Scene Perception using Machine Pareidolia of Facial Expressions

by

Kenny Hong
B.Eng (Computer)/ B.Comp.Sci (Hons.)

Thesis submitted in fulfilment of the requirements for the degree of
Doctor of Philosophy in Computer Science

May, 2013
Abstract

The aim of this thesis is to pursue the question, ‘Can a machine describe the emotional response to a scene using perceived faces and facial expressions?’. To establish the relevance of this interdisciplinary thesis question literature research was conducted not only in Computer Science but also in the disciplines of Cognitive Science and Psychology. The findings from this research revealed that humans can produce an emotional response to a scene using the facial expressions of perceived faces and face-like patterns, and this is achieved at a conscious and subconscious level, as well as beyond our visual focus. These findings provide the foundations for addressing the thesis question in the realm of computer vision and machine learning. A new face model, a new face detection algorithm and a new machine learning technique based on Support Vector Machines (SVMs) was developed and an extensive experimental evaluation was conducted. The new machine learning technique called ‘Pairwise Adaptive SVM’ (aka, pa-SVM) uses a refined parameter selection process for training, and was tested using real world datasets. The result is an improved detection and classification of faces and face-like patterns, as well as their associated facial expressions. The outcome is a machine that is capable of describing the emotional response to a scene using pareidolia of faces and facial expressions.
Declaration

This thesis contains no material which has been accepted for the awards of any other degree or diploma in any university or other tertiary institution and, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made in the text. I give consent to this copy of my thesis, when deposited in the University Library¹, being made available for loan and photocopying subject to the provisions of the Copyright Act 1968.

Signature

¹ Unless an Embargo has been approved for a determined period.
Acknowledgments

First and foremost I thank my dear parents for their unconditional support in everything that I do. Next, are my supervisors, Stephan Chalup and Robert King. I am very fortunate to have their invaluable guidance and perspective through every one of our meetings. This gratitude extends to Michael Ostwald from the architecture department and Helen Thursby from the university’s writing circle. I would also like to thank language editor Rosie Linich for her efforts in each one of my publications as well as this thesis. For my experiments to be possible, I thank David Matsumoto and the team responsible for the JACFEE and JACNEUF datasets. This includes the contributors on flickr who have selflessly placed their work under a creative commons license. Thankyou.
Contents

List of Figures vii

List of Tables x

List of Acronyms xi

1 Introduction 1
1.1 Motivation and thesis question .................................. 1
1.2 The visual percept of pareidolia ................................. 2
1.3 But why faces? A cognitive science matter .................. 3
1.4 Contribution to Facial Expression Classification ............... 3
1.5 Contribution to Holistic Face Perception ....................... 4
1.6 Contribution to Component-based Face Perception .......... 4
1.7 Contribution to Support Vector Machines .................... 5
1.8 Exploring the grey area with the training data ............... 6
1.9 Summary and overview of chapters ........................... 6
1.10 Publications .................................................. 8

2 Cognitive Science and Psychology Literature 10
2.1 Regions of the brain that are active for the processing of faces
and facial expressions – and scenes .............................. 11
2.2 How does the subconscious relate to my thesis question .... 12
2.3 How our brain regions subconsciously process faces and facial
expressions .................................................. 13
2.4 Distinct brain regions for holistic and component based pro-
cessing of faces ............................................. 14
2.5 Faces can be perceived beyond the center of our visual focus
and face resemblance is ranked automatically ................ 14
2.6 Which set of emotions and facial expressions to consider? ... 15
2.7 A pareidolia mind promotes creativity and its application by
neurologists to diagnose neurological disorders ............... 16
2.8 Are machines capable of replicating and appreciating creativity? 16
2.9 Summary .................................................. 17
3 Support Vector Machines

3.1 Why I chose SVMs as my primary learning tool

3.1.1 Literature on general classification tasks

3.1.2 Literature on evaluating facial expressions

3.1.3 Literature on detecting faces

3.1.4 Literature review summary

3.2 Why I paired SVM with a Gaussian or a polynomial kernel

3.2.1 Literature on general classification tasks

3.2.2 Literature on evaluating facial expressions

3.2.3 Literature on detecting faces

3.2.4 Literature review summary

3.3 How SVM works using a two-class approach

3.4 A mathematical description of soft margin hyperplanes

3.5 Maximising the soft margins using $C$-SVM and $\nu$-SVM

3.6 Using kernels to map input space to feature space for non-linear separability

3.7 The $C$ of $C$-SVM and the $\nu$ of $\nu$-SVM

3.8 The $\gamma$ of polynomial and Gaussian kernel

3.9 One-class classification using hyperplane and hypersphere

3.10 The $\nu$ of One-Class SVM

3.11 Computational complexity in training and testing SVMs

3.12 Frameworks of two-class classifiers for multiclass classification

3.13 Summary

4 Pairwise Adaptive Support Vector Machines

4.1 Introduction

4.2 System overview

4.2.1 Dataset description

4.2.2 $k$-fold cross validation algorithm

4.2.3 Support Vector Machines (SVM) and kernels

4.2.4 The SVM and kernel parameters to optimise

4.2.5 Calculating the SVM and kernel search parameters

4.2.6 Pseudocode for standard multiclass SVM and $pa$-SVM

4.3 Results and discussion

4.3.1 $pa$-SVM mostly outperforms standard SVM for multiclass classification

4.3.2 Infeasible parameters evident for $\nu$-SVM

4.3.3 Statistical significance defined by $p$-values
4.3.4 Favourable results compared to recent studies . . . . . . 61
4.3.5 Proper selection of \((C, \gamma)\) and \((\nu, \gamma)\) values can reduce the number of support vectors . . . . . . . . . . . . . . 63
4.3.6 Computational complexity and practicality of the \(p\alpha\)-SVM 64
4.3.7 Running time . . . . . . . . . . . . . . . . . . . . . . . . 64
4.4 Summary . . . . . . . . . . . . . . . . . . . . . . . . . . . . 65

5 Facial Expression Classification 67
5.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . 69
5.2 System overview . . . . . . . . . . . . . . . . . . . . . . . . . 71
5.3 Dataset and image preprocessing . . . . . . . . . . . . . . . . 71
5.3.1 Proposed face model . . . . . . . . . . . . . . . . . . . . 73
5.3.2 Holistic approach . . . . . . . . . . . . . . . . . . . . . . 74
5.3.3 Component approach . . . . . . . . . . . . . . . . . . . . 75
5.4 Classification of facial expressions . . . . . . . . . . . . . . . 77
5.4.1 Support Vector Machines (SVM) and kernels . . . . . . 78
5.4.2 The SVM and kernel parameters to optimise . . . . . . 78
5.4.3 Proposed pairwise adaptive Multiclass SVM . . . . . . 78
5.4.4 Multiclass classification . . . . . . . . . . . . . . . . . . 78
5.4.5 Feature vectors . . . . . . . . . . . . . . . . . . . . . . . 78
5.5 Results and discussion . . . . . . . . . . . . . . . . . . . . . . 79
5.5.1 Performances under cropping and different image pre-
processing . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 79
5.5.2 Component is the better approach . . . . . . . . . . . . . 82
5.5.3 Excluding the contemptuous expression and neutral
does impact performance . . . . . . . . . . . . . . . . . . . . . 83
5.5.4 Why the contemptuous expression is considered universal 85
5.5.5 Proposed SVM mostly outperforms standard SVM . . . 85
5.5.6 Is there an optimal parameter pair? . . . . . . . . . . . 85
5.5.7 Performance dynamics of expression pairs suggest a non
equidistant expression space . . . . . . . . . . . . . . . . . . . . 87
5.5.8 Favourable results compared to other studies . . . . . . 88
5.5.9 Was the resolution of the generated face dataset optimal? 89
5.5.10 My proposed face model . . . . . . . . . . . . . . . . . . . 89
5.5.11 Future direction: intensity analysis and identification . . 91
5.6 Summary . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 92
## 6 Holistic Face Perception

6.1 Introduction ................................................................. 100
6.2 System overview .......................................................... 103
6.3 Dataset and image preprocessing ........................................ 105
   6.3.1 Dataset preprocessing ................................................ 105
   6.3.2 Image preprocessing .................................................. 107
6.4 Classification methods ................................................... 107
   6.4.1 Support Vector Machines (SVM) and kernels .................... 107
   6.4.2 The SVM and kernel parameters to optimise .................... 110
   6.4.3 $pa$-SVM framework ................................................. 110
   6.4.4 SVM and kernel search parameters ............................... 110
6.5 Null test for discarding face detectors that see faces in nothingness 110
6.6 Window search and a bounding box for estimating the true positives rate ............................................. 111
6.7 Results and discussion ................................................... 111
   6.7.1 Edge operators over histogram equalised greyscale produces artificial edges ........................................ 113
   6.7.2 Cross validation of face detection shows non-equalised greyscale as the best and most stable performer ........ 114
   6.7.3 Null test shows Canny detects faces in nothingness – which is undesirable ....................................... 114
   6.7.4 Window search shows Sobel detected faces to be robust 115
   6.7.5 Cross validation of expression classification shows equalised greyscale as the best performer ................ 117
   6.7.6 Sobel expression classifiers over Sobel detected faces shows best performance .................................... 117
   6.7.7 Were the edge operator parameter values appropriate over histogram equalised greyscale? ................ 119
   6.7.8 Was my 25% bounding box an appropriate estimation of true positives? ............................................. 125
   6.7.9 Low resolution for detection and slightly higher resolution for expression classification .......................... 126
   6.7.10 Do the decision values reflect confidence of faces detected?126
   6.7.11 A framework for finding the face detectors of different face types .................................................. 127
   6.7.12 What if the cartoon faces were trained? ......................... 127
   6.7.13 Excluding the contemptuous expression returns better results ......................................................... 131
7 Component Face Perception

7.1 Introduction .......................................................... 135
7.2 System overview ......................................................... 138
7.3 Dataset and image preprocessing ..................................... 139
7.4 Classification methods ................................................. 142
  7.4.1 Support Vector Machines (SVM) and kernels ................. 142
  7.4.2 The SVM and kernel parameters to optimise .................. 142
  7.4.3 pa-SVM framework ............................................... 144
  7.4.4 SVM and kernel search parameters ............................. 144
7.5 Component face detector ............................................. 144
  7.5.1 Computing the Component Image ............................... 145
  7.5.2 Building the Component Height Map ......................... 145
  7.5.3 Computing the Integral Rectangle ............................. 145
  7.5.4 Merge conditions for overlapping rectangles ............... 145
  7.5.5 Matching rules to collate three shape components .......... 146
7.6 Scale invariant feature vectors for semblance classifiers ...... 147
  7.6.1 Geometry feature vectors ...................................... 147
  7.6.2 Size feature vectors ........................................... 148
  7.6.3 Shape feature vectors ......................................... 148
7.7 Example walkthrough of my component face detection process 149
7.8 Results and discussion ................................................. 152
  7.8.1 Successful detection of different face types ................ 152
  7.8.2 Cases of unsuccessful detection of different face types .... 157
  7.8.3 Semblance classifiers using joined feature vectors return
        best results ...................................................... 159
  7.8.4 Eye faces are easily differentiated from brow faces using
        two-class SVMs .................................................. 159
  7.8.5 Feature vectors are inadequate for expression classification
        159
  7.8.6 Expression analysis of face components shows mouth
        with the best result ............................................. 163
  7.8.7 Excluding the contemptuous expression returns better
        results .............................................................. 163
  7.8.8 Comparison to the literature ................................... 164
7.9 Literature revealing a component induced form of pareidolia
        and the specific brain regions involved ........................ 165
7.10 Future work .......................................................... 166
7.10.1 Improve connected components labelling algorithm . . . 166
7.10.2 Improve merge conditions . . . . . . . . . . . . . . . . . 166
7.10.3 Correcting size of rectangles for aligning face compo-
ments to the trained images . . . . . . . . . . . . . . . . . 167
7.10.4 Incorporating colour information . . . . . . . . . . . . 167
7.10.5 Clustering algorithm for multiple face detection . . . 167
7.11 Summary . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 168

8 Conclusion 170
8.1 Motivation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 170
8.2 Research to support the relevance of my thesis question . . . 170
8.3 Research into the theory behind Support Vector Machines
(SVMs) and why they were chosen . . . . . . . . . . . . . . . . . 172
8.4 Contribution to multiclass classification using SVMs . . . . 173
8.5 Contribution to facial expression classification . . . . . . . . 174
8.6 Contribution to holistic face perception . . . . . . . . . . . . 175
8.7 Contribution to component face perception . . . . . . . . . . 175
8.8 What next? . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 177
8.9 Summary . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 178

Bibliography 179
List of Figures

2.1 Regions of the brain active for faces and facial expressions – and scenes .................................................. 12
3.1 An instance of a two-class problem .................................................. 26
3.2 A comparison of different kernel mapping ........................................ 33
4.1 *pa*-SVM system overview .................................................. 48
5.1 Facial expression system overview .................................................. 72
5.2 Proposed face model .................................................. 74
5.3 Sample images of the holistic and holistic action datasets. .. 76
5.4 ROC curves of neutral vs expressions .................................................. 84
5.5 Best *C* and *γ* plot of standard method .................................................. 86
5.6 Best *C* and *γ* plot of *pa*-SVM method .................................................. 87
5.7 Example grid search from the component action approach to show that *C* and *γ* are not unique to the classification rate. .. 94
5.8 Eve - neutral face. .................................................. 95
5.9 Duplicate facial actions 1: Right is described as an intense expression of left. .................................................. 95
5.10 Duplicate facial actions 2: Left is described as a determined anger while right is controlled sadness. .................................................. 95
5.11 Duplicate facial actions 3: Left is described as surprised however it is described as fear on the right if the surprised expression is held too long. .................................................. 96
5.12 A Motion Model of Eve, which has the potential to be an emotion signature to identify Eve. .................................................. 96
5.13 Intensity graphs of Eve, which show the relevant feature points significant to an expression. .................................................. 97
6.1 Holistic face perception system overview .................................................. 104
6.2 Face model used for aligning the faces .................................................. 106
6.3 Sample images of the seven universal face expressions of emotion in greyscale, Sobel and Canny .................................................. 108
6.4 The seven cartoon faces used for the window search .................................................. 109
6.5 Bounding boxes for estimating the true positives rate of a window search .............................................. 112
6.6 Artificial edges produced by histogram equalisation ........ 114
6.7 Face detector results show non-equalised greyscale as the best and most stable performer ................................. 115
6.8 Null test shows Canny sees faces in nothingness – which is undesirable .......................................................... 116
6.9 Window search shows only Sobel to be robust .................. 118
6.10 Results of expression classifiers show equalised greyscale as best performer .................................................. 119
6.11 Expression classifiers over Sobel detected faces using the 25% bound ................................................................. 120
6.12 Expression classifiers over the exact face location of the Sobel detected faces ...................................................... 121
6.13 Alternative edge operators over histogram equalised greyscale . 122
6.14 Alternate edges perform better in cross validation results . . 123
6.15 Alternate edges suffer under null test and window search of degraded cartoon faces ............................................. 124
6.16 Comparing a 50%, 25% and 10% bounding box method for reporting true positives rate ................................. 125
6.17 Decision values of Sobel detected faces do reflect the confidence of faces detected ............................................. 127
6.18 Found faces of non-human face types ............................. 128
6.19 One-class face detection and multiclass expression classification using cartoon faces ......................................... 129
6.20 The window search showed the $15 \times 15$ face detector to be the best ............................................................... 130
6.21 Cartoon expression classifiers over the detected cartoon faces of the window search ......................................... 130
6.22 Excluding the contemptuous expression returns better results . 131

7.1 Overview of the component face detection process showing the flow of the stages and states .............................. 140
7.2 Overview of the classification process showing the flow of the stages and states .................................................. 141
7.3 Sample images of the seven universal face expressions of emotion in greyscale, Cartoon and cropped face components .... 143
7.4 Face rectangles for producing scale invariant feature vectors . . 147
7.5 Example image for describing the steps of my component face detection and expression classification method. .............. 149
7.6 Component images of my example walkthrough .................. 150
7.7 A ‘pixel based’ and a ‘block based’ component height map .... 150
7.8 Finding the noise cut-off value and showing the relevant regions of the component height map ................................. 151
7.9 Top candidate faces detected and classification of expressions . 153
7.10 Different face types detected and the expressions classified 1. . 155
7.11 Different face types detected and the expressions classified 2. . 156
7.12 Example of component height maps shifting location of face components ................................................................. 157
7.13 Example of unsuccessful detection of different face types ....... 158
7.14 Semblance classifiers using joined feature vectors, with the $\nu$-SVM and the polynomial kernel returning best rates. ............ 160
7.15 Best results for separating eye faces from brow faces when feature vectors are joined. .................................................. 161
7.16 Feature vectors for semblance classifiers are inadequate for expression classification. ................................................... 162
7.17 Expression analysis of face components shows that the mouth is the best feature for discriminating between expressions. ... 163
7.18 Excluding the contemptuous expression returns better results . 164
List of Tables

4.1 Description of the datasets from the UCI Machine Learning Repository ............................................ 49
4.2 pa-SVM using C-SVM and Gaussian kernel: Best $\mu_{\text{CCR}}$ with $\mu_{\text{NSV}}$ from the max $C$ min $\gamma$ scenario. [*' marker means statistically significant] ............................................ 56
4.3 pa-SVM using C-SVM and polynomial kernel: Best $\mu_{\text{CCR}}$ with $\mu_{\text{NSV}}$ from the max $C$ min $\gamma$ scenario. [*' marker means statistically significant] ............................................ 57
4.4 pa-SVM using $\nu$-SVM and Gaussian kernel: Best $\mu_{\text{CCR}}$ with $\mu_{\text{NSV}}$ from the min $\nu$ min $\gamma$ scenario. [*' marker means statistically significant] ............................................ 58
4.5 pa-SVM using $\nu$-SVM and polynomial kernel: Best $\mu_{\text{CCR}}$ with $\mu_{\text{NSV}}$ from the min $\nu$ min $\gamma$ scenario. [*' marker means statistically significant] ............................................ 59
4.6 List of datasets with rates that are statistically significant ................................................... 60
4.7 pa-SVM results compared with recent studies ................................................................. 62
5.1 Facial actions assigned to the feature points of the face model ........................................ 77
5.2 Overview of the pa-SVM experimental results ............................................................. 80
5.3 The mean correct classification rates of holistic approaches ........................................ 81
5.4 The mean correct classification rates of component approaches ........................................ 81
5.5 Neutral vs Expressions .................................................................................................... 83
5.6 Performance dynamics of expression pairs ........................................................................ 88
5.7 Subset of action units (AU) relevant to facial movements .............................................. 90
List of Acronyms

$k$NN  $k$ Nearest Neighbour
ANN  Artificial Neural Network
ECOC  Error Correcting Output Code
ELM  Extreme Learning Machine
ICA  Independent Component Analysis
LBP  Local Binary Patterns
LDA  Linear Discriminant Analysis
PCA  Principal Component Analysis
PDFC  Positive Definite Fuzzy Classifier
SVM  Support Vector Machine
You are currently standing in the middle of a suburban area and observing a row of houses. A feeling in your gut says one house seems not to belong, or perhaps one house stands out to your liking. When asked whether you can see a face in this particular house, to some this may not be obvious at first, however to many they can almost instantly perceive areas of the house where an eye would be, and then a mouth. By taking a step back, you see another face and this time the neatly trimmed garden hedge forms a mouth. You begin to ask yourself, are these faces happy or sad? Or maybe angry? You run them by your gut feeling, and with a rush of excitement you become aware.

1.1 Motivation and thesis question

This introductory story is a simple (and hopefully effective) way to describe the motivation behind the purpose of this thesis. The story could carry the scenario of your first time looking at cars at a car yard (presumably an exciting and unwary moment) or staring into the morning clouds at the beginning of
the day, however, we are all familiar with the appearance of a house – a building that could be as simple as two windows and a door. Ask yourself, would the windows be the eyes? And what about the door? This ability to see a face where a face does not exist is a psychological phenomenon known as ‘pareidolia’.

Returning to the initial story, and asking whether you can spot a face in the row of houses and the expressions they are making is not a challenging task. This is because seeing and evaluating faces is a natural ability in all of us. My thesis question is asking whether a machine, an unnatural ubiquitous species, can see a face or a collection of faces in a scene where a face does not exist, and whether the machine is capable of assessing the expressions of these faces. In a simplified form, ‘Can a machine describe the emotional response to a scene?’

After a quick ponder of my thesis question a veteran scientific reader would immediately be drawn to the literature relating to face detection and facial expression recognition, and the use of popular learning methods such as Artificial Neural Networks (ANN) and Support Vector Machines (SVM). However, the reader should be aware that the research described in this literature aims to find human faces where a face does exist, and recognise human facial expressions. This effectively leaves a gap where the contributions to my thesis question belong.

### 1.2 The visual percept of pareidolia

Earlier I loosely introduced the term pareidolia, which is defined as the ability to perceive significance in random and vague stimuli. Our human body has a multitude of senses that can perceive a stimuli. We are all familiar with the senses of sight, hearing, taste, smell and touch. But did you know about other internal senses such as temperature, kinesthesia, pain, balance and acceleration? Pareidolia can arise through any of these senses. However, the visual percept of sight is the one most relevant to the topic of my thesis question, scene perception.

---

1. to be technically correct, it is not absolutely all of us. I discuss this further in Chapter 2 ‘Cognitive science and Psychology literature’
1.3 But why faces? A cognitive science matter

As an example exercise, for a moment, let our eyes wonder around our surroundings and rest them on an unfamiliar area where everything is still. Ask yourself, is the scene vibrant and exciting? Or is it mundane and boring? Most importantly, are you able to describe what makes the scene vibrant? Or mundane? You begin formulating questions in your head and matching them to your gut feeling – it’s the colour ... perhaps; it’s the lines ... perhaps. These questions that we raise are all answered with conscious decisions. Could some of these awakening questions of your scene be answered by your subconscious? If so, what would they be? And could a machine mimic this achievement?

But why faces? My research in the literature of Cognitive Science (which I present in Chapter 2) shows that the brain looks for faces both at the conscious and subconscious levels. This roughly suggests that we are hard-wired to look for faces whether we like it or not, and the brain does so even when faces or minimal face-like patterns are not centred in our visual focus. Then it is logical to assume that the face plays an important role in scene perception. This leads back to the example exercise of describing our scene. To some individuals, describing our gut feeling can be a daunting task – questions can lead to confusion rather than clarity. To have a machine that is capable of describing a scene allows the vagueness to become transparent – to become clear.

