns for multi-view facial expression recognition. *Computer Vision and Image Understanding*, 115(4):541–558.

Munasinghe, M. (2018). Facial expression recognition using facial landmarks and random forest classifier. In *2018 IEEE/ACIS 17th International Conference on Computer and Information Science (ICIS)*, pages 423–427. IEEE.

Nelder, J. A. and Mead, R. (1965). A simplex method for function minimization. *The Computer Journal*, 7(4):308–313.

Ojala, T., Pietikäinen, M., and Harwood, D. (1996). A comparative study of texture measures with classification based on featured distributions. *Pattern recognition*, 29(1):51–59.

O'Toole, A. J., Harms, J., Snow, S. L., Hurst, D. R., Pappas, M. R., Ayyad, J. H., and Abdi, H. (2005). A video database of moving faces and people. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(5):812–816.

Otsu, N. (1975). A threshold selection method from gray-level histograms. *Automatica*, 11(285-296):23–27.

Panchal, P., Panchal, S., and Shah, S. (2013). A comparison of SIFT and SURF. *International Journal of Innovative Research in Computer and Communication Engineering*, 1(2):323–327.

Pantic, M., Pentland, A., Nijholt, A., and Huang, T. S. (2007). Human computing and machine understanding of human behavior: A survey. In *Artifical Intelligence for Human Computing*, pages 47–71. Springer.

Pantic, M., Valstar, M., Rademaker, R., and Maat, L. (2005). Web-based database for facial expression analysis. In *2005 IEEE international conference on multimedia and Expo*, pages 5–pp. IEEE.

Rao, Q., Qu, X., Mao, Q., and Zhan, Y. (2015). Multi-pose facial expression recognition based on SURF boosting. In *Affective Computing and Intelligent Interaction (ACII), 2015 International Conference on*, pages 630–635. IEEE.

Ratliff, M. S. and Patterson, E. (2008). Emotion recognition using facial expressions with active appearance models. In *Proceedings of the Third IASTED International Conference on Human Computer Interaction,(Innsbruck, Austria)*, pages 138–143.

Santra, B. and Mukherjee, D. P. (2016a). Local dominant binary patterns for recognition of multi-view facial expressions. In *Proceedings of the Tenth Indian Conference on Computer Vision, Graphics and Image Processing*, page 25. ACM.

Santra, B. and Mukherjee, D. P. (2016b). Local saliency-inspired binary patterns for automatic recognition of multi-view facial expression. In *Image Processing (ICIP), 2016 IEEE International Conference on*, pages 624–628. IEEE.

Savitzky, A. and Golay, M. J. (1964). Smoothing and differentiation of data by simplified least squares procedures. *Analytical Chemistry*, 36(8):1627–1639.

Scherer, K. R., Bänziger, T., and Roesch, E. (2010). *A Blueprint for Affective Computing: A sourcebook and manual*. Oxford University Press.

Setyati, E., Suprapto, Y. K., and Purnomo, M. H. (2012). Facial emotional expressions recognition based on active shape model and radial basis function network. In *Computational Intelligence for Measurement Systems and Applications (CIMSA), 2012 IEEE International Conference On*, pages 41–46. IEEE.

Shan, C., Gong, S., and McOwan, P. W. (2005). Robust facial expression recognition using local binary patterns. In *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, volume 2, pages 914–917. IEEE.

Shan, C., Gong, S., and McOwan, P. W. (2009). Facial expression recognition based on local binary patterns: A comprehensive study. *Image and Vision Computing*, 27(6):803–816.

Simonoff, J. S. (2012). *Smoothing methods in statistics*. Springer Science & Business Media.

Smith, S. W. (1997). *The scientist and engineer's guide to digital signal processing*. California Technical Pub. San Diego.

Soyel, H. and Demirel, H. (2012). Localized discriminative scale invariant feature transform based facial expression recognition. *Computers & Electrical Engineering*, 38(5):1299–1309.

Sun, N., Li, Q., Huan, R., Liu, J., and Han, G. (2017). Deep spatial-temporal

feature fusion for facial expression recognition in static images. *Pattern Recognition Letters*.

Sung, J., Lee, S., and Kim, D. (2006). A real-time facial expression recognition using the STAAM. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 1, pages 275–278. IEEE.

Susskind, J. M., Anderson, A. K., and Hinton, G. E. (2010). The toronto face database. *Department of Computer Science, University of Toronto, Toronto, ON, Canada, Tech. Rep*, 3.