1.4 Contribution to Facial Expression Classification

Given a collection of perceived abstract faces from a scene, we could conclude that the definition of pareidolia has been satisfied. That is, we looked at the scene and we have drawn significance from random and vague images that a face is present, even though the face does not exist. However, I believe that the scene is significant as we are able to produce an emotional response to it, and that this emotion is drawn from facial expressions; hence returning to my thesis question. Hypothesising a formulated question from a Cognitive Scientist would be – ‘If a face is processed by the brain, does the expression of the face get processed along with it? And if so, would an emotion be associated with it?’

In pursuing the second part of my thesis question, I present the literature

---

2The reference [176] is presented in Chapter 2
on the regions of the brain that process facial expressions and emotions, in Chapter 2. In Chapter 5, I present the literature on classifying facial expressions, as well as my findings of classifying facial expressions based on a holistic and component based approach. I also present my face model used to provide the best classification rate. My face model contains a set of feature points that are positioned across the eyes, mouth and brow, which were inserted manually into all the training data.

1.5 Contribution to Holistic Face Perception

Returning our focus to the veteran scientific reader, although the aims in the literature on face detection and facial expression papers are to find human faces and recognise human facial expressions, the question begs, ‘Can they be used for finding other faces besides the human face?’ and ‘Are they able to recognise expressions other than human facial expressions?’ To answer these questions I reviewed the literature.

My investigation into face detection papers shows that they have primarily been holistic based and they have shown to be performing very well. Although they are designed to detect human faces, they still produce false positives – meaning that the machine sees a human face where none exists. The logical idea is to exploit the false positives for my purpose. I found that this is possible using image preprocessing techniques such as edge detection. Consequently, I present my holistic based approach on pareidolia in Chapter 6.

1.6 Contribution to Component-based Face Perception

But is that enough? Are holistic based methods good enough for detecting faces for my purpose? Once again, let us return to our imagined house with the two windows and a door. When you envisioned it, where were the windows and the door located? Were their locations evocative of a human face? What would happen if we move the windows further away horizontally? Or move both windows vertically higher? Would you still consider this new arrangement to be a face? In my consideration I define it to be so, and this is where holistic based methods become less effective.

I present the literature on component based face detection in Chapter 7, and by doing so I also present my component based approach on pareidolia.
1.7. Contribution to Support Vector Machines

This method was designed to detect face-like patterns where the components such as the eyes and mouth are present. The main aim was for the machine to look for faces without the mindset of where the eyes and mouth would be with respect to the human faces in the training data.

1.7 Contribution to Support Vector Machines

I have purposely left the description of the learning method used last as I wanted to describe my motivation and my thesis question clearly. My literature review on face detection and facial expression recognition papers (as well as classification tasks in general) has shown that Support Vector Machines (SVM) are very capable when compared to many other learning methods. Consequently, I use SVMs in my experiments for analysing facial expressions and detecting faces.

Support Vector Machines are designed to solve two-class problems. This requires training data that includes both the positive and negative class. However, for face detection I use the One-Class SVM where only the positive class is present – creating a virtual negative space. This reflects the situation of the proposal that we are hard-wired to look for faces at birth – as some studies in the Cognitive Science literature suggest (see Chapter 2, ‘Cognitive Science and Psychology literature’).

My contribution to Support Vector Machines was made through my experiments in facial expression classification. At the time I wanted to show the classification rates of each pair of expressions (i.e. happy vs surprise, sad vs fear, ...) to find correlations between a machine’s capability and that of human participants in psychology studies. I wanted to compare the optimal expression pair classification performance of human and machine in order to have a ‘fair’ playing field. To find the optimised performance, I searched for the best parameters for each pair of expressions and used them for multiclass classification. As a consequence of taking this step, I realised that the standard multiclass Support Vector Machine does not do this – they have the same parameters for all the binary classifiers in their multiclass framework. To explore the significance of my approach I expanded my experiments involving real world datasets. The end result is a method I called ‘Pairwise Adaptive Support Vector Machine’ (pa-SVM), which is described and discussed in Chapter 4.
1.8 Exploring the grey area with the training data

In the journey to address my thesis question, I have designed my experiments to cover several considerations in the training data. This is because when we examine a dataset, we often ask ourselves ‘Is the dataset too small?’; ‘Is there enough information present?’; ‘What about imbalanced data?’ In some cases, these questions get convoluted with ‘What normalisation should I use?’ Perhaps one of the main questions that needs closure in the field of face detection and facial expression classification would be ‘Is the image resolution optimal?’ These considerations are important, as the outcome of a classification is influenced by what is contained in the training data and how it was prepared prior to the learning process. I linger on this issue as it is a grey area, and one that I have explored and discussed in this thesis.

1.9 Summary and overview of chapters

In this chapter, I introduced my motivation and my thesis question, ‘Can a machine describe the emotional response to a scene?’ Investigating this question led to more specific questions, ‘How can a machine describe a scene?’ and ‘What features would it use?’ A small amount of literature from Cognitive Science and Psychology was presented to support why I believe that faces and facial expressions would be useful features for scene perception. Then I described my contributions for analysing facial expressions and detecting faces using Support Vector Machines (SVM), followed by my contributions to SVM. The final section described the grey areas in the training data, which have not been fully discussed in the literature, and it is my intention to explore these issues in my experiments.

The remaining chapters of this thesis are laid out as follows:

- Chapter 2 presents further literature from Cognitive Science and Psychology that is relevant to the detection of faces and pareidolia. I explore the regions of the brain that are active for faces and facial expressions, followed by the brain regions for holistic and component based processing of faces. Next, several emotion models from relevant to the detection of faces and pareidolia. of basic emotions are presented along with my reasons for why I chose Ekman’s Universal Facial Expressions of Emotion model. The chapter concludes with literature on the link between pareidolia and creativity and its application by neurologists for diagnosing
1.9. Summary and overview of chapters

neurological disorders.

- Chapter 3 presents the theory of soft margin Support Vector Machines and the relevant parameters used in practice. The $C$-SVM and $\nu$-SVM are described using a two-class approach, followed by the different kernels for mapping data from input space to feature space – i.e. linear, polynomial and Gaussian kernel. Then One-Class SVMs are presented using the hyperplane and hypersphere approach. The chapter concludes with the frameworks of two-class classifiers for multiclass classification.

- Chapter 4 presents my Pairwise Adaptive Support Vector Machines (p-aSVM) which are a refined multiclass SVM. Real world datasets from the UCI Machine Learning Repository were used to show the significance of my method, which was originally conceived from my facial expression study.

- Chapter 5 presents Facial Expression Classification. This chapter explores and compares a holistic and component based approach to classifying facial expressions. Ekman’s *Universal Facial Expressions of Emotion* are further discussed, and my face model, which is used in subsequent chapters, is presented.

- Chapter 6 presents Holistic Face Perception. The literature on traditional holistic approaches is presented here. I devised an empirical platform to evaluate the application of a holistic approach for perceiving faces in a scene. For the training data, I used my face model to create cartoon faces that are diffused into a number of minimal face-like patterns. Several preprocessing techniques were evaluated to show which is suitable for perceiving abstract faces holistically, and for the classification of expressions.

- Chapter 7 presents Component Based Face Perception. A separate chapter was required to show the distinctions when compared to a holistic approach. The literature on component based approaches is presented here. Because the existing methods available do not meet my requirements for a suitable component face detector I developed a new method. In this chapter, I show the potential of the possible faces that can be perceived with my component face detector, and evaluate the performance of classifying expression using face components (i.e. brows, eyes
1.10. Publications

and mouth). Once again, my face model was used to extract face components from the cartoon face dataset used in the previous chapter.

• Chapter 8 presents the final argument of my thesis question – its relevance and the contributions that were made in pursuing my thesis question. My final words introduce my last question, ‘What is the next step?’

You may notice that this introduction presented Support Vector Machines last, however it was necessary to put the chapters on Support Vector Machines early as they are referred to in the remaining chapters.

1.10 Publications

The following publications have been written and submitted within the timescale of this thesis.

Journals


Conference Proceedings

• Kenny Hong, Stephan K. Chalup, Robert A.R. King, and Michael J. Ostwald. Scene perception using pareidolia of faces and expressions of emotion. In IEEE Symposium on Computational Intelligence for Creativity and Affective Computing (CICAC), 2013. Manuscript accepted for publication.

• Kenny Hong, Stephan K. Chalup, and Robert A.R. King. A component based approach for classifying the seven universal facial expressions of emotion. In IEEE Symposium on Computational Intelligence for Creativity and Affective Computing (CICAC), 2013. Manuscript accepted for publication.
1.10. Publications

- Aaron S.W. Wong, Steven Nicklin, Kenny Hong, Stephan K. Chalup, and Peter Walla. Robot emotions generated and modulated by visual features of the environment. In IEEE Symposium on Computational Intelligence for Creativity and Affective Computing (CICAC), 2013. Manuscript accepted for publication.


In the previous chapter I presented my thesis question ‘Can a machine describe the emotional response to a scene using perceived faces and facial expressions?’ Following this, I lightly touched on the reasons why faces and their expressions should be used, and I emphasised that the significance of a scene goes beyond the perception of faces from random and vague images contained within it. I propose that the significance of a scene is the result of our emotional response to it – and this emotion is drawn from the facial expressions of perceived faces.
2.1 Regions of the brain that are active for the processing of faces and facial expressions – and scenes

In this chapter I review the literature from Cognitive Science and Psychology to support my thesis question as to why faces (and their expressions) are significant for scene perception. On the Cognitive Science side, I begin by establishing the regions of our brain that are active for faces and facial expressions and scenes, followed by why the subconscious processes of these regions are more relevant to my thesis question than are the conscious processes of face perception. I then present the regions of the brain involved in holistic and component based processing of faces. Finally, I present the literature that suggests that faces can be perceived in minimal face-like patterns even when they are not centred in our visual focus; followed by the region of the brain which ranks the likeness of face patterns to genuine faces. On the Psychology side, I present the literature on basic emotions and the reasons for why I chose the Universal Facial Expressions of Emotion as my emotional model. This chapter concludes with a discussion of the literature on the link between pareidolia and creativity and its application by neurologists in diagnosing neurological disorders.

2.1 Regions of the brain that are active for the processing of faces and facial expressions – and scenes

The human brain is a powerful cognitive machine, which continuously processes signals from numerous stimuli. My first question (relevant to my thesis topic) is ‘Are there specific areas of the brain that are active for faces and facial expressions?’ From the literature review, I discovered that cognitive neuroscientists have used a variety of methods to answer this question. One of their main methods is functional Magnetic Resonance Imaging (fMRI). This technique is useful in revealing the neural pathways of the brain, and from their studies they have isolated the main regions of the brain involved in the processes relevant to face perception, facial expression processing and scene perception. These regions are – the inferior occipital and fusiform gyri (responsible for face perception), and the amygdala (which is responsible for processing emotions); they lie close to each other as shown in Figure 2.1. The figure also shows the parahippocampal gyrus – a study has used functional magnetic resonance imaging (fMRI) to show that this region is active for processing scenes [55]. A supporting study that analysed fMRI results of brain regions during viewing of emotional scenes and emotional faces shows that the
activation of the amygdala has the greatest overlap [187], and the activation is stronger for facial expressions than for scenes [86, 195]. In subsequent sections I further discuss the three regions (the inferior occipital and fusiform gyri and the amygdala) involved in the processing of faces and facial expressions.

Figure 2.1: The regions of the brain that are active for faces and facial expressions – and scenes. Each region has its own responsibilities. The inferior occipital and fusiform gyri are responsible for face perception, the amygdala is responsible for processing emotions, and the parahippocampal gyrus is responsible for scene perception. This illustration is based on images from [22, 156].

2.2 How does the subconscious relate to my thesis question

Before I dive into the specifics of these brain regions and their activation by faces and facial expressions, I want the reader to bear in mind the meaning of the subconscious and how it plays into my thesis question.

Lets visualise our subconscious to be a pool of knowledge, which perpetually gathers information from various stimuli in our surrounding environment. Our pool is weirdly shaped, with varying depths at the bottom and overflow-
2.3. How our brain regions subconsciously process faces and facial expressions

The information that overflows is lost and can no longer be retrieved. I denote this loss to be the unconscious, which is the extreme state of the subconscious. A slow moving water wheel hovers around this pool and each time it dips into the pool’s surface the information is carried to our conscious, however to get information from the bottom requires specialised methods.

The slow moving water wheel contrasts to the rapid flow of information into this pool. It is this speed that many researchers have used as one of the main cues to when a subconscious process has occurred [84, 112]. You ask, ‘How does this relate to my thesis question?’ When we perceive a scene, we may lack the ability to fully describe our emotional response to it as the answers in our conscious mind are limited by our ability to question our subconscious. The rapid rate at which the subconscious is processed means that it is rich in information, and the ability for a machine to relay this information allows our emotional response to a scene to become transparent – to become clear.

2.3 How our brain regions subconsciously process faces and facial expressions

From this understanding of the subconscious, ‘How do our brain regions subconsciously process faces and facial expressions?’ In the literature review I found that faces are processed subconsciously by the early activation of the inferior occipital and fusiform gyri (using magnetoencephalography (MEG) readings) when human subjects are presented with images of objects, faces and face-like objects (i.e. pareidolia) [84], and this rapid activation is seen across the subcortical pathway. The rapid activation of this subcortical pathway has been linked to the activation of the amygdala, which have been shown to be involved in the processing of emotions from facial expressions both in newborns [112] and adults [2, 191, 199] (for further reading [187, 202]). In particular, it is the right amygdala which is more involved in a subconscious process while the left amygdala is more involved in a conscious process [162]. The study of newborns recognising faces and facial expressions [112] implies that the brain may be hard-wired to look for faces from birth, and the use of schematic face-like patterns in the study suggests that newborns are capable of perceiving faces from minimal face-like patterns (once again, pareidolia). The short and direct path of the subcortical face route is viewed by some as the ‘low-road’, while others believe that the subconscious decoding of faces and
2.4 Distinct brain regions for holistic and component based processing of faces

I have now established that there are brain regions that are active for subconsciously processing faces and facial expressions – this being the inferior occipital and fusiform gyri for face perception and the amygdala for processing of emotions. Because of my curiosity, I questioned why there are two brain regions required for face perception. However, before I go any further (and to be technically in line with the neuroscience terminology), the specific region of the fusiform gyrus for processing faces is known by cognitive neuroscientists as the ‘fusiform face area (FFA)’ and the specific region of the inferior occipital gyrus for processing faces is known as the ‘occipital face area (OFA)’ [70].

Returning to my curiosity, the question is ‘Do the occipital face area (OFA) and fusiform face area (FFA) have specific roles for processing faces?’ There have been many studies that have shown the FFA processes faces holistically [234] and argue that it is also sensitive to face components [6, 108]. Further studies have supported the theory of the FFA processing faces holistically [234] (using fMRI with multi-voxel pattern analysis). However, it is the OFA that is more sensitive to processing face components [6] (using fMRI with blood-oxygen-level dependence responses). In addition, the OFA has been shown to give preference to face components, primarily the eyes, followed by the mouth and then the nose [6]. There are debates as to whether a hierarchical order between the OFA and FFA exists – currently a more complex interplay between the OFA and FFA is assumed [7].

2.5 Faces can be perceived beyond the center of our visual focus and face semblance is ranked automatically

The distinction between the brain regions involved in processing faces by components and holistically were important considerations when I approached my thesis question in relation to face detection methods for my machine. Pareidolia is an instance of seeing a face by the sum of its parts, but is it correct
2.6. Which set of emotions and facial expressions to consider?

A study using noisy images and fMRI suggest that faces can be perceived with minimal face-like patterns even when these features are not centred in our visual focus [176]. In addition, the ranking of face likeness (from images unlike faces to genuine face images) has been shown to engage the left fusiform face area (FFA), then categorical face/non-face judgements occur on the right [148]. These findings have a direct influence on the realisation of how my machine would rank perceived faces as meaningful. The ability of seeing faces that are not centred in our visual focus means that a machine should perceive faces in different areas of an image. Similar to how the left fusiform would rank the face likeness to that of a genuine face, I foresee my machine would replicate this phenomena using machine learning techniques.

2.6 Which set of emotions and facial expressions to consider?

So far I have presented the literature to establish the regions of the brain that are active for faces and facial expressions, and how the brain can perceive faces beyond the centre of our visual focus and ranks their likeness to genuine faces automatically. I return the reader to my thesis question – I have described that it is not just the perceived faces that are significant in a scene, but the emotion produced from them. I have presented relevant studies that show that the brain region known as the amygdala is responsible for processing emotions and is activated by faces and facial expressions. The important question here is, ‘Which set of emotions and facial expressions to consider?’

From the literature review, there are four prominent theories of basic emotions, and the advocates of each have presented their arguments to the relevance of their emotional model – Ekman and Cordaro [53]; Izard [106]; Levenson [126]; and Panksepp and Watt [161]. The similarities and differences of each theorist are presented and discussed by Tracy and Randles [208]. Examining each of the emotional models described, I find that Ekman and Cordaro’s model is the most relevant for my thesis question. This is because I believe that the emotions of Interest/Seeking (Izard, Levenson, and Panksepp and Watt), Lust/Love (Panksepp and Watt, and Levenson), Care/Relief (Panksepp and Watt, and Levenson), and Pride (Tracy and Randles) are difficult to recognise through facial expressions, when compared to Ekman and Cordaro’s model of the seven emotions – Happy, Sad, Fear, Anger, Disgust, Contempt and Sur-
2.7 A pareidolia mind promotes creativity and its application by neurologists to diagnose neurological disorders

A pareidolia mind promotes creativity and its application by neurologists to diagnose neurological disorders. Further, Ekman and Cordaro have shown that their facial expressions of emotion are recognised universally \[54\]. So with regard to my thesis question, if my machine is able to recognise the universal facial expressions, the emotions produced should also be universally recognised.

2.8 Are machines capable of replicating and appreciating creativity?

The literature on the mind and the link between pareidolia and creativity raises an interesting question – if a machine is capable of seeing meaning from meaningless data, is this machine replicating the creative process of an individual? Would you consider this machine to be creative? Further on the literature, a study has presented arguments that the enduring popularity of
2.9 Summary

In the previous chapter I introduced my thesis question, ‘Can a machine describe the emotional response to a scene using perceived faces and facial expressions?’ By breaking down this question, I ask, ‘Why faces?’ ‘What is it that makes the face a unique feature for scene perception?’ ‘What about facial expressions? What purpose do they serve?’ – addressing these questions is one of the main purposes of this chapter.

My initial step presented literature from Cognitive Science to show the reader that there are specific regions of the brain that are active for faces and facial expressions. These regions are the inferior occipital and fusiform gyri for processing face perception and the amygdala for processing emotions. This finding shows that faces and their expressions are processed whether we like it or not. It is a feature that is always switched on and because of this it is perhaps a useful feature to use in the analysis of scene perception.

Before diving any further into the chapter, I presented to the reader the reasons for why the subconscious is relevant to my question. From the literature review, researchers have used the rate at which the brain processes information to separate the subconscious from the conscious processes. This finding shows that the subconscious is rich in information and that we are not able to fully interpret a scene immediately because our consciousness is inherently slow.

I then looked at the reasons why there are two regions of the brain for face perception, and from my investigation I found that the inferior occipital gyrus is specialised in component based face processing and the fusiform gyrus processes faces holistically. This finding contrasts the two main footprints in face detection at present (which I explore further in their respective chapters – Chapter 7 on component based face detection and Chapter 6 on holistic based face detection).

My arguments then flow towards pareidolia where I present a study that
suggests that faces can be perceived from minimal face-like patterns even when
these features are not centred in our visual focus. This finding is relevant
as it justifies that a machine does not need to exhibit any special focus on
searching for faces at or near the centre of an image (under the assumption
that the centre of the image is our centre of visual focus). In addition, I
presented a study showing that the ranking of face-likeness to genuine faces is
modulated by the left fusiform gyrus – I foresee my machine would replicate
this phenomenon.

Up to this point in the chapter, my reason for focusing on facial expres-
sions is the emotion produced by them (which I consider the significance of
a scene). This raises the question – ‘Which set of emotions and facial ex-
pressions to consider?’ Several models of emotion were presented and I chose
Ekman and Cordaro’s model of seven emotions (Happy, Sad, Fear, Anger,
Disgust, Contempt and Surprise) because the emotions could be robustly
recognised through facial expressions and the authors have shown it to be
universally recognised. This was an important point because if my machine
is able to recognise the universal facial expressions then the emotions they
produce should be recognised universally.

Further along the topic of pareidolia, the literature review shows that it
is an ability that exists in the general population, and that it is a specific
example of Apophenia – the phenomenon of seeing connections in random and
unrelated events. A trail of literature is then presented to show the linking
of Apophenia to schizotypy and positive schizotypy to creativity. This finding
leads us to the question of whether a machine that is created to perceive a scene
through pareidolia is replicating the creative process of an individual, and
would it be considered creative? Tying in with the issue of creativity, the final
literature review showed that the enduring popularity of historical artworks is
due to our propensity for seeing and appreciating the visual creativity of faces
and animals – leaving a question to the reader, ‘Could a machine be used to
appreciate creative artworks? And would it rank creativity based on faces and
expressions?’

Concluding the chapter summary, the materials that have been presented
so far ground the relevance and importance of my thesis question, ‘Can a
machine describe the emotional response to a scene using perceived faces and
facial expressions?’ What follows next are the materials from machine learn-
ing and image processing techniques – these are the main constructs leading
towards the realisation of my perceptive machine.
Chapter 3

Support Vector Machines

The contents of this chapter have been published in [96, 97].

Contents

3.1 Why I chose SVMs as my primary learning tool .... 20
  3.1.1 Literature on general classification tasks .... 21
  3.1.2 Literature on evaluating facial expressions .... 22
  3.1.3 Literature on detecting faces .... 23
  3.1.4 Literature review summary .... 23

3.2 Why I paired SVM with a Gaussian or a polynomial
    kernel .... 24
  3.2.1 Literature on general classification tasks .... 24
  3.2.2 Literature on evaluating facial expressions .... 24
  3.2.3 Literature on detecting faces .... 25
  3.2.4 Literature review summary .... 25

3.3 How SVM works using a two-class approach .... 25

3.4 A mathematical description of soft margin hyperplanes 27

3.5 Maximising the soft margins using C-SVM and ν-SVM 28

3.6 Using kernels to map input space to feature space for
    non-linear separability .... 31

3.7 The C of C-SVM and the ν of ν-SVM .... 32

3.8 The γ of polynomial and Gaussian kernel .... 34

3.9 One-class classification using hyperplane and hyper-
    sphere .... 35

3.10 The ν of One-Class SVM .... 38

3.11 Frameworks of two-class classifiers for multiclass clas-
    sification .... 39

3.12 Summary .... 40
In Chapter 1, I presented my thesis question ‘Can a machine describe the emotional response to a scene using perceived faces and facial expressions?’ The previous chapter presented the literature from Cognitive Science and Psychology as to why faces and facial expressions are the appropriate features for describing a scene. Then I presented a number of emotional models from prominent theorists of basic emotions, and my reasons for choosing the Universal Facial Expressions of Emotion. My final remarks from that chapter described how our ability to see faces from minimal face-like patterns links pareidolia to the creative process of an individual. To fully explore my thesis question, this and successive chapters present literature from my discipline in the field of machine learning and image preprocessing techniques.

The primary purpose of this chapter is to get the reader familiar with the relevant background on Support Vector Machines (SVMs) and the parameters that are used in subsequent chapters of this thesis. Firstly, I present a literature review to support my decision to use SVMs as my primary learning tool. Then I describe how SVMs work in the tradition of a two-class approach. As my main focus is on real world datasets, I present the theory of SVM based on soft margins and how they can be maximised using two common methods – known as $C$-SVM and $\nu$-SVM. I then discuss how separability of non-linear data can be achieved by the use of kernels for mapping the input space to feature space. Afterwards I present the parameters that are used in the two SVM types described ($C$-SVM and $\nu$-SVM), along with the Gaussian and polynomial kernel parameter $\gamma$. Prior to concluding the chapter I present One-Class SVMs where I explain the hyperplane and hypersphere approach as well as their associated parameters used in practice. The final sections present the computational complexity in training and testing SVMs and the scenarios of combining two-class classifiers for classifying data with multiple classes – multiclass classification.

3.1 Why I chose SVMs as my primary learning tool

To fully explore my thesis question the use of machine learning tools is inevitable. This section presents the literature review that explains why I chose SVM as my preferred method. I divide the literature into general classification tasks (relevant to Chapter 4 for my ‘Pairwise Adaptive SVM’), evaluating facial expressions (relevant to Chapters 5-7 for analysing facial expressions);
3.1. Why I chose SVMs as my primary learning tool

and lastly, detecting faces (relevant to Chapters 6-7 for face perception).

The studies quoted in these sections serve three purposes. The first is to show the reader the large number of machine learning methods available. The second is to show the broad applications possible using machine learning algorithms, seen in the number of datasets and test scenarios conducted in each quoted study. The third is to compare the classification rates of these quoted studies to new methods in the respective chapters to show why my contributions are effective. I discuss cases where their results return a higher classification rate than my results.

3.1.1 Literature on general classification tasks

For the literature on general classification tasks I examined studies that used datasets from the UCI Machine Learning Repository [64]; this is because they are real world instances that have been used extensively for evaluating and benchmarking new machine learning algorithms. I begin by presenting these studies from recent years (from 2010 to 2013) to give the reader a perspective on the large number of methods that have been considered for this repository. Then I focus my arguments on selected studies that have experimented SVMs with other classifiers; this is to support and further conclude the use of SVM as my preferred method.

A large number of machine learning algorithms have been explored over recent years (from 2010 to 2013) using datasets from the UCI Machine Learning Repository. They include Artificial Neural Networks (ANN) (i.e. Multilayer Perceptrons) [82, 101, 137, 159, 230], k Nearest Neighbors (kNN) [9, 68, 197], Distance Weighted k Neighbor [101], Linear Discriminant Analysis (LDA) [29, 233], Linearly Bounding Discriminant Analysis [152], fuzzy classifier using non-dominance criterion [62], Positive Definite Fuzzy Classifier (PDFC) [68], Ripper [68], C4.5 [68, 230], Multiresponse Support Vector Regression [29], Parzen window classifier [233], Error Correcting Output Code classifier (ECOC) [59], Recursive ECOC classifier [203], online ECOC [58], multiclass multi-kernel relevance vector machines [171], and vector-valued regularised kernel function approximation [72]. In many instances SVMs worked in conjunction with other classifiers such as decision trees [138], ANN [230], and fuzzy logic [68] methods. In particular, some of the best results of selected UCI datasets using ECOC are paired with SVMs [59, 203].

In the studies mentioned so far, those which compared SVM with other classifiers have shown SVM to be one of the methods that returned a high cor-
rect classification rate [29, 59, 68, 137, 230]. Examining these studies closely, [68] used two ranking schemes to show that SVM came second (behind PDFC) using accuracy rate and third (first was C4.5 then PDFC) using Cohen’s kappa metric [34]; this is against the 1 and 3 Nearest Neighbors, and Ripper; [59] showed that SVM with ECOC returned a higher rate than Gentle AdaBoost with ECOC for the majority of their selected UCI datasets; [29] showed that SVM and variants of SVM returned better testing accuracy than Fisher’s Linear Discriminant Analysis; [230] showed that SVM outperformed C4.5 in many of the experiments – which also combined SVM and C4.5 with Principal Component Analysis (PCA) and least angle regression to their algorithm learning based ANN; and lastly, [137] showed that Extreme Learning Machines (ELM) (which is based on a single hidden layer feed forward ANN) can generalise as well as SVM but this only applies to large sample sizes.

The literature presented for general classification tasks suggests that SVM is still one of the most popular and capable performing classifiers to date. The use of SVM as my preferred option is further substantiated by recent literature that compared SVM with other classification methods. Next, I explore the literature specific to evaluating facial expressions.

3.1.2 Literature on evaluating facial expressions

For evaluating facial expressions SVMs have been the most studied method [17, 33, 43, 110, 139, 196, 236]. However, there are other methods that have been used to classify facial expressions; these include ANN [12, 39, 125], kNN [43, 61, 196], Naïve Bayes [43], Bayesian Networks [43] and decision trees [43]. Dimensionality reduction methods used prior to classification include Principal Component Analysis (PCA) [5, 12, 61, 157], LDA [142] and Independent Component Analysis (ICA) [33, 61]. Meta algorithms such as Adaptive Boosting (AdaBoost) are often used to improve performance [117, 196, 227, 236]. Further, the concatenation of these methods has been tried and tested; for example PCA with ANN [12], ANN with SVM [125], PCA/ICA with kNN [61] and LDA with kNN [196].

From the studies mentioned, only a small number have compared SVM with other classifiers [43, 196]. Examining these studies closely, [196] showed that PCA projection of Local Binary Patterns (LBP) with SVM returned a higher classification rate than LDA combined with Nearest Neighbour (kNN), and [43] showed that SVM returned the highest classification rates followed by kNN, Naïve Bayes, Bayesian Networks and C4.5 decision tree.
3.1. Why I chose SVMs as my primary learning tool

The literature presented for evaluating facial expressions still suggests that SVM is one of the most capable performing classifiers for this task. Next, I explore literature specific to face detection.

3.1.3 Literature on detecting faces

For detecting faces, the literature review of face detection surveys [121, 226, 228, 232] showed that there are an abundance of methods available. The most popular face detection methods use Haar-like features in a cascade of weak classifiers with Adaptive Boosting (a.k.a AdaBoost) [215] and its popularity is due to its computational speed. However, the two classifiers that are common across all the surveys were ANN and SVM [121, 226, 228, 232]. A direct comparison between ANN and SVM was limited to only one survey [226] which indicated that ANN was better because some of the results for SVM were not applicable. However, this missing data was fulfilled by [90] which shows that SVMs returned a higher classification rate for face detection. A similar conclusion was drawn in a recent study [63] which uses a Particle Swarm Optimisation (PSO) to train an ANN.

From my Cognitive Science and Psychology literature review in Chapter 2, the studies reveal that a neural basis exists for perceiving faces and facial expressions, and is defined within specific brain regions. This suggests that our brains are hard-wired to look for faces from birth. From this suggestion I hypothesised that the encoded knowledge would primarily be influenced by face data (of the positive class) rather than non-face data (of the negative class). My hypothesis directs our attention to one-class classification methods. Two comparative studies on general tasks have shown that one-class classification using SVM has outperformed its competitors [73, 239]. In one study where the one-class SVM did not achieve the overall best performance, it did show more robustness than other methods [174]. Narrowing my review for face detection, there is literature of one-class SVMs being used [24, 111], however, those studies are specific to finding human faces.

3.1.4 Literature review summary

The literature review on general classification tasks, facial expressions and face detection studies shows that SVMs are robust in multiclass and one-class scenarios. Its robust application to facial expressions and face detection is one reason why I have chosen SVM as my preferred method. In subsequent
3.2 Why I paired SVM with a Gaussian or a polynomial kernel

chapters I compare my results with the studies mentioned so far, and discuss further the cases where their results returned a higher classification than mine. Next, I examine my presented facial expressions and face detection studies to discuss my decision on which SVM kernels to use.

3.2 Why I paired SVM with a Gaussian or a polynomial kernel

SVMs by themselves are suitable for linearly separable data [211]. When SVMs are paired with a kernel, they become viable for non-linearly separable data. In this section I present the reasons why I have chosen the Gaussian and polynomial kernel to pair with SVMs for my studies. I revisit the literature presented in the previous section, and using the same structure I present the chosen kernels by those studies which I have segregated into general classification tasks, evaluating facial expressions and detecting faces.

3.2.1 Literature on general classification tasks

For general classification tasks, the Gaussian kernel has been widely used [27, 57–59, 72, 80, 82, 130, 138, 203, 212, 233]. Among those studies, some have experimented with the linear kernel [57, 59, 155, 214, 233] and very few with the polynomial kernel [68, 171]. The studies using the linear kernel showed that the Gaussian kernel classified better [57, 59, 233], and the studies using the polynomial kernel had one using a degree of 1 (i.e. linear) [68] and the other a degree of 2 [28]. Like the studies in Section 3.1.1, the cited studies in this subsection use datasets from the UCI Machine Learning Repository. Since the former study using a polynomial kernel with a degree of 1 is a modified linear kernel, I follow the latter study where they used a low degree of 2 and a constant of 1 in the experiments throughout this thesis.

3.2.2 Literature on evaluating facial expressions

For evaluating facial expressions, several facial expression studies have reported higher results using Gaussian followed by the polynomial and then the linear kernel [13, 114, 196]; while others have used the Gaussian [81, 166, 220], polynomial [217] and linear kernel [33] as their default choice. In fewer cases the polynomial [125, 236] and linear kernel [135] were reported to perform better than the Gaussian kernel.
3.3. How SVM works using a two-class approach

3.2.3 Literature on detecting faces

For detecting faces, the Gaussian kernel is the default choice [24, 111]; however, [24] use linear kernels at the early stages of the cascade to filter out the regions of interest quickly.

3.2.4 Literature review summary

The literature presented shows that the most commonly used kernel is the Gaussian followed by the linear and the polynomial. It also shows that the linear kernel performs worst of the three in most cases. I included the polynomial kernel to provide an alternative to Gaussian, however, fixing the degree to 2 for the polynomial, similar to [28]. A degree higher than 2 I expect would incur a longer training time than a Gaussian kernel. One reason for this is that an increase in the degree of the polynomial introduces more support vectors to fit a decision plane closer to the data points. This is similar to increasing the $\gamma$ of the Gaussian kernel. I discuss this issue further in this chapter, particularly in Section 3.8.

3.3 How SVM works using a two-class approach

In accord with the traditional approach for explaining how SVM works, let’s start with a two-class problem, which I denote as the positive and negative class.

Let us visualise an instance of the positive and negative class datapoints being graphed onto a two dimensional plane as shown in Figure 3.1. From this figure, three lines can be seen separating the two-classes. The middle line is known as the separating hyperplane which linearly divides the datapoints of the two-classes. The datapoints nearest to the separating hyperplane are known as support vectors, and the distance between the support vectors of the classes perpendicular to the separating hyperplane is known as the margin of the separating hyperplane. This margin of the separating hyperplane is often described by the hyperplanes created using the support vectors (shown by the dashed outer lines), which is parallel to the separating hyperplane. A primary aim of SVM is to find a separating hyperplane where the support vectors of the two-classes are as far apart as possible, i.e. maximising the margin – this is because by having a large margin it becomes easier to determine the class when a new datapoint is presented – a procedure known as generalisation.
3.3. How SVM works using a two-class approach

Figure 3.1: An instance of a two-class problem. The middle line is the separating hyperplane which linearly divides the datapoints of the two-classes, and the datapoints that are closest to the separating hyperplane are known as support vectors. A primary aim of SVM is to find the separating hyperplane where the support vectors of the two-classes are as far apart as possible. The slack variables $\xi_i$ are introduced to handle noise for real world datasets – creating a soft margin.
3.4. A mathematical description of soft margin hyperplanes

In an ideal scenario datapoints of each class would lie neatly in their separated space so that a linear separation is clear. In other words there should not exist any other datapoints within the margin of the separating hyperplane – creating a tight separating margin known as a *hard margin*. However, in real world datasets a small number of datapoints do exist within this margin (called noise) and is described by the use of slack variables (shown by $\xi_p$ and $\xi_n$) – creating a loose separating margin known as a *soft margin*.

### 3.4 A mathematical description of soft margin hyperplanes

In the following, I will refer back to Figure 3.1 as we revisit the description of how SVM works using mathematical notation in the context of soft margin separation.

Let’s begin by formulating a description of the margin between the hyperplanes created using the support vectors of the classes by first defining the hyperplane of the positive class to be:

$$w \cdot x_p + b = 1$$  \hspace{1cm} (3.1)

and the hyperplane of the negative class to be:

$$w \cdot x_n + b = -1$$  \hspace{1cm} (3.2)

where $b$ is the *bias* or the offset of the hyperplane from the origin in the input space, $x_p$ and $x_n$ are the datapoints on the hyperplane of the respective class, and $w$ is the normal to the hyperplane which defines its orientation.

The separating hyperplane sitting at midpoint would then be:

$$w \cdot x + b = 0$$

To get the margin of the separating hyperplane let’s subtract the negative plane (3.2) from the positive plane (3.1), and project the difference vector onto the separating hyperplane defined by the normal $w$:

$$x_p - x_n = \frac{2}{||w||}$$  \hspace{1cm} (3.3)
3.5 Maximising the soft margins using \( C\)-SVM and \( \nu\)-SVM

As the aim is to maximise the margin (3.3) let’s minimise \(|w|\). This is equivalent to minimising an objective function:

\[
\frac{||w||^2}{2} \tag{3.4}
\]

Since soft margins are used then slack variables are introduced. The new objective function to minimise becomes ([36]):

\[
\min_{w,\xi} \left[ \frac{1}{2} w \cdot w + C \sum_{i=1}^{m} \xi_i \right] \tag{3.5}
\]

subject to the constraints:

\[
y_i(\text{\( w \) \cdot \text{x}_i + b}) \geq 1 - \xi_i \quad \forall i \tag{3.6}
\]

where \( \xi_i \) are the slack variables controlled by the parameter \( C \); and if \( \xi_i > 0 \) we have a margin error.

This is then a constraint optimisation problem and the reason why most literature square \(|w|\) and \( \frac{1}{2} \) is because (3.5) and (3.6) become a Lagrange function which we are to minimise, and in the differentiation process the \( \frac{1}{2} \) cancels out. In addition, because SVM is realised using non-linear programming, there are first order necessary conditions that are used so that optimality can be met – they are known as Karush-Kuhn-Tucker (KKT) conditions.

The primal form of the Lagrange function with the Lagrange multipliers \((\alpha, r)\) satisfying (3.5)-(3.6) is:

\[
L(w, \xi, b, \alpha) = \frac{1}{2} (w \cdot w) + C \sum_{i=1}^{m} \xi_i - \sum_{i=1}^{m} \alpha_i (y_i (w \cdot x_i + b) - 1 + \xi_i) - \sum_{i=1}^{m} r_i \xi_i \tag{3.7}
\]

where \( \alpha_i \geq 0 \) to satisfy (3.6), and \( r_i \geq 0 \) to satisfy \( \xi_i \geq 0 \).

The KKT conditions are:

\[
r_i \xi_i = 0 \tag{3.8}
\]

\[
\alpha_i (y_i (w \cdot x_i + b) - 1 + \xi_i) = 0 \tag{3.9}
\]
3.5. Maximising the soft margins using \( C \)-SVM and \( \nu \)-SVM

To get the minimum, let’s take the derivatives of \( L \) with respect to \( w \), \( b \), \( \xi \) and set them to zero:

\[
\frac{\partial L}{\partial w} = w - \sum_{i=1}^{m} \alpha_i y_i x_i = 0 \quad (3.10)
\]

\[
\frac{\partial L}{\partial b} = \sum_{i=1}^{m} \alpha_i y_i = 0 \quad (3.11)
\]

\[
\frac{\partial L}{\partial \xi_i} = C - \alpha_i - r_i = 0 \quad (3.12)
\]

To get the dual form let’s substitute (3.10)-(3.12) into (3.7):

\[
\max_{\alpha} \left[ \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) \right] \quad (3.13)
\]

which must be maximised with respect to \( \alpha_i \), subject to the constraints:

\[
\alpha_i \geq 0 \quad (3.14)
\]

\[
\sum_{i=1}^{m} \alpha_i y_i = 0 \quad (3.15)
\]

\[
0 \leq \alpha_i \leq C \quad (3.16)
\]

The first two constraints arise from (3.7) and (3.11) respectively. The third constraint (3.16) is a result of the KKT condition \( r_i \xi_i = 0 \) (3.8) with the Lagrange derivative \( C - \alpha_i - r_i = 0 \) (3.12), i.e. assuming we have a margin error \( (\xi_i > 0) \), then (3.8) implies \( r_i = 0 \), and using (3.12) we get \( 0 \leq \alpha_i \leq C \) (known as a box constraint).

The approach described is one method of maximising the soft margin and the realisation of this SVM is known as a \( C \)-SVM where the \( C \) is from (3.5).

Another popular method for maximising the soft margin is by introducing a parameter \( \nu \) with (3.4) which is:

a) an upper bound on the fraction of margin errors \( (\xi_i > 0) \), and
b) a lower bound on the fraction of support vectors.

The objective function to minimise with the \( \nu \) parameter is then ([193]):

\[
\min_{w,\xi,\rho} \left[ \frac{1}{2} w \cdot w - \nu \rho + \frac{1}{m} \sum_{i=1}^{m} \xi_i \right] \quad (3.17)
\]
subject to the constraints:

\[ y_i(w \cdot x_i + b) \geq \rho - \xi_i \quad (3.18) \]
\[ \xi_i \geq 0 \quad (3.19) \]
\[ \rho \geq 0 \quad (3.20) \]

The primal form of the Lagrange function with the Lagrange multipliers \((\alpha, \beta, \delta)\) satisfying (3.17)-(3.20):

\[
L(w, \xi, b, \rho, \alpha, \beta, \delta) = \frac{1}{2}(w \cdot w) - \nu \rho + \frac{1}{m} \sum_{i=1}^{m} \xi_i
- \sum_{i=1}^{m} (\alpha_i[y_i(w \cdot x_i + b) - \rho + \xi_i] + \beta_i \xi_i) - \delta \rho
\] (3.21)

where \(\alpha_i \geq 0, \beta_i \geq 0\) and \(\delta_i \geq 0\).

The KKT conditions are:

\[
\beta_i \xi_i = 0 \quad (3.22)
\]
\[
\delta \rho = 0 \quad (3.23)
\]
\[
\alpha_i[y_i(w \cdot x_i + b) - \rho + \xi_i] = 0 \quad (3.24)
\]

To get the minimum, let’s take the derivatives of \(L\) with respect to \(w, \xi, b, \rho\) and set them to zero:

\[
\frac{\partial L}{\partial w} = w - \sum_{i=1}^{m} \alpha_i y_i x_i = 0 \quad (3.25)
\]
\[
\frac{\partial L}{\partial \xi_i} = \frac{1}{m} - \alpha_i - \beta_i = 0 \quad (3.26)
\]
\[
\frac{\partial L}{\partial b} = \sum_{i=1}^{m} \alpha_i y_i = 0 \quad (3.27)
\]
\[
\frac{\partial L}{\partial \rho} = \sum_{i=1}^{m} \alpha_i - \delta - \nu = 0 \quad (3.28)
\]

To get the dual form let’s substitute (3.25)-(3.28) into (3.21):
3.6 Using kernels to map input space to feature space for non-linear separability

So far I have only presented the theory of soft margin SVMs with the assumption that datapoints of classes are linearly separable in the input space. However, in real world datasets they are mostly non-linear. The general approach is to map the input space into a higher dimensional space known as feature space. In this space the datapoints of the classes are linearly separable by a hyperplane which maps to the nonlinear boundary in the input space.

To obtain this mapping let’s begin by substituting the dot product \((\mathbf{x}_i \cdot \mathbf{x}_j)\) of the dual form [(3.13) and (3.29)] with a mapping function:

\[
\mathbf{x}_i \cdot \mathbf{x}_j \rightarrow \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)
\]

where \(\Phi(\mathbf{x}_i)\) is the mapping function representing a kernel substitution:
3.7. The $C$ of $C$-SVM and the $\nu$ of $\nu$-SVM

The dual form of $C$-SVM (3.13) can then be rewritten as:

$$
\max_{\alpha} \left[ \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j K(x_i \cdot x_j) \right]
$$

(3.33)

and the dual form of $\nu$-SVM (3.29) can then be rewritten as:

$$
\max_{\alpha} \left[ -\frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j K(x_i \cdot x_j) \right]
$$

(3.34)

There are three kernel functions widely used in practice:

1. Linear: $K(x_i, x_j) = x_i \cdot x_j$

2. Polynomial: $K(x_i, x_j) = (\gamma x_i \cdot x_j + r)^d \quad \gamma > 0$

3. Gaussian: $K(x_i, x_j) = e^{-\gamma ||x_i - x_j||^2} \quad \gamma > 0$

To illustrate how each of the above kernel functions differ, let us visualise three instances of the positive and negative class datapoints being graphed onto a two dimensional plane in the input space as shown in Figure 3.2. Looking at the three instances, we can see that a linear separation is not possible for the datapoints of the input space in Figure 3.2b and 3.2c. By applying a mapping of the datapoints in Figure 3.2b and 3.2c from input space to feature space using a polynomial and a Gaussian kernel respectively, a separation of the classes can be achieved. Although in the input space the hyperplane does not appear linear, in feature space it is.