Suwa, M. (1978). A preliminary note on pattern recognition of human emotional expression. In *Proc. of The 4th International Joint Conference on Pattern Recognition*, pages 408–410.

Tcherkassof, A., Dupré, D., Meillon, B., Mandran, N., Dubois, M., and Adam, J.-M. (2013). DynEmo: A video database of natural facial expressions of emotions. *The International Journal of Multimedia & Its Applications*, 5(5):61–80.

T.F. Cootes, G. E. and Taylor, C. (1999). Comparing active shape models with active appearance models. In *Bmvc*, volume 99, pages 173–182. Citeseer.

Uijlings, J. R., Smeulders, A. W., and Scha, R. J. (2010). Real-time visual concept classification. *IEEE Transactions on Multimedia*, 12(7):665–681.

Van Kuilenburg, H., Wiering, M., and Den Uyl, M. (2005). A model based method for automatic facial expression recognition. In *Proceedings of the 16th European Conference on Machine Learning (ECML'05)*, pages 194–205. Springer.

Viola, P. and Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–511. IEEE.

Viola, P. and Jones, M. J. (2004). Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154.

Whitehill, J., Bartlett, M. S., and Movellan, J. R. (2013). Automatic facial expression recognition. Technical report, Oxford Univ. Press.

Wright, R., Riedel, R., Sechrest, L., Lane, R. D., and Smith, R. (2018). Sex differences in emotion recognition ability: The mediating role of trait emotional awareness. *Motivation and Emotion*, 42(1):149–160.

Wu, Y., Wang, Z., and Ji, Q. (2013). Facial feature tracking under varying facial expressions and face poses based on restricted boltzmann machines. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3452–3459.

Yang, H., Ciftci, U., and Yin, L. (2018). Facial expression recognition by de-expression residue learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2168–2177.

Yap, M. H., See, J., Hong, X., and Wang, S.-J. (2018). Facial micro-expressions grand challenge 2018 summary. In *Automatic Face & Gesture Recognition (FG 2018), 2018 13th IEEE International Conference on*, pages 675–678. IEEE.

Yin, L., Wei, X., Sun, Y., Wang, J., and Rosato, M. J. (2006). A 3d facial expression database for facial behavior research. In *7th international conference on automatic face and gesture recognition (FGR06)*, pages 211–216. IEEE.

Yu, X., Huang, J., Zhang, S., Yan, W., and Metaxas, D. N. (2013). Pose-free facial landmark fitting via optimized part mixtures and cascaded deformable shape model. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1944–1951.

Yuqian, Z. and Bertram, E. (2016). Action unit selective feature maps in deep networks for facial expression recognition. In *The 2017 International Joint Conference on Neural Networks (IJCNN 2017)*. IEEE.

Zafeiriou, S., Chrysos, G., Roussos, A., Ververas, E., Deng, J., and Trigeorgis, G. (2018). The 3d menpo facial landmark tracking challenge. *Institute of Electrical and Electronics Engineers*, pages 12503–2511.

Zhang, L., Chen, J., Lu, Y., and Wang, P. (2008). Face recognition using scale invariant feature transform and support vector machine. In *Young Computer Scientists, 2008. ICYCS 2008. The 9th International Conference for*, pages 1766–1770. IEEE.

Zhang, Z., Luo, P., Loy, C. C., and Tang, X. (2018). From facial expression recognition to interpersonal relation prediction. *International Journal of Computer Vision*, 126(5):550–569.

Zhao, G., Huang, X., Taini, M., Li, S. Z., and PietikäInen, M. (2011). Facial

expression recognition from near-infrared videos. *Image and Vision Computing*, 29(9):607–619.

Zhou, S.-R., Yin, J.-P., and Zhang, J.-M. (2013). Local Binary Pattern (LBP) and Local Phase Quantization (LBQ) based on Gabor filter for face representation. *Neurocomputing*, 116:260–264.

Zhou, Y., Antonakos, E., Alabort-i Medina, J., Roussos, A., and Zafeiriou, S. (2016). Estimating correspondences of deformable objects. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5791–5801.

Zhu, M., Xia, J., Jin, X., Yan, M., Cai, G., Yan, J., and Ning, G. (2018). Class weights random forest algorithm for processing class imbalanced medical data. *IEEE Access*, 6:4641–4652.