3.7 The $C$ of $C$-SVM and the $\nu$ of $\nu$-SVM

Now that the theory of soft margin SVMs is established lets focus our attention on the parameters that are used in practice.

Recall the objective function of a $C$-SVM (3.5):

$$
\min_{w, \xi} \left[ \frac{1}{2} w \cdot w + C \sum_{i=1}^{m} \xi_i \right]
$$
3.7. The $C$ of $C$-SVM and the $\nu$ of $\nu$-SVM

Figure 3.2: A comparison of different kernel mapping – we can see that linear separation is not possible in (b) and (c). By applying a mapping of the data-points using the kernels as shown, a separation of the classes can be achieved. Although in the input space the hyperplane does not appear linear, in feature space it is.
3.8. The $\gamma$ of polynomial and Gaussian kernel

The parameter $C$ controls the penalty of the margin errors ($\xi_i > 0$). For a small $C$ the soft margin increases while a large $C$ will lead to a hard margin decision boundary.

As for a $\nu$-SVM, recall the objective objection (3.17):

$$\min_{w, \xi, \rho} \left[ \frac{1}{2} w \cdot w - \nu \rho + \frac{1}{m} \sum_{i=1}^{m} \xi_i \right]$$

The parameter $\nu$ controls the penalty on the fraction of margin errors ($\xi_i > 0$) and the fraction of support vectors. For small $\nu$ the soft margin decreases while a large $\nu$ leads to a large margin.

The two parameters described ($C$ of $C$-SVM and $\nu$ of $\nu$-SVM) need to be tuned for a given problem. The search for the best parameter $C$ or $\nu$ starts with knowing the interval for which the values are valid. For $C$ the interval is $[0, \infty]$ and for $\nu$ the interval is $[0, 1]$. Because $\nu$ is limited between zero and 1, finding the best $\nu$ is easier than finding the best $C$. However, without looking for the best $C$ it is difficult to judge whether $\nu$-SVM is superior in returning the best performance compared to a $C$-SVM, as the best performance is determined by the classifier’s correct classification rate.

3.8 The $\gamma$ of polynomial and Gaussian kernel

To tackle non-linear datasets, SVM uses kernels for mapping the dataset from input space to feature space so that linear separability can be achieved. The use of kernels means that there are more parameters to consider.

From the literature review, the main kernel used to tackle real world datasets with non-linearity is the Gaussian kernel:

$$K(x_i, x_j) = e^{-\gamma \|x_i - x_j\|^2} \quad \gamma > 0$$

The parameter $\gamma$ controls the shape of the decision boundary, i.e. the separating hyperplane. For small $\gamma$ the number of support vectors is minimised (i.e. large margin) while a large $\gamma$ increases the number of support vectors, which leads to a more flexible hyperplane and a smaller margin – otherwise known as overfitting.

The other kernel of interest for real world datasets is the polynomial kernel:

$$K(x_i, x_j) = (\gamma x_i \cdot x_j + r)^d \quad \gamma > 0$$
3.9. One-class classification using hyperplane and hypersphere

From the above description we can see that there are additional parameters (besides $\gamma$) to consider. The degree of the polynomial is similar to the role of $\gamma$ in the Gaussian kernel. However, from the literature review a low-degree polynomial has shown to be used in practice where the degree $d$ is set to 2 and $r$ is set to 1 [28]. This leaves us with the parameter $\gamma$ which acts in a similar way to the $\gamma$ in the Gaussian kernel.

By using kernels with soft margin SVMs we are faced with searching for two best parameters, $C$ or $\nu$ with $\gamma$. This task is complicated by the unclear intervals of $C$ and $\gamma$. I explore and discuss this issue further in subsequent chapters.

3.9 One-class classification using hyperplane and hypersphere

Prior to concluding this chapter, let’s extend our knowledge of soft margin SVMs so that it can be applied to datasets with only one class – a method that is properly named One-Class SVM. The task of a One-Class SVM is to create a soft boundary around the positive class so that it accepts as many of the positive class datapoints as possible. This effectively creates a virtual negative space. In the literature there are two popular approaches for One-Class SVM. One uses a hyperplane in feature space and the other uses a hypersphere in feature space.

Let’s begin with the theory of One-Class SVM using a hyperplane in feature space.

The object function to minimise is (194):

$$
\min_{\mathbf{w}, \xi, \rho} \left[ \frac{1}{2} \mathbf{w} \cdot \mathbf{w} - \rho + \frac{1}{\nu m} \sum_{i=1}^{m} \xi_i \right]
$$

(3.35)

subject to the constraints:

$$\mathbf{w} \cdot \Phi(\mathbf{x}_i) \geq \rho - \xi_i$$

(3.36)

$$\xi_i \geq 0$$

(3.37)

where $\Phi$ is the mapping function from input space to feature space, $\mathbf{w}$ is the weight vector in feature space, $\rho$ is the offset of the hyperplane, $\xi_i$ are the slack variables for soft margin separation, and $\nu$ is the parameter to be tuned.
The primal form of the Lagrange function with the Lagrange multipliers \((\alpha, \beta)\) satisfying (3.35)-(3.37):

\[
L(w, \xi, \rho, \alpha, \beta) = \frac{1}{2}(w \cdot w) - \rho + \frac{1}{\nu m} \sum_{i=1}^{m} \xi_i \\
- \sum_{i=1}^{m} \alpha_i [(w \cdot \Phi(x_i)) - \rho + \xi_i] - \sum_{i=1}^{m} \beta_i \xi_i
\] (3.38)

where \(\alpha_i \geq 0\) and \(\beta_i \geq 0\).

The KKT conditions are:

\[
\beta_i \xi_i = 0 \quad (3.39)
\]
\[
\alpha_i [(w \cdot \Phi(x_i)) - \rho + \xi_i] = 0 \quad (3.40)
\]

To get the minimum, let’s take the derivatives of \(L\) with respect to \(w\), \(\xi\), \(\rho\) and set them to zero:

\[
\frac{\partial L}{\partial w} = w - \sum_{i=1}^{m} \alpha_i \Phi(x_i) = 0 \quad (3.41)
\]
\[
\frac{\partial L}{\partial \xi_i} = \frac{1}{\nu m} - \alpha_i - \beta_i = 0 \quad (3.42)
\]
\[
\frac{\partial L}{\partial \rho} = \sum_{i=1}^{m} \alpha_i - 1 = 0 \quad (3.43)
\]

To get the dual form let’s substitute (3.41)-(3.43) into (3.38):

\[
\max_{\alpha} \left[ -\frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j (\Phi(x_i) \cdot \Phi(x_j)) \right] \quad (3.44)
\]

which must be maximised with respect to \(\alpha_i\), subject to the constraints:

\[
\sum_{i=1}^{m} \alpha_i = 1 \quad (3.45)
\]
\[
0 \leq \alpha_i \leq \frac{1}{\nu m} \quad (3.46)
\]
3.9. One-class classification using hyperplane and hypersphere

The first constraint arises from (3.43). The second constraint (3.46) is a result of the KKT condition $\beta_i \xi_i = 0$ (3.39) with the Lagrange derivative $\frac{1}{\nu m} - \alpha_i - \beta_i = 0$ (3.42), i.e. assuming we have a margin error ($\xi_i > 0$), then (3.39) implies $\beta_i = 0$, and using (3.42) we get $0 \leq \alpha_i \leq \frac{1}{\nu m}$.

The dual form (3.44) with a kernel substitution is rewritten as:

$$\max_{\alpha} \left[ -\frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j K(x_i, x_j) \right]$$ (3.47)

Now let’s introduce the theory of One-Class SVM using a hypersphere in feature space.

The objective function to minimise is ([204]):

$$\min_{w, \xi} \left[ R^2 + \frac{1}{\nu m} \sum_{i=1}^{m} \xi_i \right]$$ (3.48)

subject to the constraints:

$$R^2 + \xi_i \geq (x_i - a) \cdot (x_i - a)$$ (3.49)
$$\xi_i \geq 0$$ (3.50)

where $R$ is the minimal radius of the hypersphere centred at $a$ and $\xi_i$ are the slack variables for soft margin separation.

The primal form of the Lagrange function with the Lagrange multipliers $(\alpha, \beta)$ satisfying (3.48)-(3.50):

$$L(R, a, \xi, \alpha, \beta) = R^2 + \frac{1}{\nu m} \sum_{i=1}^{m} \xi_i - \sum_{i=1}^{m} \beta_i \xi_i - \sum_{i=1}^{m} \alpha_i \left( R^2 + \xi_i - (x_i \cdot x_i - 2a \cdot x_i + a \cdot a) \right)$$ (3.51)

where $\alpha_i \geq 0$ and $\beta_i \geq 0$.

The KKT conditions are:

$$\beta_i \xi_i = 0$$ (3.52)
$$\alpha_i [R^2 + \xi_i - (x_i - a) \cdot (x_i - a)] = 0$$ (3.53)
3.10. The \( \nu \) of One-Class SVM

To get the minimum, let’s take the derivatives of \( L \) with respect to \( R, \xi, a \) and set them to zero:

\[
\frac{\partial L}{\partial R} = 1 - \sum_{i=1}^{m} \alpha_i = 0 \quad R > 0 \quad (3.54)
\]

\[
\frac{\partial L}{\partial \xi_i} = \frac{1}{\nu m} - \alpha_i - \beta_i = 0 \quad (3.55)
\]

\[
\frac{\partial L}{\partial a} = \sum_{i=1}^{m} \alpha_i \mathbf{x}_i - a = 0 \quad (3.56)
\]

To get the dual form let’s substitute (3.54)-(3.56) into (3.51):

\[
\max_{\alpha} \left[ \sum_{i=1}^{m} \alpha_i \mathbf{x}_i \cdot \mathbf{x}_i - \sum_{i,j=1}^{m} \alpha_i \alpha_j \mathbf{x}_i \cdot \mathbf{x}_j \right] \quad (3.57)
\]

which must be maximised with respect to \( \alpha_i \), subject to the constraints:

\[
\sum_{i=1}^{m} \alpha_i = 1 \quad (3.58)
\]

\[
0 \leq \alpha_i \leq \frac{1}{\nu m} \quad (3.59)
\]

The first constraint arises from (3.54). The second constraint (3.59) is a result of the KKT condition \( \beta_i \xi_i = 0 \) (3.52) with the Lagrange derivative \( \frac{1}{\nu m} - \alpha_i - \beta_i = 0 \) (3.55), i.e. assuming we have a margin error \( \xi_i > 0 \), then (3.52) implies \( \beta_i = 0 \), and using (3.55) we get \( 0 \leq \alpha_i \leq \frac{1}{\nu m} \).

The dual form (3.44) with a kernel substitution is rewritten as:

\[
\max_{\alpha} \left[ \sum_{i=1}^{m} \alpha_i K(\mathbf{x}_i, \mathbf{x}_i) - \sum_{i,j=1}^{m} \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) \right] \quad (3.60)
\]

3.10 The \( \nu \) of One-Class SVM

I have now presented the theory of One-Class SVM using a hyperplane and a hypersphere approach. Next I discuss the parameters. Recall the objective function of a One-Class SVM using a hyperplane approach (3.35):

\[
\min_{w, \xi, \rho} \left[ \frac{1}{2} \mathbf{w} \cdot \mathbf{w} - \rho + \frac{1}{\nu m} \sum_{i=1}^{m} \xi_i \right]
\]
3.11 Computational complexity in training and testing SVMs

The parameter $\nu$ controls the penalty on the fraction of margin errors ($\xi_i > 0$) and the fraction of support vectors. For small $\nu$ the soft margin decreases while a large $\nu$ leads to a large margin.

For a One-Class SVM using a hypersphere approach, recall the objective function (3.48):

$$\min_{w, \xi} \left[ R^2 + \frac{1}{\nu m} \sum_{i=1}^{m} \xi_i \right]$$

The parameter $\nu$ controls the radius of the hypersphere and the number of training samples it can hold. For small $\nu$ more datapoints are placed inside the ‘ball’ while a large $\nu$ squeezes the size of the ‘ball’.

Both the hyperplane and hypersphere approach have a parameter $\nu$ which needs to be tuned for a given problem. The search for the best $\nu$ starts with knowing the interval for which the values are valid, and this is $(0, 1]$. We can see that the interval of $\nu$ described for One-Class SVM is similar to the $\nu$-SVM interval ($\nu \in [0, 1]$) except zero is not valid.

3.11 Computational complexity in training and testing SVMs

The computational complexity of training an SVM is $O(m^3)$ where $m$ is the number of training samples [19, 209]. This computational complexity is due mostly to solving the quadratic programming (QP) problems that arises from SVMs. For the testing phase the computational complexity of an SVM is linear to the number of support vectors and constant to the number of test samples.

3.12 Frameworks of two-class classifiers for multiclass classification

The final section presents the different frameworks of two-class classifiers for multiclass classification. From the literature review there are two primary frameworks for tackling problems with more than two-classes.

The first is known as a one-vs-all framework. This approach divides an $n$-class problem into $n$ two-class problems. Each two-class problem is a binary classifier that learns from positive data corresponding to the positive class and
negative data corresponding to the remaining classes [130, 184]. The size of the negative data is generally larger when compared to the size of the positive data. This difference in size of data samples between the two-classes is known as imbalanced data and is a natural phenomenon in real world datasets.

The second framework, known as a one-vs-one framework, divides an $n$-class problem into $\binom{n}{2}$ two-class problems where each binary classifier is a pair of the classes [101]. Compared to the one-vs-all framework, the class sizes for the binary classifiers of the one-vs-one framework are more likely balanced, or at least close to balanced, i.e. it reduces any severe effects of imbalanced data at the cost of a large increase in the number of classifier tasks.

The two frameworks described are explored further in the next chapter where I begin looking deeper into other issues of multiclass classification.

3.13 Summary

In Chapter 1, I presented my thesis question ‘Can a machine describe the emotional response to a scene using perceived faces and facial expressions?’ I have presented the relevant literature from Cognitive Science and Psychology in the previous chapter as to why faces and facial expressions are the appropriate features for describing a scene, and why the Universal Facial Expressions of Emotion was chosen. My final remarks from that chapter described how our ability to see faces from minimal face-like patterns link pareidolia to the creative process of an individual.

Before I explored my thesis question any further, a background into the theory of my primary machine learning tool was presented. I began with a literature review in support of why I chose SVM as my preferred method. The review was divided into general classification, evaluating facial expression, and face detection. From these quoted studies I showed that SVM is still a popular and most capable classifier that is appropriate for my study. The remaining sections presented the mathematics behind SVM and its parameters used in my experiments.

In general, the purpose of SVM is to find a linear separation of datapoints of two-classes defined by a separating hyperplane. The datapoints nearest to the separating hyperplane are known as support vectors and the distance between the support vectors of the classes, which is measured perpendicular to the separating hyperplane, is known as the margin of the separating hyperplane. A primary aim of SVM is to find a separating hyperplane where the support
vectors of the two-classes are as far apart as possible i.e. maximal margin separation.

This rough description is akin to visualising a block of land from above with a 2 lane road running through it. The line in the middle of the road is the separating hyperplane and the parallel hyperplanes formed by the support vectors are the edge of the road. Widening the road increases the width of both lanes and from above we can see a clear decision of the land separation – which is akin to having a larger margin. Because of the larger margin this ease in separating one class from the other is known as generalisation.

From the literature review, there were several ways of providing the theory for SVMs. As my work explores real world datasets, I presented a theory of soft margin SVMs. Soft margin SVMs allow datapoints to occur within the margin of the separating hyperplanes – i.e. margin errors. After presenting the two primary soft margin SVMs, C-SVM and \( \nu \)-SVM, I then covered the use of kernels for mapping the datapoints of the classes from input space to feature space. The use of kernels was vital when linear separation was not possible in the input space but possible in feature space, and the three kernels that were presented were the linear, the polynomial and Gaussian kernel.

Next I discussed the relevant parameters. The soft margin SVMs I use are always paired with a kernel substitution. In this instance, the relevant parameters that require tuning are \( C \) of C-SVM or \( \nu \) of \( \nu \)-SVM with \( \gamma \) of polynomial and Gaussian kernel. The reason as to why only \( \gamma \) requires tuning for the polynomial kernel is to model the performance of a low-degree polynomial used in practice where the degree \( d \) is set to 2 and the constant \( r \) is set to 1.

The chapter then led to the theory of One-Class SVM. One-Class SVMs are useful when a training data contains datapoints of only one class. Once again I presented the theory using soft margins since I am exploring real world datasets. The aim is then to create a soft boundary around this one class so that it accepts as much of the class data points as possible – effectively creating a virtual negative space. Two popular methods were presented. One using a hyperplane approach and the other using a hypersphere approach. Both have a \( \nu \) parameter that needed to be tuned.

The final section then presented the frameworks that extend SVMs from solving a two-class classification problem to a dataset with multiple classes – i.e. multiclass classification. The two frameworks presented were the one-vs-all and the one-vs-one. The one-vs-all framework divides an \( n \)-class problem into \( n \) two-class problems where each binary classifier learns from positive data.
corresponding to the positive class and negative data corresponding to the remaining classes, while a one-vs-one framework divides an $n$-class problem into $\binom{n}{2}$ two-class problems where each binary classifier is a pair of the classes [101]. Compared to the one-vs-all framework, the class sizes for the binary classifiers of the one-vs-one framework are more likely balanced, or at least close to balanced. The two frameworks described are explored further in the next chapter where I begin looking deeper into other issues of multiclass classification.

Concluding the chapter summary, I have presented the main theory of SVM in a two-class, one-class and multiclass domain. The main essence is for the reader to understand where the parameters are conceived in its mathematical form, and how the SVM behaves with the changes to those parameters. In subsequent chapters I present methods in finding the best parameters for providing the best performance and understanding the dynamics of those parameters.
The contents of this chapter have been published in [97].

Contents

4.1 Introduction .................................. 44
4.2 System overview .............................. 47
  4.2.1 Dataset description .......................... 47
  4.2.2 k-fold cross validation algorithm .......... 49
  4.2.3 Support Vector Machines (SVM) and kernels .. 50
  4.2.4 The SVM and kernel parameters to optimise ... 50
  4.2.5 Calculating the SVM and kernel search parameters . 50
  4.2.6 Pseudocode for standard multiclass SVM and pa-SVM . 51
4.3 Results and discussion ...................... 55
  4.3.1 pa-SVM mostly outperforms standard SVM for multiclass classification ........ 60
  4.3.2 Infeasible parameters evident for \( \nu \)-SVM ........ 60
  4.3.3 Statistical significance defined by \( p \)-values ........ 60
  4.3.4 Favourable results compared to recent studies .... 61
  4.3.5 Proper selection of \((C, \gamma)\) and \((\nu, \gamma)\) values can reduce the number of support vectors .......... 63
4.4 Summary ..................................... 64

In Chapter 1, I presented my thesis question ‘Can a machine describe the emotional response to a scene using perceived faces and facial expressions?’ The dissertation of this question led us to chapter 2, which explored literature from the disciplines of Cognitive Science and Psychology to substantiate its relevance and importance – this is an early indication that my thesis is an
interdisciplinary quest. The path then led us into my field of machine learning and image processing techniques, where, in the previous chapter I presented to the reader the theory of Support Vector Machines and the parameters where tuning is required.

In this chapter I describe and experimentally evaluate a new variation of multiclass classification using soft margin Support Vector Machines (SVM). The technique, called ‘Pairwise Adaptive Support Vector Machines’ (pa-SVM), is a one-vs-one multiclass framework where each binary classifier is optimised towards using the parameters that obtain the best correct classification rate. Two pa-SVMs were created and evaluated – one using $C$-SVM and the other using $\nu$-SVM. The parameters to optimise are $C$ of $C$-SVM or $\nu$ of $\nu$-SVM with $\gamma$ of a Gaussian kernel or a degree 2 polynomial kernel. An exponential grid search and a 10-fold cross validation algorithm were used to determine (for each binary classifier) the best $(C, \gamma)$ for the pa-SVM using the $C$-SVM and the best $(\nu, \gamma)$ for the pa-SVM using the $\nu$-SVM.

My experiments used 23 real world datasets from the UCI Machine Learning Repository [64]. The results show that the pa-SVM approach mostly outperforms the standard approach, and that the number of support vectors are minimised by selecting the max $C$ min $\gamma$ scenario (using the $C$-SVM) or the min $\nu$ min $\gamma$ scenario (using the $\nu$-SVM) when the best classification rate of a binary classifier has more than one parameter pair. A highlight of my experience with $\nu$-SVM is that there are datasets where no feasible hyperplanes exist – even though it seems a favourable choice because $\nu \in [0, 1]$ over $C$-SVM where $C \in [0, \infty)$. Furthermore, the comparison of my results with recent studies using the same datasets show that the new technique is very competitive.

4.1 Introduction

Two-class classification is a well understood task in the machine learning community. The task becomes more complex when more classes are added (i.e. multiclass classification) and particularly when the size of the classes differ greatly (i.e. classification of imbalanced data). The concept of pairwise comparison of classes (“pairwise concept”) which is fundamental to two-class classification is also inherent in multiclass classification. I describe a number of issues in multiclass classification to outline the core components of a multiclass classifier, and propose a refinement in the pairwise concept to improve classification performance.
4.1. Introduction

In Section 3.12, I have described two frameworks for multiclass classification. The simplest and least structured pairwise representation is the one-vs-all framework. Its naïve structure means that classification performance would rely heavily on how well each classifier is tuned to the problem [177]; which is a challenging task because of the imbalanced negative data.

Imbalanced data is a natural phenomenon in real world datasets and it is a problem because the uneven class distribution creates a decision boundary that is biased towards the majority class and less sensitive towards the minority class. To compensate, a pairwise method oversamples the patterns of the minority class or undersamples the patterns of the majority class. An undesirable effect of oversampling the minority class is overfitting the class [88]. This effect is often tackled by modifying the duplicated samples to generalise the decision boundary by means of randomised interpolation between data points within the feature space [140] as well as adaptive oversampling the minority classes near the borderline [85]. A drawback of undersampling the majority class is the possible loss of crucial and valuable data [3]. In either case, sampling the data is regarded as a better alternative than introducing weights in the algorithm (often as cost matrices to offset the bias of the classifier towards the majority class [102]) which generally results in a problem specific algorithm and does not offer the versatility of using different classification methods [169].

As previously noted, imbalanced data is evident in a one-vs-all framework because the negative data is generally larger than the positive data. However, in a one-vs-one framework any severe effects of imbalanced data are reduced because the class sizes for the binary classifiers of the one-vs-one framework are more likely balanced, or at least close to balanced. However, due to the increased number of binary classifiers compared to the one-vs-all framework, the one-vs-one framework raises questions as to how the binary classifiers are evaluated to determine the class. I denote this evaluation process as a strategy and in the literature there are two strategies commonly used: the first determines the class by the classifier with the highest probability (a.k.a. weighted strategy) while the second determines the class with the majority number of votes (a.k.a. majority voting strategy) [184]. Each strategy has its strengths and weaknesses based on the framework chosen. In a one-vs-all framework, the small number of classifiers (when compared to a one-vs-one framework) means that class ties are more likely to occur, and this is where weighted strategies using probabilities have been the preferred choice [68, 177, 184]. However, studies such as [95] attempt to break class ties by using a Naïve Bayes classifier to dynamically order the classifiers of the tied classes. As
for the one-vs-one framework, the increased number of classifiers means that determining the correct class does not necessarily need to rely on a weighted strategy, but can be decided upon by the number of votes [66]; although weights have been used between binary classifiers to determine the class [184]. Further, the one-vs-one framework has been shown to perform best for large datasets in practice [31, 89]. The implied benefits of reducing the effects of imbalanced data and the practical applications for large datasets makes the one-vs-one framework the preferred choice over a one-vs-all framework.

I have so far outlined three core components of a multiclass classifier. The first is the framework, the second is the strategy used within the framework to determine the class, and the third is the learning classifier – otherwise known as the base learner. I propose a pairwise refined method whereby using the strengths of each component will lead to an improved performance for multiclass classification. In this respect, the one-vs-one framework is well suited to the task because it is more pairwise refined than the one-vs-all framework, and a number of studies have shown that this framework offers a better classification rate than a one-vs-all framework [58, 59, 68, 130, 138, 184], as well as a faster performance [130, 138]. As for the strategy, the weighted strategy has gained some interest recently using methods such as decision trees [101, 138, 181, 184], adaptive voting based on optimal maximum a-posteriori probability (MAP) predictions of labels [101], weighted voting using the two-class/class-pair posterior probabilities [184, 201], and weighted matrix of discrete probability density functions produced by the set of dichotomisers in an ECOC classifier [59]. However, the main concern with using a weighted strategy is arriving at a solution that does not generalise well in cases of imbalanced class data (resulting in a problem specific algorithm); because of this the simplicity of the majority voting strategy is my preferred choice with the one-vs-one framework. The literature review in Section 3.1.1 provided the reasons for why SVM is my preferred classifier for this study.

The chosen framework and the default strategy (one-vs-one with majority voting) used in current SVMs for multiclass classification is consistent with my approach [30]. However, this may not be an optimal solution because current SVMs used in practice have parameters that are one and the same for all their binary classifiers. I propose a ‘Pairwise Adaptive SVM (pa-SVM)’, where each binary classifier within the multiclass framework has optimised parameters for its task. For the assessment, a number of real world datasets have been chosen from the UCI Machine Learning Repository [64], and since I am evaluating on real world datasets soft margin SVMs are used, along with
a kernel substitution for tackling non-linear separation of data by mapping
them from input space to feature space\textsuperscript{1}. This means that there are two sets
of parameters that need to be optimised – one for SVM and other for the
kernel. The datasets are normalised in my experiments as this is a common
preprocessing technique for improving generalisation and classification accu-
rracy \cite{76}, and to maintain a consistent low biased classification accuracy, a
standard 10-fold cross validation method is used throughout my experiments
\cite{180}.

To the best of my knowledge this method was not found in the literature,
nor was it available in any standard SVM library. The contribution of this
chapter is to provide a description and evaluation of this technique. The idea
of finding the best parameters for each binary classifier of multiclass SVMs
arose from my comparative study on classifying discrete facial expressions
using a holistic versus a component based approach\textsuperscript{2} \cite{96}. My aim here is to
investigate the generalisability of this method for other datasets and assess its
performance.

A system overview is presented in the next section, followed by a descrip-
tion of the datasets and the methods. Section 4.3 presents the experimental
results and discussion, followed by concluding remarks in Section 4.4.

4.2 System overview

My pa-SVM system consists of six steps. An illustration of the steps is shown
in Figure 4.1. The first step is to normalise the dataset and then partition
the dataset by class. The class partitions are then divided into training and
validation sets for use in a k-fold cross validation algorithm. The SVM and
kernel grid parameters are obtained for the standard and pairwise adaptive
multiclass experiments, and the results are then compared. Details of further
steps are explained in subsequent subsections.

4.2.1 Dataset description

Twenty three real world datasets most commonly used in recent studies were
sourced from the UCI Machine Learning Repository \cite{64}. Table 4.1 lists the
dataset’s name, size and number of classes. Any dataset with a separate

\textsuperscript{1}Refer to Chapter 3 for further reading on soft margin SVM and kernels.

\textsuperscript{2}Presented in Chapter 5 ‘Facial Expressions Analysis’
4.2. System overview

Figure 4.1: System overview showing the flow of the stages and states.
training and testing set (marked by the * symbol) were merged together and the values of each dataset were normalised between +1 and -1.

Table 4.1: Description of the datasets from the UCI Machine Learning Repository [64]. Datasets with a separate training and testing set are merged together for this study (marked by *).

<table>
<thead>
<tr>
<th>Name</th>
<th>Size</th>
<th>Number of classes</th>
</tr>
</thead>
<tbody>
<tr>
<td>abalone</td>
<td>4177</td>
<td>29</td>
</tr>
<tr>
<td>audiology*</td>
<td>266</td>
<td>8</td>
</tr>
<tr>
<td>balance</td>
<td>625</td>
<td>3</td>
</tr>
<tr>
<td>car</td>
<td>1728</td>
<td>4</td>
</tr>
<tr>
<td>dermatology</td>
<td>366</td>
<td>6</td>
</tr>
<tr>
<td>e.coli</td>
<td>336</td>
<td>8</td>
</tr>
<tr>
<td>glass</td>
<td>214</td>
<td>6</td>
</tr>
<tr>
<td>iris</td>
<td>150</td>
<td>3</td>
</tr>
<tr>
<td>isolet*</td>
<td>7797</td>
<td>26</td>
</tr>
<tr>
<td>krkopt</td>
<td>28056</td>
<td>18</td>
</tr>
<tr>
<td>letter</td>
<td>20000</td>
<td>16</td>
</tr>
<tr>
<td>nursery</td>
<td>12960</td>
<td>5</td>
</tr>
<tr>
<td>optdigits*</td>
<td>5620</td>
<td>10</td>
</tr>
<tr>
<td>page-blocks</td>
<td>5473</td>
<td>10</td>
</tr>
<tr>
<td>pendligits*</td>
<td>10992</td>
<td>10</td>
</tr>
<tr>
<td>satimage*</td>
<td>6435</td>
<td>6</td>
</tr>
<tr>
<td>segment*</td>
<td>2310</td>
<td>7</td>
</tr>
<tr>
<td>vehicle</td>
<td>846</td>
<td>4</td>
</tr>
<tr>
<td>vowel528</td>
<td>528</td>
<td>11</td>
</tr>
<tr>
<td>vowel990</td>
<td>990</td>
<td>11</td>
</tr>
<tr>
<td>waveform</td>
<td>5000</td>
<td>3</td>
</tr>
<tr>
<td>wine</td>
<td>178</td>
<td>3</td>
</tr>
<tr>
<td>yeast</td>
<td>1484</td>
<td>10</td>
</tr>
</tbody>
</table>

4.2.2 \textit{k}-fold cross validation algorithm

A \textit{k}-fold cross validation algorithm was employed for evaluating optimised parameters of the grid search. The algorithm partitions a dataset $X$ into $k$ (training, validation) pairs where the length of the training set is $(k-1) \times \text{len}(X)/k$, while the length of the validation set is $\text{len}(X)/k$; and each sample
of $X$ exists no more than once across the validation sets. The pseudocode is described in Algorithm 1. A 10-fold was chosen to maintain a consistent low biased accuracy [180].

**Algorithm 1** Dividing dataset into training and validation sets for $k$-fold cross validation

Let $X$ be the dataset array and the index of the first element is 0.
Let $X[i]$ define a sample element of dataset $X$ at index $i$.
Let len($X$) return the length of the dataset $X$.
Let $\%$ define a modulo operation.
Let $[j][i]$ be a 2D array indexed by $j$ and $i$ respectively.

1: for $j \leftarrow 0, K - 1$ do
2: for $i \leftarrow 0, \text{len}(X) - 1$ do
3: if $i \% K == j$: then
4: $\text{validation}[j][i] \leftarrow X[i]$
5: else
6: $\text{training}[j][i] \leftarrow X[i]$
7: end if
8: end for
9: end for

---

### 4.2.3 Support Vector Machines (SVM) and kernels

From the literature review of kernels with SVMs in Section 3.2.1, the Gaussian kernel and the degree 2 polynomial kernel with a constant of 1 was chosen.

### 4.2.4 The SVM and kernel parameters to optimise

From the theories of SVM and kernels discussed in Chapter 3 and specifically Sections 3.7-3.8, the two pairs of parameters to tune are thus $(C, \gamma)$ and $(\nu, \gamma)$, where $C$ is a parameter for $C$-SVM, $\nu$ is a parameter for $\nu$-SVM and $\gamma$ is a parameter for the Gaussian and polynomial kernel.

### 4.2.5 Calculating the SVM and kernel search parameters

The sequences of $(C, \gamma)$ are grown exponentially by 2 to the power of $x$. This procedure is commonly used in practice [32]. I used very similar values to [32].

$$C \in \{2^{-5}, 2^{-4}, ..., 2^{14}\}$$
and

\[ \gamma \in \{ 2^{-17}, 2^{-16}, ..., 2^2 \} \]

However, for the sequences of \((\nu, \gamma)\) I grow \(\nu\) by a linear step of 0.05 up to 1.0; while \(\gamma\) remains the same. i.e.

\[ \nu \in \{ 0.05, 0.10, ..., 0.95, 1.00 \} \]

### 4.2.6 Pseudocode for standard multiclass SVM and \(pa\)-SVM

The significant difference between the standard multiclass SVM and \(pa\)-SVM is that each of the binary classifiers of the \(pa\)-SVM can have individually different \((C, \gamma)\) or \((\nu, \gamma)\) values. I describe the pseudocode of the standard SVM and \(pa\)-SVM in Algorithm 2 and Algorithm 3 respectively.

Looking at Algorithm 2, the standard multiclass SVM begins looping over the SVM and kernel parameters for each fold and for each pair of classes. At each pair of classes a binary classifier is trained and validated using the partitioned data from the \(k\)-fold cross validation algorithm. At each fold the classes are determined by the majority voting strategy. This then leads to the mean correct classification rate \(\mu_{\text{CCR}}\) and the mean number of support vectors \(\mu_{\text{NSV}}\) of all binary classifiers over the folds. I keep track of infeasible parameters where a binary classifier could not be constructed by the parameters specified, and the algorithm is terminated if a particular pair of classes does not have any feasible parameters – regardless of the fold.

Looking at Algorithm 3, the \(pa\)-SVM starts by looping over each pair of classes for each fold and for each SVM and kernel parameter. Once again, at each pair of classes the binary classifiers are trained and validated using the partitioned data from the \(k\)-fold cross validation algorithm. However, during the initial search phase of finding the best SVM and kernel parameter, the validation stage at each binary classifier of a particular pair of classes uses only the validation data of that pair. Then later when the best SVM and kernel parameter are found for each pair of classes, the complete framework is evaluated using the combined validation data of all classes at each fold. The majority voting strategy is then used to determine the class that leads to the \(\mu_{\text{CCR}}\) and \(\mu_{\text{NSV}}\) of all binary classifiers over the folds. As with the standard SVM algorithm, I keep track of infeasible parameters to ensure a feasible solution space is identified by trimming any infeasible parameters occurring in any fold.
To evaluate multiple parameters of a binary classifier with the same best correct classification rate, I expanded my experiments to cover four scenarios of min/max $C$ and min/max $\nu$ with min/max $\gamma$. This meant that to cover the four scenarios, four experiments were conducted for the $pa$-SVM using the $C$-SVM, and four for the $pa$-SVM using the $\nu$-SVM. In the algorithm I decided to choose the min/max SVM parameter first followed by the min/max of the kernel parameter.
4.2. System overview

Algorithm 2 Standard SVM algorithm for multiclass classification

Let $j$ be the index of a $k$-fold cross validation algorithm.
Let $p$ be a class.
Let $(p, n)$ be a pair of classes.
Let $getValidationData(j)$ return the validation data of the $j$th fold across all classes.
Let $getTrainingSet(p, j)$ return the training set of the $j$th fold of class $p$.
Let $s$ be the SVM parameter which can be either $C$ or $\nu$.
Let $e(p, n)$ hold a set of infeasible $(s, \gamma)$ values for each $(p, n)$ classes

1: for all $(s, \gamma) \in$ grid search do
2: for all $j \in k$-fold do
3: for all $(p, n) \in$ binary classifiers do
4: \hspace{1em} $trainingData \leftarrow getTrainingSet(p, j) +$
\hspace{2em} getTrainingSet(n, j)
5: \hspace{1em} try
6: \hspace{2em} train classifier $(p, n)$ with $trainingData$
\hspace{2em} using parameters $(s, \gamma)$
7: \hspace{2em} tally the number of support vectors
\hspace{2em} of classifier $(p, n)$
8: catch
9: \hspace{1em} add $(s, \gamma)$ to $e(p, n)$
10: end try catch
11: end for
12: for all $(p, n) \in e$ do
13: \hspace{1em} if $e(p, n)$ contains all $(s, \gamma)$ then
14: \hspace{2em} exit. no viable solution.
15: \hspace{1em} end if
16: end for
17: $validationData \leftarrow getValidationData(j)$
18: for all $d \in validationData$ do
19: \hspace{1em} for all $(p, n)$ do
20: \hspace{2em} tally the correct output of
\hspace{2em} binary classifier $(p, n)$ with sample $d$
21: \hspace{1em} end for
22: \hspace{1em} determine class by max. votes
23: \hspace{1em} end for
24: calculate $j$th fold rate
25: calculate $j$th fold number of support vectors
26: end for
27: calculate $\mu_{CCR}$ over folds
28: calculate $\mu_{NSV}$ over folds
29: end for
30: determine best $(s, \gamma)$ scenario by max $\mu_{CCR}$ and min $\mu_{NSV}$
Algorithm 3 pa-SVM algorithm for multiclass classification

Let $j$ be the index of a $k$-fold cross validation algorithm.
Let $p$ be a class.
Let $(p,n)$ be a pair of classes.
Let $getTrainingSet(p,j)$ return the training set of the $j$th fold of class $p$.
Let $getValidationPair(j,p,n)$ return the validation data of the $j$th fold of classes $p$ and $n$.
Let $getValidationData(j)$ return the validation data of the $j$th fold across all classes.
Let $s$ be the SVM parameter which can be either $C$ or $\nu$.
Let $e(p,n)$ hold a set of infeasible $(s,\gamma)$ values for each $(p,n)$ classes

1: for all $(p,n) \in$ binary classifiers do
2:   for all $j \in k$-fold do
3:       $trainingData \leftarrow getTrainingSet(p,j)+$
4:        $getTrainingSet(n,j)$
5:   for all $(s,\gamma) \in$ grid search do
6:       try
7:         train classifier $(p,n)$ with $trainingData$
8:         using parameters $(C,\gamma)$
9:         store the number of support vectors
10:        of classifier $(p,n)$
11:       catch
12:       $(s,\gamma)$ to $e(p,n)$
13:   end try catch
14:   $validationData \leftarrow$
15:      $getValidationPair(j,p,n)$
16:   for all $s \in validationData$ do
17:     tally the correct output of
18:     binary classifier $(p,n)$ with sample $s$
19:   end for
20: end for
21: for all $(p,n) \in e$ do
22:   for all $(s,\gamma)$ in $e(p,n)$ do
23:     remove from solution space.
24: end for
25: for all $(s,\gamma)$ get best $\mu_{CCR}$ over folds
26: if multiple $(s,\gamma)$ with best $\mu_{CCR}$ select
27:   best $(s,\gamma)$ by min/max $(s,\gamma)$ scenario
28: end for
4.3. Results and discussion

In a previous publication [97], I presented the results across the four min/max scenarios that are defined by the parameter \( C \) (using the \( C \)-SVM) and \( \gamma \) (using the Gaussian kernel). Since then, I have extended my experiments using the \( \nu \)-SVM with the Gaussian and the polynomial kernel. The new algorithm, which takes into account infeasible parameters, was repeated for the \( C \)-SVM. From my analysis of the results, I am satisfied to focus on presenting the results from the max \( C \) min \( \gamma \) scenario (using the \( C \)-SVM) and the min \( \nu \) min \( \gamma \) scenario (using the \( \nu \)-SVM), for several reasons. Firstly, in all scenarios there were no correct classification rates from the standard SVMs that were higher than the correct classification rates from the \textit{pa}-SVM method; secondly, the two scenarios described have the smallest number of support vectors; and lastly, only a small number of rates from the two scenarios were not the highest rates,
4.3. Results and discussion

but the differences are minimal.

An overview of the experimental results is shown across Tables 4.2-4.5. The first two tables show the results of the pa-SVM using the C-SVM classifier and the last two tables show the results of the pa-SVM using the \( \nu \)-SVM classifier. Each table outlines the best \( \mu_{CCR} \) and \( \mu_{NSV} \) from the two scenarios described above. I discuss the details of the results in the following subsections.

Table 4.2: pa-SVM using C-SVM and Gaussian kernel: Best \( \mu_{CCR} \) with \( \mu_{NSV} \) from the max \( C \) min \( \gamma \) scenario. [\(*\) marker means statistically significant]

<table>
<thead>
<tr>
<th>Dataset</th>
<th>( \mu_{CCR} )</th>
<th>( p)-value</th>
<th>( \mu_{NSV} )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>pa-SVM</td>
<td>standard</td>
<td>( p)-value</td>
</tr>
<tr>
<td>abalone</td>
<td>29.45</td>
<td>28.05</td>
<td>1.7e-02*</td>
</tr>
<tr>
<td>audiology</td>
<td>88.69</td>
<td>83.10</td>
<td>9.1e-03*</td>
</tr>
<tr>
<td>balance</td>
<td>100</td>
<td>100</td>
<td>0.34</td>
</tr>
<tr>
<td>car</td>
<td>99.19</td>
<td>99.07</td>
<td>0.34</td>
</tr>
<tr>
<td>dermatology</td>
<td>98.35</td>
<td>98.35</td>
<td>0.34</td>
</tr>
<tr>
<td>ecoli</td>
<td>90.43</td>
<td>88.64</td>
<td>5.3e-03*</td>
</tr>
<tr>
<td>glass</td>
<td>77.50</td>
<td>75.48</td>
<td>0.33</td>
</tr>
<tr>
<td>iris</td>
<td>98.00</td>
<td>98.00</td>
<td>0.37</td>
</tr>
<tr>
<td>isoleo</td>
<td>97.68</td>
<td>97.19</td>
<td>5.5e-04*</td>
</tr>
<tr>
<td>krkopt</td>
<td>88.26</td>
<td>87.80</td>
<td>1.6e-03*</td>
</tr>
<tr>
<td>letter</td>
<td>98.47</td>
<td>97.98</td>
<td>1.4e-04*</td>
</tr>
<tr>
<td>nursery</td>
<td>98.62</td>
<td>98.53</td>
<td>0.50</td>
</tr>
<tr>
<td>optdigits</td>
<td>99.50</td>
<td>99.41</td>
<td>0.21</td>
</tr>
<tr>
<td>page-blocks</td>
<td>97.31</td>
<td>97.06</td>
<td>0.20</td>
</tr>
<tr>
<td>pendigits</td>
<td>99.79</td>
<td>99.69</td>
<td>3.3e-03*</td>
</tr>
<tr>
<td>satimage</td>
<td>93.29</td>
<td>92.73</td>
<td>1.3e-02*</td>
</tr>
<tr>
<td>statlogsegment</td>
<td>97.84</td>
<td>97.75</td>
<td>0.71</td>
</tr>
<tr>
<td>statlogvehicle</td>
<td>86.85</td>
<td>85.66</td>
<td>6.3e-02*</td>
</tr>
<tr>
<td>vowel528</td>
<td>99.82</td>
<td>99.64</td>
<td>0.34</td>
</tr>
<tr>
<td>vowel990</td>
<td>99.80</td>
<td>99.80</td>
<td>0.34</td>
</tr>
<tr>
<td>waveform</td>
<td>87.08</td>
<td>86.74</td>
<td>0.11</td>
</tr>
<tr>
<td>wine</td>
<td>99.44</td>
<td>98.92</td>
<td>0.34</td>
</tr>
<tr>
<td>yeast</td>
<td>62.87</td>
<td>61.26</td>
<td>2.5e-02*</td>
</tr>
</tbody>
</table>
### 4.3. Results and discussion

Table 4.3: \(pa\)-SVM using \(C\)-SVM and polynomial kernel: Best \(\mu_{CCR}\) with \(\mu_{NSV}\) from the max \(C\) min \(\gamma\) scenario. [\(*\) marker means statistically significant]

<table>
<thead>
<tr>
<th>Dataset</th>
<th>(\mu_{CCR})</th>
<th>(\mu_{NSV})</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(pa)-SVM</td>
<td>standard p-value</td>
</tr>
<tr>
<td>abalone</td>
<td>29.26</td>
<td>27.51</td>
</tr>
<tr>
<td>audiology</td>
<td>86.02</td>
<td>82.67</td>
</tr>
<tr>
<td>balance</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>car</td>
<td>97.57</td>
<td>97.11</td>
</tr>
<tr>
<td>dermatology</td>
<td>98.35</td>
<td>98.35</td>
</tr>
<tr>
<td>ecoli</td>
<td>89.48</td>
<td>87.95</td>
</tr>
<tr>
<td>glass</td>
<td>80.05</td>
<td>77.42</td>
</tr>
<tr>
<td>iris</td>
<td>98.67</td>
<td>98.67</td>
</tr>
<tr>
<td>isolek</td>
<td>97.43</td>
<td>97.14</td>
</tr>
<tr>
<td>krkopt</td>
<td>52.30</td>
<td>52.02</td>
</tr>
<tr>
<td>letter</td>
<td>97.27</td>
<td>96.39</td>
</tr>
<tr>
<td>nursery</td>
<td>83.37</td>
<td>83.34</td>
</tr>
<tr>
<td>optdigits</td>
<td>99.31</td>
<td>99.13</td>
</tr>
<tr>
<td>page-blocks</td>
<td>97.33</td>
<td>97.17</td>
</tr>
<tr>
<td>pendigits</td>
<td>99.69</td>
<td>99.59</td>
</tr>
<tr>
<td>satimage</td>
<td>91.41</td>
<td>90.77</td>
</tr>
<tr>
<td>statlogsegment</td>
<td>97.71</td>
<td>97.49</td>
</tr>
<tr>
<td>statlogvehicle</td>
<td>86.97</td>
<td>86.73</td>
</tr>
<tr>
<td>vowel528</td>
<td>99.45</td>
<td>99.27</td>
</tr>
<tr>
<td>vowel990</td>
<td>99.70</td>
<td>99.70</td>
</tr>
<tr>
<td>waveform</td>
<td>86.96</td>
<td>86.70</td>
</tr>
<tr>
<td>wine</td>
<td>99.44</td>
<td>98.89</td>
</tr>
<tr>
<td>yeast</td>
<td>62.47</td>
<td>60.55</td>
</tr>
</tbody>
</table>

\(\dagger\) denotes cases where some classifiers produce perfect classification, and thus ANOVA analysis of classification rates is unreliable.
4.3. Results and discussion

Table 4.4: pa-SVM using ν-SVM and Gaussian kernel: Best $\mu_{CCR}$ with $\mu_{NSV}$ from the min $\nu$ min $\gamma$ scenario. [* marker means statistically significant]

<table>
<thead>
<tr>
<th>Dataset</th>
<th>$\mu_{CCR}$</th>
<th>$p$-value</th>
<th>$\mu_{NSV}$</th>
<th>$p$-value</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>pa-SVM</td>
<td>standard</td>
<td></td>
<td>pa-SVM</td>
</tr>
<tr>
<td>balance</td>
<td>100</td>
<td>99.84</td>
<td>0.34</td>
<td>121</td>
</tr>
<tr>
<td>car</td>
<td>99.25</td>
<td>99.01</td>
<td>0.17</td>
<td>1246</td>
</tr>
<tr>
<td>dermatology</td>
<td>98.61</td>
<td>97.82</td>
<td>8.2e-02</td>
<td>140</td>
</tr>
<tr>
<td>glass</td>
<td>79.12</td>
<td>72.94</td>
<td>4.0e-03*</td>
<td>303</td>
</tr>
<tr>
<td>iris</td>
<td>98.00</td>
<td>98.00</td>
<td>0.37</td>
<td>37</td>
</tr>
<tr>
<td>isole</td>
<td>97.77</td>
<td>97.13</td>
<td>5.8e-04*</td>
<td>24955</td>
</tr>
<tr>
<td>letter</td>
<td>98.25</td>
<td>97.93</td>
<td>3.7e-04*</td>
<td>80691</td>
</tr>
<tr>
<td>optdigits</td>
<td>99.39</td>
<td>99.25</td>
<td>0.10</td>
<td>3982</td>
</tr>
<tr>
<td>pendifgts</td>
<td>99.64</td>
<td>99.55</td>
<td>3.0e-02*</td>
<td>8497</td>
</tr>
<tr>
<td>satimage</td>
<td>92.90</td>
<td>92.40</td>
<td>2.3e-02*</td>
<td>4618</td>
</tr>
<tr>
<td>statlogsegment</td>
<td>97.79</td>
<td>97.32</td>
<td>0.13</td>
<td>1169</td>
</tr>
<tr>
<td>statlogvehicle</td>
<td>86.49</td>
<td>83.90</td>
<td>2.6e-02*</td>
<td>529</td>
</tr>
<tr>
<td>vowel528</td>
<td>99.82</td>
<td>99.45</td>
<td>0.34</td>
<td>749</td>
</tr>
<tr>
<td>vowel990</td>
<td>99.80</td>
<td>99.80</td>
<td>1.0†</td>
<td>1137</td>
</tr>
<tr>
<td>waveform</td>
<td>87.08</td>
<td>86.64</td>
<td>0.15</td>
<td>4295</td>
</tr>
<tr>
<td>wine</td>
<td>99.44</td>
<td>99.44</td>
<td>0.34</td>
<td>32</td>
</tr>
</tbody>
</table>

† denotes cases where some classifiers produce perfect classification, and thus ANOVA analysis of classification rates is unreliable.

No solution was found because of infeasible parameters for the following datasets: abalone, audiology, ecoli, krkopt, nursery, page-blocks and yeast.
Table 4.5: pa-SVM using ν-SVM and polynomial kernel: Best $\mu_{CCR}$ with $\mu_{NSV}$ from the min $\nu$ min $\gamma$ scenario. [* marker means statistically significant]

<table>
<thead>
<tr>
<th>Dataset</th>
<th>$\mu_{CCR}$ pa-SVM</th>
<th>standard</th>
<th>p-value</th>
<th>$\mu_{NSV}$ pa-SVM</th>
<th>standard</th>
</tr>
</thead>
<tbody>
<tr>
<td>balance</td>
<td>100</td>
<td>100</td>
<td>0.34</td>
<td>115</td>
<td>118</td>
</tr>
<tr>
<td>car</td>
<td>97.00</td>
<td>96.18</td>
<td>0.11</td>
<td>602</td>
<td>356</td>
</tr>
<tr>
<td>dermatology</td>
<td>98.61</td>
<td>98.08</td>
<td>0.17</td>
<td>140</td>
<td>246</td>
</tr>
<tr>
<td>glass</td>
<td>81.58</td>
<td>76.05</td>
<td>5.0e-02*</td>
<td>236</td>
<td>240</td>
</tr>
<tr>
<td>iris</td>
<td>98.67</td>
<td>98.67</td>
<td>0.33</td>
<td>41</td>
<td>89</td>
</tr>
<tr>
<td>isolet</td>
<td>97.54</td>
<td>97.01</td>
<td>1.8e-04*</td>
<td>16904</td>
<td>18954</td>
</tr>
<tr>
<td>letter</td>
<td>95.87</td>
<td>95.59</td>
<td>3.8e-03*</td>
<td>28730</td>
<td>28712</td>
</tr>
<tr>
<td>optdigits</td>
<td>98.95</td>
<td>98.93</td>
<td>0.67</td>
<td>2858</td>
<td>3108</td>
</tr>
<tr>
<td>pendigits</td>
<td>99.38</td>
<td>99.33</td>
<td>0.17</td>
<td>5815</td>
<td>4866</td>
</tr>
<tr>
<td>satimage</td>
<td>90.91</td>
<td>88.61</td>
<td>8.3e-05*</td>
<td>2558</td>
<td>3566</td>
</tr>
<tr>
<td>statlogsegment</td>
<td>97.45</td>
<td>97.19</td>
<td>0.30</td>
<td>777</td>
<td>1374</td>
</tr>
<tr>
<td>statlogvehicle</td>
<td>86.97</td>
<td>83.31</td>
<td>3.7e-04*</td>
<td>447</td>
<td>689</td>
</tr>
<tr>
<td>vowel528</td>
<td>99.45</td>
<td>98.91</td>
<td>8.1e-02</td>
<td>675</td>
<td>911</td>
</tr>
<tr>
<td>vowel990</td>
<td>99.70</td>
<td>99.70</td>
<td>1.0†</td>
<td>1089</td>
<td>1550</td>
</tr>
<tr>
<td>waveform</td>
<td>86.92</td>
<td>86.68</td>
<td>0.37</td>
<td>2807</td>
<td>5104</td>
</tr>
<tr>
<td>wine</td>
<td>99.44</td>
<td>99.44</td>
<td>0.34</td>
<td>35</td>
<td>90</td>
</tr>
</tbody>
</table>

† denotes cases where some classifiers produce perfect classification, and thus ANOVA analysis of classification rates is unreliable.

No solution was found because of infeasible parameters for the following datasets: abalone, audiology, ecoli, krkopt, nursery, page-blocks and yeast.
4.3. Results and discussion

4.3.1 \textit{pa}-SVM mostly outperforms standard SVM for multi-class classification

As I examined the $\mu_{CCR}$ columns in Tables 4.2 to 4.5, the overall performance of the \textit{pa}-SVM approach is better because there is no particular mean rate in the standard SVM approach that is higher than the mean rate of the \textit{pa}-SVM approach.

4.3.2 Infeasible parameters evident for $\nu$-SVM

However, when I examined the classification rates using the $\nu$-SVM there are several rows with missing results. This is because there was at least one fold where there were no feasible parameters that satisfy a particular pair of classes during the training stage of a binary classifier. The act of tracking infeasible parameters has been taken into account by my experiments.

4.3.3 Statistical significance defined by \textit{p}-values

A two-way ANOVA study \cite{213}, which compares the \textit{pa}-SVM and standard approach (by taking into account the fact that they are based on the same partitions of the 10-folds) reveals that the \textit{pa}-SVM approach is capable of achieving statistical significance on low performing datasets. Table 4.6 lists the datasets of the different SVM types and kernels that returned a \textit{p}-value less than 0.05, denoting statistical significance.

Table 4.6: List of datasets with rates that are statistically significant

<table>
<thead>
<tr>
<th>C+Gaussian</th>
<th>C+poly.</th>
<th>$\nu$+Gaussian</th>
<th>$\nu$+poly.</th>
</tr>
</thead>
<tbody>
<tr>
<td>abalone</td>
<td>abalone</td>
<td>glass</td>
<td>glass</td>
</tr>
<tr>
<td>audiology</td>
<td>audiology</td>
<td>isole</td>
<td>isole</td>
</tr>
<tr>
<td>ecoli</td>
<td>car</td>
<td>letter</td>
<td>letter</td>
</tr>
<tr>
<td>isole</td>
<td>isole</td>
<td>pendigits</td>
<td>pendigits</td>
</tr>
<tr>
<td>krkopt</td>
<td>krkopt</td>
<td>satimage</td>
<td>satimage</td>
</tr>
<tr>
<td>letter</td>
<td>letter</td>
<td>statlogvehicle</td>
<td>statlogvehicle</td>
</tr>
<tr>
<td>pendigits</td>
<td>pendigits</td>
<td>yeast</td>
<td>yeast</td>
</tr>
<tr>
<td>satimage</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>statlogvehicle</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>yeast</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The statistics on the remaining datasets were inconclusive between the
approaches because the $p$-values were above 0.05; however, the estimates (i.e. the difference in the mean rate between approaches) favoured the $pa$-SVM method.

### 4.3.4 Favourable results compared to recent studies

A total of 252 datasets across 31 recent studies were compared and Table 4.7 summarises the 20 results from the 13 studies that gave a better $\mu_{CCR}$. In a previous publication [97], I compared results from all four min/max scenarios using the $C$-SVM and Gaussian kernel against the results from the literature. I have revised Table 4.7 to show the figures based on the scenario that returned the lowest number of support vectors using the best of $C$-SVM and $\nu$-SVM with either the Gaussian or polynomial kernel. The table lists each dataset’s name and the number of datasets tested that is same as mine. The name of the base learners and the framework were included for comparison, as well as the fold used for cross validation. The final column is the improvement difference, which is calculated by the $\mu_{CCR}$ rate subtracting the $pa$-SVM results of Tables 4.2-4.5.

A general overview of Table 4.7 is that the number of datasets that have outperformed the $pa$-SVM’s method to the number of datasets taken into account by each study are very small. I now examine the factors that may have contributed towards these improvements.

The first notable bias in the improvement difference of Table 4.7 can be caused by the author’s choice of fold for cross validation; followed by their use of a training and test set. For instance, [130] gave the highest improvement difference of 23.23% for the yeast dataset and 1.43% for the satimage dataset. Their parameters for the Gaussian kernel were optimised by a 5-fold cross validation method, and the authors divided the yeast and satimage datasets into training and test sets (the split ratio for the satimage dataset followed that of the source [64]). Similarly, the results of [214] have used a separate training and validation set for the segment dataset; however, a cross validation method was not clearly described. 38% of Table 4.7 used a fold other than the standard 10 fold, and 23% of the table divides a dataset into training and testing sets. In some recent examples, the dataset is split into training, validation and test sets [197, 205]. Looking further into the folds of Table 4.7, 8% use a 5×2-fold and 15% use a 5-fold. Studies such as [180] have shown that a 5 or 10-fold in a cross validation method has a less biased accuracy than a 2-fold; yet a 2-fold [80, 225] and 3-fold [201] have been used in recent
Table 4.7: A small number of recent studies that gave a better result shown by the (diff.) column, which is the improvement difference and sorted by the number of datasets tested that is same as mine.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#</th>
<th>base learners</th>
<th>framework</th>
<th>fold</th>
<th>diff.</th>
</tr>
</thead>
<tbody>
<tr>
<td>satimage</td>
<td>3</td>
<td>SVM vector projection</td>
<td>majority</td>
<td>5</td>
<td>1.43\dagger</td>
</tr>
<tr>
<td>yeast</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>23.23\dagger</td>
</tr>
<tr>
<td>waveform</td>
<td>4</td>
<td>RBF SVM</td>
<td>one-vs-one</td>
<td>10</td>
<td>4.26</td>
</tr>
<tr>
<td>segment</td>
<td>5</td>
<td>Random Forest</td>
<td>majority</td>
<td>NA\textsuperscript{1}</td>
<td>0.37\textsuperscript{2}</td>
</tr>
<tr>
<td>wine</td>
<td>5</td>
<td>(HDA/CDA+Q)\textsuperscript{3}</td>
<td>all at once</td>
<td>10</td>
<td>0.01</td>
</tr>
<tr>
<td>iris</td>
<td>7</td>
<td>(LBDA+NN)\textsuperscript{4} binary SVMs</td>
<td>distances</td>
<td>10</td>
<td>0.47\textsuperscript{1}</td>
</tr>
<tr>
<td>wine</td>
<td>7</td>
<td>Laplacian SVM SRSVM\textsuperscript{5}</td>
<td>one-vs-all</td>
<td>2\dagger</td>
<td>0.43</td>
</tr>
<tr>
<td>pen digits</td>
<td>8</td>
<td>&amp;NN</td>
<td>weighted</td>
<td>10</td>
<td>0.2</td>
</tr>
<tr>
<td>majority</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>car</td>
<td>9</td>
<td>SVM binary tree</td>
<td>weighted\textsuperscript{5}</td>
<td>10</td>
<td>0.81</td>
</tr>
<tr>
<td>vehicle</td>
<td>10</td>
<td>ECOC+ Linear SVM</td>
<td>RE-ECOC\textsuperscript{7} RE-Sparse\textsuperscript{8}</td>
<td>10</td>
<td>4.74</td>
</tr>
<tr>
<td>car</td>
<td>11</td>
<td>(Fuzzy rules + SVM)\textsuperscript{9}</td>
<td>majority</td>
<td>5</td>
<td>0.81</td>
</tr>
<tr>
<td>dermatology segment</td>
<td>13</td>
<td>Recursive ECOC via shared binary SVMs</td>
<td>one-vs-one one-vs-all</td>
<td>10</td>
<td>0.69</td>
</tr>
<tr>
<td>opt digits</td>
<td>13</td>
<td>(ECOC + LW)\textsuperscript{10}</td>
<td>sparse coding</td>
<td>10</td>
<td>0.32</td>
</tr>
<tr>
<td>nursery segment</td>
<td>19</td>
<td>dichotomy trees forests of nested</td>
<td>weighted Ada/Multi boost</td>
<td>5×2</td>
<td>0.36</td>
</tr>
</tbody>
</table>

1. 80% of data were randomly selected for training and 20% for validation.
2. random forest outperformed SVM, however the fold used was not described.
4. Linear boundary discriminant analysis + 1 nearest neighborhood classifier.
5. Structural Regularised SVM.
6. The weighting criteria of the tree uses Fisher’s Discriminant Ratio (FDR) between classes.
7. Re-coded problem dependent ECOC design.
8. Re-coded to sparse random ECOC design.
9. PDFC (Positive definite fuzzy classifier) = fuzzy rule-based where rules are extracted from trained SVM.
10. ECOC + Loss-Weighted Decoding.
\dagger. Cross validation was used to obtain parameters while a training and test set produced the result.
\dagger. Each dataset is divided randomly into training and test set, where each set contains almost half the samples in each class.
4.3. Results and discussion

studies. The differences in the cross validation fold and particularly the use of a training and test or validation set makes a direct comparison difficult. I note that recent work suggests that a nested cross validation algorithm can provide a closer estimate to the true error rate [23]. This path is open for further exploration, but it was necessary to use a $k$ fold cross validation algorithm in order to compare with previous studies using the same datasets.

Apart from the previous discussion of the results regarding [130], the next 2 entries with the highest improvement difference use SVMs as their base learner. The first is the waveform dataset from [212]. The same RBF kernel was used, however the authors selected the kernel parameters experimentally by a heuristic rule. This gave one of the better results in Table 4.7 but the authors described that a suboptimal result was achieved for one of their datasets, which they were unable to explain. The second entry is the vehicle dataset from [59] that uses an Error Correcting Output Code (ECOC) classifier with a linear SVM. The improvement difference worked particularly well for that dataset, however only 2 of the 10 same datasets showed an improvement over the $pa$-SVM approach.

The next issue to note from Table 4.7 is the number of entries using similar framework and base learners. The one-vs-one/majority voting framework makes up 46% of the table with SVMs taking up 62%. These figures reflect and reinforce the choices I made for the framework and base learner of my $pa$-SVM. Lastly, I single out the iris dataset from Table 4.7; the focus is not on the 2 places it has on the table but its 150 sample size to outline that SVMs are less affected by small training data and can achieve very good results [59, 152, 225].

4.3.5 Proper selection of $(C, \gamma)$ and $(\nu, \gamma)$ values can reduce the number of support vectors

A suitable selection of $(C, \gamma)$ and $(\nu, \gamma)$ can lead to a SVM classifier with a low number of support vectors. A low number of support vectors is desirable for achieving good generalisation and computational efficiency [36, 129]. For most results in Tables 4.2-4.5 the $\mu_{NSV}$ for $pa$-SVMs is much lower than that for the standard SVMs if $(C, \gamma)$ and $(\nu, \gamma)$ are chosen appropriately.

The current parameter selection process uses a grid search, where the values for the $C$-SVM are similar to [32] with their LIBSVM library, while the $\nu$ uses an increment of 0.05 with the $\gamma$ values being the same. I acknowledge that the grid search is an exhaustive approach; however, it is a sensible choice
4.3. Results and discussion
to understand how a new method behaves through the parameter space. Then
at a later stage, methods such as heuristics can be experimented with to speed
up the parameter selection process.

My analysis of the number of support vectors amongst the four min/max
scenarios show that for pa-SVM the $\mu_{NSV}$ varies much more than for standard
SVMs. The reason is in how the parameters $[(C, \gamma) \text{ for } C\text{-SVM and } (\nu, \gamma) \text{ for } \nu\text{-SVM}]$ were selected between the approaches for each binary classifier. In
the standard SVM approach, the parameters for each binary classifier are
fixed, so there can be no multiple parameters occurring. The observation of a
low $\gamma$ returning fewer support vectors agrees with studies such as [69] where
the authors observed an increase in the number of support vectors as $\gamma$ was
increased.

4.3.6 Computational complexity and practicality of the pa-
SVM

In Section 3.11, I mentioned that the computational complexity of training an
SVM is $O(m^3)$ where $m$ is the number of training samples, and for the testing
phase the computational complexity of an SVM is linear to the number of
support vectors and constant to the number of test samples.

For pa-SVM the computational complexity is similar. The reason is be-
cause the number of iterations through parameter space is the same, i.e. the
number of iterations through parameter space for the standard SVM is defined
by $\text{number of folds} \times \text{number of parameters} \times \text{number of binary classifiers}$
while for the pa-SVM the number of iterations through parameter space is
defined by $\text{number of folds} \times \text{number of binary classifiers} \times \text{number of param-
eters}$. The result are the individual parameters returned by pa-SVM for each
binary classifier.

Given that most of the datasets tested the pa-SVM returned a lower num-
ber of support vectors when compared to the standard SVM, the computa-
tional complexity in the testing phase is thus faster. This promotes the
practicality of the pa-SVM approach.

4.3.7 Running time

The experiments were ran across a number of 64 bit Microsoft Windows work-
stations and servers with single and multicore Intel processors and a large
amount of memory (ranging from 8GB to 48GB of RAM). Because of this
setup a benchmark on the running times can not be compared accurately.
The general overview was that the results for most datasets were achieved within a week. The running time for the top three datasets with the largest number of samples (krkopt, letter and nursery) were notably longer. Their running times range from a month (nursery) to three months (krkopt), and the pa-SVM method was notably faster than the standard SVM method (up to 60% faster).

The long running times do hinder on the practicality of using SVMs. However, recent SVM methods using parallel computation on the GPU (‘Graphics Processing Unit’) has shown to improve the speed of SVMs up to 98.9× [128], 121× [37], 112.19× [93], and 129.64× [127] faster. I did not implement parallel computation on the GPU because the focus of the investigation was on improving precision and not the running time. However, it is apparent that implementing parallel computation on the GPU will dramatically improve the speed of training SVMs and as a result will improve on the practicality of using pa-SVMs.

4.4 Summary

The results of my experiments support my hypothesis that refining the pairwise concept leadsto an improved classification performance. This pairwise concept was used to outline three core components of a multiclass classifier. The first is the framework, the second is the strategy used within the framework to determine the class, and third is the learning classifier. From the literature review I identified the one-vs-one framework, the majority voting strategy and the Support Vector Machine (SVM) learning classifier as the preferred options for these components. However, this may not be an optimal solution because current SVMs used in practice have parameters that are one and the same for all its binary classifiers. The proposed pairwise refinement is to make each binary classifier within the multiclass framework have optimised parameters for its task. To compare with other recent methods, a number of datasets from the UCI Machine Learning Repository were chosen for benchmarking, and to maintain a consistent low biased classification accuracy, a standard 10-fold cross validation method was used throughout my experiments.

My experimental results show that the proposed pairwise adaptive method mostly outperforms the standard approach. Because of the way I evaluate the best parameters for each binary classifier, multiple pairs of parameters with the same best rate frequently occur. To evaluate multiple parameters of a binary classifier with the same best correct classification rate, I expanded
my experiments to cover four scenarios of min/max $C$ and min/max $\nu$ with min/max $\gamma$. The explored scenarios showed that the total number of support vectors can be minimised by selecting the max $C$ min $\gamma$ of $C$-SVM and min $\nu$ min $\gamma$ of $\nu$-SVM when the best classification rate of a binary classifier has more than one parameter pair. However, for some datasets the $\nu$-SVM has regions of infeasible parameters where no solution can be found.

My comparison with other recent methods shows that the new method is very competitive. The main concern with the comparison was in regard to studies that used a separate training and test set instead of a cross validation method, and others that used a different cross validation fold altogether. The number of datasets from the very few studies that returned a better result was small, and those with a marginal increase did use a separate training and test set.

I foresee my future work to include a parallel analysis to assess the improvements in speed as a result of this current refinement, as well as the extensibility of this study to a different framework (such as one-vs-all or a weighted strategy), and the generalisability of this classification technique using a different learning classifier.

Concluding the chapter summary, I have presented a refined framework for multiclass classification using two soft margin Support Vector Machines. In subsequent chapters further experiments using this refined framework will be explored and analysed to validate its robust performance.
Chapter 5

Facial Expression Classification

The contents of this chapter have been published in [96, 98].

Contents

5.1 Introduction ........................................... 68
5.2 System overview ....................................... 70
5.3 Dataset and image preprocessing ......................... 70
  5.3.1 Proposed face model ............................... 72
  5.3.2 Holistic approach ................................. 73
  5.3.3 Component approach ............................... 74
5.4 Classification of facial expressions ....................... 76
  5.4.1 Support Vector Machines (SVM) and kernels .......... 77
  5.4.2 The SVM and kernel parameters to optimise .......... 77
  5.4.3 Proposed pairwise adaptive Multiclass SVM .......... 77
  5.4.4 Multiclass classification ........................... 77
  5.4.5 Feature vectors ..................................... 77
5.5 Results and discussion ................................ 78
  5.5.1 Performances under cropping and different image preprocessing ........................................... 78
  5.5.2 Component is the better approach .................... 81
  5.5.3 Excluding the contemptuous expression and neutral does impact performance ............................... 82
  5.5.4 Why the contemptuous expression is considered universal .................................................... 84
  5.5.5 Proposed SVM mostly outperforms standard SVM ................................................................. 84
  5.5.6 Is there an optimal parameter pair? .......................... 84
  5.5.7 Performance dynamics of expression pairs suggest a non equidistant expression space ....................... 86
  5.5.8 Favourable results compared to other studies .......... 87
Recapping my thesis question, ‘Can a machine describe the emotional response to a scene using perceived faces and facial expressions?’, the question has two integrated parts; finding faces and determining their expression. Since I have started by presenting the classification method, I follow through with the classification of facial expressions. I would like to point out that the pairwise adaptive selection of SVM model parameters described in this chapter uses an earlier version of the method, which later matured to a method that I called ‘Pairwise Adaptive SVM’ (Chapter 4).

In this chapter, I examine the classification dynamics of Ekman’s seven Discrete Universal Facial Expressions of Emotion – happy, sad, surprised, fearful, disgusted, angry, and contemptuous – plus the neutral face. The aim is to provide a comparative bridge so that studies that differ by the number of discrete expressions, choice of dataset and the number of different preprocessing techniques can be compared. To realise this aim my experiments use a number of image preprocessing techniques (greyscale, Local Binary Patterns, Sobel and Canny edge detection, and manually inserted feature points, which I refer to as components) that are organised into four approaches: holistic, holistic action, component and component action (where an action approach is produced by a difference between a neutral and an expressive face). The facial expressions are classified using a Gaussian kernel paired with a standard and a proposed multiclass Support Vector Machine (SVM) that uses pairwise adaptive model parameters. The significance of this study is that it provides a thorough understanding of the choices that impact a classifier’s performance on the seven Universal Facial Expressions and the performance dynamics between expression pairs. A further contribution is my proposed face model, which uses feature points around the eyes, brows and mouth to provide a best correct classification rate of 98.57%. An important emphasis in my discussion is where I justify the use of the contemptuous expression (an expression that has rarely been explored in the machine learning literature) by noting its universal presence across human cultures, and its asymmetric nature, which is not shared with other expressions.
5.1 Introduction

Facial expression classification has gained popularity because of its potential for applications in human computer interaction (HCI) – where the ability to automate the capture of emotions, personality traits and cognitive intentions from facial expressions [52] would be a significant advancement [107]. However, studies of facial expression classification differ by the number of discrete expressions, choice of dataset and the number of different processing techniques, which makes comparison of results between studies difficult [60, 94, 231]. In this chapter I present these choices leading to my proposal.

Happy, sad, surprised, fearful, angry and disgusted are the typical six discrete expressions that have been used in many facial expression studies [124, 141, 220]; only a few studies have included a contemptuous expression [8, 25, 139, 182]. Paul Ekman and Wallace V. Friesen have described how these seven expressions are universally recognised [46, 54], and have appropriately named them the ‘universal facial expressions of emotion’. In addition, they have devised a widely used system of metrics known as ‘The Facial Action Coding System’ (FACS) which captures facial actions that are applicable to facial expressions [48]. Returning to the choices of discrete expressions, the neutral face has been included by some researchers as an additional expression to classify [81, 123, 124, 132, 134, 136, 141, 219, 224] while others analyse the difference images that are produced by subtracting the neutral face from an expressive face [13, 17, 166, 196, 217, 235].

Next, is the dataset to consider; this is an important choice because the number of discrete expressions to explore is limited by those expressions covered. For instance, the JAFFE [141], Cohn-Kanade [114], FEEDTUM [216] and MMI [163] are common datasets that have been used in [196], [33, 206, 220, 236, 238]), [116, 217], and [196, 220], respectively. Each of these datasets cover the typical six expressions plus the neutral face. The exception is the new extended Cohn-Kanade dataset, where the contemptuous expression is also covered [139]. Other datasets such as the Yale [71], AR [144], and SFED07 [164] datasets have been used (to a lesser extent) in [166] and [224] – these datasets cover a subset of the typical six discrete expressions [166] and have additional expressions that are different [224]. Out of all the datasets mentioned, only the Cohn-Kanade [114] and MMI [163] have cited certification of their face images using facial expression experts.

Given a chosen dataset, a number of image preprocessing techniques usually occur; the choices can be overwhelming because of the many different
5.1. Introduction

methods available. For instance, edge detection [35, 172] and histogram equalisation [33, 123, 224] have been used to counter the effects of noise from poor lighting. The method of Local Binary Patterns (LBP) has also become popular for tackling poor lighting, and has shown promising results in facial expression classification [132, 196]. Further, wavelet compression techniques have been used to reduce noise by taking into account the spatial information of the face – i.e. Haar-like wavelets using the Integral Image [215], and Gabor wavelets have been used in [236] and [123, 132, 134, 136, 141], respectively. The image preprocessing techniques described so far operate over the full image. I refer to this procedure as a holistic approach, and this term is extended to a holistic action approach when several full images are used to produce a final image – i.e. the previously described difference images [11, 42], and optical flow fields which are computed by the motion vector of pixels from an image sequence [206, 229].

Another popular image preprocessing technique is the use of feature points (a.k.a landmarks) on a face image. I refer to this as a component approach, and this term is extended to a component action approach when the difference between the feature points of the expressive and neutral face are computed. One of the main benefits of a component approach is the ease of normalising a face dataset where the feature points are used for alignment. An example of a component approach is the Active Appearance Model (AAM) [124] used for classifying facial expressions [44, 139], and when the model is used for classifying the intensity of facial expressions [17, 166] I consider it to be a component action approach. A major issue with the component and the component action approach is deciding where and how many feature points on a face is sufficient. For instance, [14, 196, 236] chose the eyes, [224] chose the eyes and middle of the nose, [166] chose the eyes and nose tip, [207] chose the eyes and lateral nose corners, [125] chose the eyes and philtrum, and [134] chose the eyes, nose and mouth.

The literature discussed so far describes the many choices that a researcher is required to make to carry out their study. I have shown that some of these studies differ in the number of discrete expressions, the dataset used and the number of different preprocessing techniques – this makes comparison of results between studies difficult. A thorough study that can provide a comparative bridge among these studies is required. This leads to my proposal. My aim is to provide this comparative bridge by examining the classification dynamics of the seven universal facial expressions of emotion using a number of image preprocessing techniques, organised into four approaches: holistic, holis-
tic action, component and component action. The significance is to provide a thorough understanding of the choices that can impact a classifier’s performance on the seven universal facial expressions and the performance dynamics between expressive pairs. For the classification task I propose a modified multiclass Support Vector Machine (SVM) which uses pairwise adaptive model parameters, that I called a ‘Pairwise Adaptive Support Vector Machine’ (pa-SVM) [97]. I compare this to a standard multiclass SVM with the hypothesis that a component based approach will outperform a holistic approach despite the ongoing interest in holistic approaches. The modified multiclass Support Vector Machine (SVM) used in this chapter has been refined and tested on several datasets [97].

A system overview is presented in the next section, followed by a description of the dataset and image preprocessing used in Section 5.3. Section 5.4 presents the classification method followed by the experimental results and discussion in Section 5.5. I provide my concluding remarks in Section 5.6.

5.2 System overview

My system consists of several steps as illustrated in Figure 5.1. The first step is to manually insert feature points for each face in the dataset so that the faces can be aligned. After image preprocessing I have four datasets: greyscale, Local Binary Patterns, edges using Canny and Sobel operator, and feature points. The differences between the expressive and neutral face are produced for each dataset. The resultant datasets are then organised into four approaches: holistic, holistic action, component and component action. Details of each module and further steps are explained in subsequent sections.

5.3 Dataset and image preprocessing

In this section I describe the face dataset, how the images were aligned using my proposed face model, and which image preprocessing techniques were used.

The face dataset was generated using the JACFEE and JACNEUF datasets, which are certified by the Paul Ekman Group LLC [167]. I specifically chose these datasets because they have discrete expression classes (the seven universal facial expressions of emotion as well as neutral faces). The generated dataset consists of 280 coloured face images, including 140 neutral faces and 20 images of seven different facial expressions. 60 feature points
Figure 5.1: System overview showing the flow of the stages and states.
were manually inserted for each face image along the brows, eyes and mouth. These feature points became my proposed face model, and I use the two feature points of the inner eye corners (a.k.a lacrimal caruncle) for aligning the face dataset. Once aligned the images are cropped and resized to 100x100 pixels for image preprocessing. The results of a face crop and a head crop are obtained and discussed in Section 5.5.

5.3.1 Proposed face model

Returning to my face model, an illustration is provided in Figure 5.2. In this illustration the feature points are coloured red, blue and green, representing respectively a stable, active or passive state during an expression. The reasons for my chosen feature points are as follows:

1. **Stable**: I chose the inner eye corners, otherwise known as the lacrimal caruncle because they are invariant across facial expressions, and as they are stable points they are ideal for aligning the faces in the dataset.

2. **Active**: I chose the brow end points, mid-upper and mid-lower eyelids, lip end points, and mid-upper and mid-lower lips because I consider them to be the minimal set of points required to describe and distinguish between expressions – i.e. the positions of these points differ greatest across the different expressions.

3. **Passive**: I consider passive points to be the points that are positioned between active points. Similar to active points, they do vary between expressions, however they are not essential in separating one expression from the next. They are there to enhance the shape of the brow, eye and mouth movements.

**Affine transformations**: The individual face images are aligned via translation, rotation and scaling using the inner eye corners such that they lie on the horizontal $x$-axis and the midpoint between the inner eye corners sits on the origin of the transformed plane. The transformation matrix is as follows:

$$
\text{normalise}(x, y) = \text{scale}(x, y) \times \text{rotate}(x, y) \times \text{translate}(x, y)
$$

$$
\text{translate}(x, y) = \begin{bmatrix}
-\frac{x_L+x_R}{2} \\
-\frac{y_L+y_R}{2}
\end{bmatrix}
\begin{bmatrix}
  x \\
  y
\end{bmatrix}
$$
5.3. Dataset and image preprocessing

Figure 5.2: Face model showing the feature points [1=stable, 2=active, and the remainders are passive], facial actions and directions of horizontal and vertical expansion and contraction.

where $x_L, x_R, y_L, y_R$ are the left and right inner eye corners.

\[
\text{rotate}(x, y) = \begin{bmatrix}
\cos(-\theta) & -\sin(-\theta) \\
\sin(-\theta) & \cos(-\theta)
\end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}
\]

where $\theta$ is the angle between the interval joining the inner eye corners and the horizontal $x$-axis.

\[
\text{scale}(x, y) = \frac{1}{2x_R} \begin{bmatrix} x \\ y \end{bmatrix}
\]

where $x_R$ is right eye corner.

5.3.2 Holistic approach

Four image preprocessing techniques were used: greyscale, Local Binary Patterns (LBP), edges and feature points. As feature points have been described previously, a description of the first three is as follows:
5.3. Dataset and image preprocessing

**Greyscale**

The greyscale uses an ITU-R 601-2 luma transform on RGB (red green blue) colours [1]:

\[
greyscale = \text{red} \times 0.299 + \text{green} \times 0.587 + \text{blue} \times 0.114
\]

**Local Binary Patterns**

The Local Binary Pattern (LBP) uses a 3x3 neighborhood LBP operator [196]; and is a non-parametric transform.

**Edges using Canny and Sobel**

The Canny and Sobel edge operators were used for the edge images. Both operators have several parameters that impact the final edge image. Selection of the Sobel and Canny parameters was by visual evaluation for ideal face edges. The parameters and the ranges are as follows—lower threshold: 85 [0-255] and upper threshold: 170 [0-255] for Sobel and Canny, and for Sobel only a blur of 0 [0-50] and a gain of 5 [1-10].

The greyscale, Local Binary Patterns and edges represent the holistic approach. To obtain the holistic action approach I subtract the neutral faces from the expressive faces. Examples of the holistic and holistic action approach are shown in Figure 5.3. In the figure, I showed the average of expressions from rows 1 to 8 in their respective preprocessed images, and the alignment using the inner eye corners is evident by the clarity around that area. Rows 1-5 show a face crop, while rows 6-8 show a head crop and the holistic action approach where the difference images between an expressive and a neutral face are averaged. The contemptuous expression in row 1 shows the visual asymmetry of this facial expression.

5.3.3 Component approach

My previously described face model and its feature points form the component approach. To obtain the component action approach I use the differences between the feature points of an expressive and a neutral face, which I refer to as facial actions.

The facial actions assigned to each feature point are shown in Table 5.1 and illustrated in Figure 5.2. In the Table (5.1), I describe four primary facial actions: left, right, up and down; and secondary facial actions: horizontal and
Figure 5.3: Sample images of the holistic and holistic action datasets. The average of each expression is shown from rows 1 to 8 in their respective preprocessed images. The use of the inner eye corners for alignment of faces is evident by the clarity around that area. Rows 1-5 show a face crop, while rows 6-8 show a head crop and the holistic action approach where the difference images between an expressive and a neutral face are averaged. The contemptuous expression in row 1 shows the visual asymmetry of this facial expression. The face dataset was generated from the JACFEE and JACNEUF datasets [167].
Table 5.1: Facial actions assigned to the feature points of the face model in Figure 5.2. When I refer to the upper or lower lip I mean both the inner edge and outer edge of the lip.

<table>
<thead>
<tr>
<th>Feature points</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>brow endpoints</td>
<td>left, right, up, down</td>
</tr>
<tr>
<td>midpoint of upper &amp; lower eyelids</td>
<td>up, down</td>
</tr>
<tr>
<td>mouth corner points</td>
<td>up, down, left, right</td>
</tr>
<tr>
<td>midpoint of inner edge upper lip</td>
<td>up, down</td>
</tr>
<tr>
<td>midpoint of inner edge lower lip</td>
<td>up, down</td>
</tr>
<tr>
<td>inner edge upper lip</td>
<td>horizontal contr./expansion</td>
</tr>
<tr>
<td>inner edge lower lip</td>
<td>horizontal contr./expansion</td>
</tr>
<tr>
<td>midpoints of upper lip</td>
<td>vertical contr./expansion</td>
</tr>
<tr>
<td>midpoints of bottom lip</td>
<td>vertical contr./expansion</td>
</tr>
</tbody>
</table>

vertical contraction and expansion. The horizontal contraction and expansion uses the difference of the sum of the distances between the active and passive feature points along the inner edge of the lips; while the vertical contraction and expansion uses the difference of the distance between the midpoints of the inner and outer edges of the lips. The horizontal and vertical facial actions were introduced to describe the mouth more clearly as it has a higher degree of movement than the brows and the eyes. The equation is as follows:

$$\text{diff} = \sum_{i=0}^{n-1} (E_{i+1} - E_i)^2 - \sum_{i=0}^{n-1} (N_{i+1} - N_i)^2$$  \hspace{1cm} (5.1)

where $E_{i,i=0,...,n-1}$ represents the feature points of an expressive face, $N_{i,i=0,...,n-1}$ represents the feature points of a neutral face, and diff indicates an expansion if $\text{diff}>0$ and a contraction if $\text{diff}<0$.

5.4 Classification of facial expressions

The literature review in Section 3.1.2 presents the reasons why I have chosen SVM as my preferred classifier for evaluating facial expressions. Specifically, I have chosen the $C$-SVM.
5.4. Classification of facial expressions

5.4.1 Support Vector Machines (SVM) and kernels

From the literature review of kernels with SVMs in Section 3.2.2, I have chosen to conduct my experiments using a Gaussian kernel.

5.4.2 The SVM and kernel parameters to optimise

From the theories of SVM and kernels in Chapter 3 and specifically Sections 3.7-3.8, my choice of the C-SVM with a Gaussian kernel has a pair of parameters where tuning is required. They are \((C, \gamma)\), where \(C\) is a parameter for the C-SVM and \(\gamma\) is a tuning parameter for the Gaussian kernel.

5.4.3 Proposed pairwise adaptive Multiclass SVM

A pseudocode of my \(pa\)-SVM method is described in Section 4.2.6. The main difference is the experiments of this chapter use the grid search and cross validation algorithm provided by the LIBSVM library. For the remaining chapters the grid search and cross validation algorithm are as described in Chapter 4.

5.4.4 Multiclass classification

A description of two primary frameworks for multiclass classification is provided in Section 3.12. I use the one-against-one approach because of its suitability for application to a large multiclass dataset [31].

5.4.5 Feature vectors

Four feature vectors were defined for each of the approaches. The holistic and holistic action approaches have a feature vector of dimension \(100 \times 100\) using the image intensities; the component approach has a feature vector of dimension \(120\) using the \(xy\) coordinates of the 60 feature points, and the component action approach has a feature vector of dimension \(44\) covering the facial actions described in Table 5.1. Results were also obtained for a cut down \(36\) dimension feature vector of the component action approach where only the left, right, up, and down actions are used – i.e. the horizontal and vertical expansions and contractions have been removed.

With the exception of the component approach all other feature vectors were normalised between -1 and +1.
results and discussion

An overview of the experimental results is shown in Table 5.2 which summarises the numerical results of the holistic approach in Table 5.3 and the component approach in Table 5.4. Looking at the summary Table (5.2), we can see where performances vary among different image preprocessing of different approaches with and without the contemptuous expression. I discuss the results of these tables further in subsequent subsections by first presenting my findings of the different image preprocessing techniques, then discussing which approach is better, followed by the impacts to the classification rate when neutral and the contemptuous expression are excluded. Next, I discuss how my proposed multiclass SVM compares to the standard method, then present the averaged performance dynamics of all the expressive pairs of my proposed SVM method to suggest a weighted expression space. Following this, I discuss how my results compared with other studies of discrete expressions and the impact of using different image resolutions. The final section discusses my proposed face model, and lastly, a possible future direction for my face model for intensity analysis of expressions and identification.

5.5.1 Performances under cropping and different image preprocessing

The first notable impact is that cropping closer to the face returns better results, as shown by the higher rates of the holistic and component approaches in Table 5.3 and 5.4. Returning to the summary Table (5.2) the top performer for the holistic approaches was greyscale, except when the neutral face was excluded. However, greyscale, Sobel, Canny and LBP cover only the holistic approaches.

The feature points of the component approach returned the best rates. One reason is the minimal noise when compared to other preprocessing techniques, and no requirement for tuning of parameters. However, the feature points need to be manually inserted. On the other hand, tuning the edge operator’s parameters is required in the holistic approach and it affects the visual clarity of the facial features of the final edge image.

As there is currently no literature on determining the optimal parameters for edge filtering on face images, my logic was to observe the edges of the facial features as the upper and lower thresholds (of the Canny and Sobel edge operator) changed in ratios of 1:1, 1:2, 1:3 and so on. This gave an initial
Table 5.2: Overview of the experimental results. We can see where performances vary among different image preprocessing of different approaches with and without the contemptuous expression. A summary of the overall performances is shown at the bottom of the table.

<table>
<thead>
<tr>
<th>Approaches</th>
<th>Include contempt</th>
<th>Exclude contempt</th>
</tr>
</thead>
<tbody>
<tr>
<td>holistic</td>
<td>Include neutral: greyscale &gt; Sobel &gt; Canny ≥ LBP</td>
<td>Include neutral: greyscale &gt; Sobel &gt; Canny ≥ LBP</td>
</tr>
<tr>
<td></td>
<td>Exclude neutral: Sobel &gt; LBP &gt; Canny &gt; greyscale</td>
<td>Exclude neutral: LBP &gt; Canny &gt; Sobel</td>
</tr>
<tr>
<td>holistic action</td>
<td>greyscale &gt; LBP &gt; Sobel &gt; Canny</td>
<td>greyscale &gt; LBP &gt; Canny &gt; Sobel</td>
</tr>
<tr>
<td>component</td>
<td>exclude neutral &gt; include neutral</td>
<td>include horizontal and vertical expansion and contraction returns higher result</td>
</tr>
<tr>
<td>component action</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Summary</td>
<td>cropping: face crop &gt; head crop</td>
<td></td>
</tr>
<tr>
<td></td>
<td>contempt: exclude &gt; include</td>
<td></td>
</tr>
<tr>
<td></td>
<td>neutral: exclude &gt; include</td>
<td></td>
</tr>
<tr>
<td></td>
<td>approaches: component action &gt; component &gt; holistic &gt; holistic action</td>
<td></td>
</tr>
</tbody>
</table>
### 5.5. Results and discussion

Table 5.3: The mean correct classification rates of holistic approaches using a 10 fold cross validation technique. The best mean correct classification rate of each approach is in bold. Lower rates returned by my proposed method are marked by the (*) symbol.

<table>
<thead>
<tr>
<th>crop</th>
<th>preprocessing</th>
<th>include contempt</th>
<th>exclude contempt</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>standard %</td>
<td>proposed %</td>
</tr>
<tr>
<td>Holistic – reported % are split into include neutral/exclude neutral</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>face</td>
<td>Gray</td>
<td>62.14/50.71</td>
<td>65.71/53.57</td>
</tr>
<tr>
<td></td>
<td>LBP</td>
<td>51.79/57.14</td>
<td>51.79/57.86</td>
</tr>
<tr>
<td></td>
<td>Sobel</td>
<td>57.14/58.57</td>
<td>57.86/58.57</td>
</tr>
<tr>
<td></td>
<td>Canny</td>
<td>51.43/53.57</td>
<td>51.79/53.57</td>
</tr>
<tr>
<td>head</td>
<td>Gray</td>
<td>56.07/41.43</td>
<td>62.86/46.43</td>
</tr>
<tr>
<td></td>
<td>LBP</td>
<td>50.00/47.86</td>
<td>50.00/47.14*</td>
</tr>
<tr>
<td></td>
<td>Sobel</td>
<td>55.36/51.43</td>
<td>56.07/54.29</td>
</tr>
<tr>
<td></td>
<td>Canny</td>
<td>50.36/48.57</td>
<td>50.36/55.00</td>
</tr>
<tr>
<td>Holistic Action</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>face</td>
<td>Gray</td>
<td>61.43</td>
<td>60.71*</td>
</tr>
<tr>
<td></td>
<td>LBP</td>
<td>49.29</td>
<td>45.71*</td>
</tr>
<tr>
<td></td>
<td>Sobel</td>
<td>40.71</td>
<td>44.29</td>
</tr>
<tr>
<td></td>
<td>Canny</td>
<td>37.86</td>
<td>39.29</td>
</tr>
<tr>
<td>head</td>
<td>Gray</td>
<td>58.57</td>
<td>55.71*</td>
</tr>
<tr>
<td></td>
<td>LBP</td>
<td>50.71</td>
<td>46.43*</td>
</tr>
<tr>
<td></td>
<td>Sobel</td>
<td>44.29</td>
<td>42.86*</td>
</tr>
<tr>
<td></td>
<td>Canny</td>
<td>38.57</td>
<td>32.86*</td>
</tr>
</tbody>
</table>

Table 5.4: The mean correct classification rates of component approaches using a 10 fold cross validation technique. The best mean correct classification rate of each approach is in bold. Lower rates returned by my proposed method are marked by the (*) symbol.

<table>
<thead>
<tr>
<th>vector length</th>
<th>include contempt</th>
<th>exclude contempt</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component – reported % are split into include neutral/exclude neutral</td>
<td></td>
<td></td>
</tr>
<tr>
<td>120</td>
<td>76.79/74.29</td>
<td>76.07*/80.00</td>
</tr>
<tr>
<td>Component Action</td>
<td></td>
<td></td>
</tr>
<tr>
<td>44</td>
<td>94.29</td>
<td>98.57*</td>
</tr>
<tr>
<td>36</td>
<td>90.00</td>
<td>83.57*</td>
</tr>
</tbody>
</table>
upper and lower threshold of 128:128. When the thresholds are the same like this, specular edges appear. Because of specular edges I moved to a 1:2 ratio of lower 85 and upper 170. After trials of other ratios I perceived that the 1:2 ratio was good for both the Sobel and Canny edge operators. With regard to the Sobel edge operator, I have set blur to 0 and a mid-gain of 5 as a good compromise between high and low attenuation of the edge images.

5.5.2 Component is the better approach

An examination of the holistic and component results Tables (5.3 and 5.4) reveals that the component action approach returned the best result with an accuracy of 98.57%. This was followed by the component approach with 80.83% best accuracy and the holistic approach with a best accuracy of 71.15%. Least accurate was the holistic action approach with a best of 62.50%. The reported percentages are from the proposed pairwise adaptive multiclass SVM, which performed favourably in many of the experiments conducted.

There is still an on-going debate between the merits of the holistic and component approaches [231]. My results indicate that a component approach improves classification of discrete facial expressions over a holistic approach. Could it be the influence of holistic approaches that are dominant in the area of facial recognition? When face recognition research was first introduced, the primary methods relied mostly on feature points and distances between them [189]. This resulted in poor recognition, and researchers began looking into holistic methods that dealt with intensity values such as greyscale [10]. Then facial expression studies emerged at the peak popularity of holistic methods [113] and soon a large set of literature on holistic approaches for facial expression classification followed [60, 231].

The effectiveness of component approaches can be seen in other face studies such as detection [18, 74, 92, 103, 104, 160, 221] and recognition [91]. In particular, studies such as [38]1 show that most expressions can be conveyed from a single facial area. This suggests the redundancy of a holistic approach for use in facial expressions – and perhaps why the holistic results are inferior to those of a component approach.

1used 9 expressions: agree, disagree, disgust, thinking, happy, sad, surprised, confused, clueless
5.5. Results and discussion

5.5.3 Excluding the contemptuous expression and neutral does impact performance

As mentioned in the introduction, the contemptuous expression and neutral have been excluded in most of the classification studies, with the former being covered substantially less. Returning to the summary Table (5.2), the numerical results of the holistic and component approaches in Table 5.3 and 5.4 show that when the contemptuous expression and neutral are excluded, the overall classification performance improves. I examine these two expressions more closely.

The neutral face is difficult to classify. One reason for this is its subtle resemblance to other facial expressions, and this can be linked to the overgeneralisation of an emotion recognition system [188]. To analyse this impact I use a two-class SVM with a Gaussian kernel to compare the neutral face with the rest of the expressions, and the results of this are shown in Table 5.5. From the table, a high rate of 76.15% for the holistic and 88.85% for the component approach suggests that SVMs are capable of distinguishing between a neutral and an expressive face. These results are reflected in the Receiver Operator Characteristic Curves (ROC) of Figure 5.4.

Table 5.5: Neutral vs Expressions. A high rate of 76.15% for holistic and 88.85% for component suggests that SVMs are capable of distinguishing between a neutral and an expressive face. The ‘Area Under the ROC Curve’ (AUC) of Figure 5.4 is shown next to the respective rates column.

<table>
<thead>
<tr>
<th>Preprocessing</th>
<th>Include Contempt</th>
<th>Exclude Contempt</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gray</td>
<td>74.29/68.57</td>
<td>0.79/0.74</td>
</tr>
<tr>
<td>LBP</td>
<td>70.36/68.21</td>
<td>0.73/0.68</td>
</tr>
<tr>
<td>Sobel</td>
<td>70.71/66.43</td>
<td>0.70/0.65</td>
</tr>
<tr>
<td>Canny</td>
<td>66.07/64.29</td>
<td>0.64/0.63</td>
</tr>
</tbody>
</table>

Component

<table>
<thead>
<tr>
<th>Preprocessing</th>
<th>Include Contempt</th>
<th>Exclude Contempt</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gray</td>
<td>74.29/68.57</td>
<td>0.79/0.74</td>
</tr>
<tr>
<td>LBP</td>
<td>70.36/68.21</td>
<td>0.73/0.68</td>
</tr>
<tr>
<td>Sobel</td>
<td>70.71/66.43</td>
<td>0.70/0.65</td>
</tr>
<tr>
<td>Canny</td>
<td>66.07/64.29</td>
<td>0.64/0.63</td>
</tr>
</tbody>
</table>
Figure 5.4: ROC curves reflecting the results of Table 5.5. The component approach (i.e. feature points) is the best performer for distinguishing between a neutral and an expressive face. This is shown by the ‘Inc point’ and ‘Exc point’ curves, where ‘Inc’ and ‘Exc’ is including and excluding the contemptuous expression, respectively.
5.5.4 Why the contemptuous expression is considered universal

Ekman describes the contemptuous expression as a tightening and slight raising of the lip corner, primarily on one side of the face [49]. This visual asymmetry separates it from the rest of the facial expressions. A primary question is ‘What makes the contemptuous expression a minority?’ Critics such as Russell have mentioned that the contemptuous expression is the least well established of the universal expressions [185]. One controversy is the expression being mistaken to be disgusted ([185]) or a variant of disgusted ([190]). Russell questions the validity of Ekman’s forced choice methods, and Ekman later responds to Russell’s work as being a mistaken critique [50].

There are a number of studies that support the universality of the contemptuous expression and its separation from the disgusted expression. For instance, Ekman’s initial study in 1986 reported an average recognition of 75% across 10 cultures for the contemptuous expression [54]. Ekman’s replicated study with Heider in 1988 [49] showed strong agreement, as did supporting studies made by Matsumato in 1992 [146].

In [51] Ekman cited that his concept of the universality of the contemptuous expression was conceived by evidence from literate cultures, as locating an isolated preliterate culture was not feasible. Ekman also mentioned that ‘fearful’ and ‘surprised’ were only distinguishable in literate cultures. In this study I assumed an application for a literate culture, and based on this assumption a fearful expression can be distinguished from a surprised expression and the contemptuous expression is universally recognised.

5.5.5 Proposed SVM mostly outperforms standard SVM

My proposed pairwise adaptive multiclass SVM shows better results than the standard multiclass SVM in many cases. Out of 56 experiments, only 17 returned a slightly lower classification rate – 11 in holistic action, 3 in holistic, 2 in component action and 1 in the component approach (they are marked by the (*) symbol in the holistic and component approach Table 5.3 and 5.4).

5.5.6 Is there an optimal parameter pair?

A question that I consider to be relevant in this study is whether there is an optimal parameter pair in the parameter space for classifying expressions. To address this question, I use Figure 5.5 and 5.6, which plot (respectively) the
5.5. Results and discussion

$C$, $\gamma$ and correct classification rates of all the approaches for the standard and proposed multiclass SVMs. Each approach in the figures is represented by colour coded circles and the largest circles represent the most frequent pair.

Looking at Figure 5.5, the standard multiclass SVM has only 2 repeated $C$ and $\gamma$ values of $\log_2(3,-15)$ and $\log_2(9,-15)$ with classification rates of 38.57% and 61.43% using a holistic action approach. This small amount is not considered an optimal pair. However, the pairwise adaptive multiclass SVM has 54 repeated $C$ and $\gamma$ values of $\log_2(-5,-7)$ with classification rates of 100% (component approach) and 87.5% (holistic approach). In both figures, the component action approach is the best performer as shown at the top layer of the plots.

Figure 5.7 shows a contour plot of the correct classification rates of the grid search in LIBSVM [32]. The contour lines mean that the best parameter pairs are not unique. LIBSVM chooses one arbitrarily.

![Standard multiclass SVM](image)

**Figure 5.5:** This 3D plot displays the correct classification rate of $(C, \gamma)$ pairs for all experiments of the standard multiclass SVM where each approach is shown by colour coded circles.
5.5. Results and discussion

Figure 5.6: In the pairwise adaptive multiclass SVM approach a \((C, \gamma)\) pairing can occur several times in one experiment. The frequency of \((C, \gamma)\) pairs is indicated by the number of increasing circles. The most frequently occurring pair is \(\log_2(-5, -7)\) with classification rates of 100% (component approach) and 87.5% (holistic approach).

5.5.7 Performance dynamics of expression pairs suggest a non-equidistant expression space

Recently the topic of expression space based on discrete facial expressions was investigated [136, 219]. In Table 5.6, I present the performance dynamics of expression pairs where each rate is an average of all the experiments using the pa-SVM method. Examining this Table (5.6), the results of the paired expressions indicate that some pairs of expressions are easier to distinguish than others. This suggests an expression space where expressions are not equidistant. For instance, the fearful and surprised pair in Table 5.6 returned the lowest average rate while the happy and surprised pair gave the best average
The poor performance of the fearful and surprised pair corresponds to the difficulty in distinguishing the two expressions that is apparent in preliterate cultures [51] – suggesting that the expression space of fearful and surprised has a larger overlapped region than that of happy and surprised.

Further, these results agree with studies such as [17] which showed that human participants performed poorly at separating anger and disgust while happy and surprised were easily distinguished. In the same study, they excluded sad and fear as they decided that these expressions were not as distinctive as the other basic expressions. The fear-sad expression pair in Table 5.6 agrees with their decision, as it is ranked third last. Examining the table specifically on the contemptuous expression (because of its visual asymmetry), the table shows that the classifier can easily distinguish it from surprised, but only poorly distinguish it from angry.

The above mentioned studies with human participants cover only a small subset of my results shown in Table 5.6. An extensive literature review shows a lack of human participant research for the remaining expression pairs in Table 5.6.

Table 5.6: Performance dynamics of expression pairs. Each rate is an average of all the experiments using the pa-SVM method for that particular pair of expressions.

<table>
<thead>
<tr>
<th>Pair</th>
<th>Rate</th>
<th>Pair</th>
<th>Rate</th>
<th>Pair</th>
<th>Rate</th>
<th>Pair</th>
<th>Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>hap,sur</td>
<td>96.88</td>
<td>sur,dis</td>
<td>90.98</td>
<td>sad,neu</td>
<td>87.88</td>
<td>con,sad</td>
<td>81.25</td>
</tr>
<tr>
<td>hap,fea</td>
<td>93.75</td>
<td>fea,neu</td>
<td>90.69</td>
<td>ang,fea</td>
<td>87.59</td>
<td>hap,dis</td>
<td>80.45</td>
</tr>
<tr>
<td>sur,ang</td>
<td>93.04</td>
<td>ang,neu</td>
<td>90.25</td>
<td>hap,ang</td>
<td>87.05</td>
<td>hap,con</td>
<td>78.93</td>
</tr>
<tr>
<td>sur,con</td>
<td>92.32</td>
<td>hap,sad</td>
<td>89.82</td>
<td>ang,sad</td>
<td>85.09</td>
<td>con,ang</td>
<td>78.21</td>
</tr>
<tr>
<td>sur,neu</td>
<td>92.13</td>
<td>dis,fea</td>
<td>88.84</td>
<td>dis,sad</td>
<td>84.11</td>
<td>sad,fea</td>
<td>76.69</td>
</tr>
<tr>
<td>hap,neu</td>
<td>91.94</td>
<td>con,fea</td>
<td>88.13</td>
<td>sur,sad</td>
<td>82.86</td>
<td>ang,dis</td>
<td>74.20</td>
</tr>
<tr>
<td>dis,neu</td>
<td>91.06</td>
<td>con,neu</td>
<td>80.06</td>
<td>con,dis</td>
<td>81.61</td>
<td>fea,sur</td>
<td>74.11</td>
</tr>
</tbody>
</table>

5.5.8 Favourable results compared to other studies

To compare with other relevant studies I examined the best results of facial classification studies using the Cohn-Kanade and MMI datasets. For the Cohn-Kanade dataset, the studies that employed six expression classes (the typical six - happy, sad, surprised, fearful, angry and disgusted) achieved 93.85% [236] and 93.8% [131]; those with seven expression classes (the typical six plus
neutral) achieved 93.8% [134] and 93.3% [13]. For the MMI dataset, the studies that employed six expression classes achieved 82.68% [220]; those with seven expression classes (once again, the typical six plus neutral) achieved 86.9% [196] and 93.61% [123].

In comparison, my best results employing six expression classes achieved 80.83% via a component approach, and for seven expression classes the experiments achieved 98.33% via a component action approach. One main difference between my results and the mentioned studies is that I use a 10 fold cross validation algorithm while some use a separate training and test set [131], and others use a different cross validation method ([236] used 2 fold, [13, 123, 134, 220] used leave-one-out). The use of a separate training and test set or a fold lower than 10 will return a higher classification rate.

Another factor that can impact the overall performance is the choice of image resolution; which I discuss next.

5.5.9 Was the resolution of the generated face dataset optimal?

Several studies have experimented with multiple pixel resolutions to determine an optimal balance of image height and width. For instance, [132] compared 64×64, 48×48, 32×32 and 16×16 on the JAFFE dataset, [196] compared 110×150, 55×75, 36×48, 27×37, 18×24 and 14×19 on the JAFFE, Cohn-Kanade and MMI datasets, and [131] compared 288×384, 144×192, 72×96, 36×48 and 18×24 on the Cohn-Kanade dataset. Each showed that the performance decreased when the face data were downsampled – i.e. [132], [196], and [131] returned reduced performance when the pixel resolutions fell below 64×64, 110×150 and 72×96 respectively. We chose 100×100 pixels because it sits between 64×64 and 110×150 and the expressive features were visually clear.

5.5.10 My proposed face model

My final discussion presents how I arrived at my face model and how I validate it. I begin with a literature review of other face models in the field where the brows, eyes and mouth are used the most often. However, some studies include the jawline [17, 175, 220], chin [116], forehead and between the eyes [81], and other studies include the cheeks [116] as well as wrinkles and furrows [235]. From these studies, I ask ‘Which facial features are relevant and which are redundant for describing a facial expression?’ To address this question, I note
that the majority of the mentioned studies attribute their work based on the ‘Facial Action Coding System’ (FACS) by Paul Ekman and W.V. Friesen [48] — where the intent is to capture facial movements, which are categorised into a set of descriptors known as action units.

Table 5.7: Subset of action units (AU) relevant to facial movements [47]. Action units such as Jaw thrust (AU 29) requires depth measurements, which is beyond the 2D capability of my face model.

<table>
<thead>
<tr>
<th>AU</th>
<th>Description</th>
<th>AU</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Inner brow raiser</td>
<td>2</td>
<td>Outer brow raiser</td>
</tr>
<tr>
<td>4</td>
<td>Brow lowerer</td>
<td>5</td>
<td>Upper (eye)lid raiser</td>
</tr>
<tr>
<td>6</td>
<td>Cheek raiser</td>
<td>7</td>
<td>(eye)lid tightener</td>
</tr>
<tr>
<td>9</td>
<td>Nose wrinkler</td>
<td>10</td>
<td>Upper lip raiser</td>
</tr>
<tr>
<td>11</td>
<td>Nasolabial deepener</td>
<td>12</td>
<td>Lip corner puller</td>
</tr>
<tr>
<td>13</td>
<td>Sharp lip puller</td>
<td>14</td>
<td>Dimpler</td>
</tr>
<tr>
<td>15</td>
<td>Lip corner depressor</td>
<td>16</td>
<td>Lower lip depressor</td>
</tr>
<tr>
<td>17</td>
<td>Chin raiser</td>
<td>18</td>
<td>Lip puckerer</td>
</tr>
<tr>
<td>20</td>
<td>Lip stretcher</td>
<td>21</td>
<td>Neck tightener</td>
</tr>
<tr>
<td>22</td>
<td>Lip funneler</td>
<td>23</td>
<td>Lip tightener</td>
</tr>
<tr>
<td>24</td>
<td>Lip presser</td>
<td>25</td>
<td>Lips part</td>
</tr>
<tr>
<td>26</td>
<td>Jaw drop</td>
<td>27</td>
<td>Mouth stretch</td>
</tr>
<tr>
<td>28</td>
<td>Lip suck</td>
<td>29</td>
<td>Jaw thrust</td>
</tr>
<tr>
<td>31</td>
<td>Jaw clencher</td>
<td>38</td>
<td>Nostril dilator</td>
</tr>
<tr>
<td>39</td>
<td>Nostril compressor</td>
<td>43</td>
<td>Eyes closed</td>
</tr>
<tr>
<td>45</td>
<td>Blink</td>
<td>46</td>
<td>Wink</td>
</tr>
</tbody>
</table>

In Table 5.7, [47] gives a subset of action units relevant to facial movements. From this, it seems logical that using features such as the cheek would be used for detecting the raising cheek action unit (AU 6). However, the cheek is visually a void area of the face, which can be very difficult to detect by machine. Another example is the use of features such as the wrinkles and furrows around the nose to detect the wrinkling of the nose action unit (AU 9); to which I question, How can we separate transient wrinkles from permanent wrinkles (i.e. ageing)? Both examples pose a number of challenges for a machine to detect successfully.

From these examples, I picked the feature points for my face model based on two principles. The first principle is that the facial features where the
points would be located should have sufficient information for detection, and the second is that these points can be used to describe or deduce an action unit. Returning to Figure 5.2 of my face model, I have decided the feature points to be positioned around the brows, eyes and mouth because the edges around these areas are distinct. The feature points of my face model can be used to deduce an action unit. For instance, I deduce a cheek raise (AU 6) by the raising of the mouth corners and raising of the lower eyelids. I deduce a wrinkling of the nose (AU 9) by the raising of the upper lip, lower eyelids and lowering of the inner brows. The remaining action units of FACS can be deduced by this inference logic; except for jaw thrust (AU 29) as it requires depth measurements.

Next, to validate my proposed face model I use photos of Eve from the Paul Ekman Group LLC [167]. Eve was used extensively in Paul Ekman’s research to display a range of subtle and discrete facial expressions. 41 face images of Eve were obtained, consisting of 4 happy, 10 sad, 6 surprised, 7 fearful, 7 angry, 4 disgusted, 2 contemptuous and 1 neutral. Figure 5.8 is a neutral face of Eve (passed through a Sobel edge operator).

When I ran the facial actions of my face model using the actions as described in Table 5.1 I obtained some face images with duplicated facial actions as shown by Figures 5.9, 5.10 and 5.11 (each passed through a Sobel edge operator). Figure 5.9 shows two angry expressions with the one on the right having a higher intensity. In Figure 5.10 the left is described as a determined anger and the right as a controlled sadness, and in Figure 5.11 the left is described as surprised; however, if this expression is held for a longer time it could be fear or controlled fear on the right. These duplicates do affect the capability of my model in differentiating the seven discrete universal facial expressions. This is because duplicate 1 and 2 can be resolved using intensity analysis, while duplicate 3 could be resolved using time series analysis – which can also play a part for resolving duplicate 2. These images with duplicated facial actions were retained in the study.

5.5.11 Future direction: intensity analysis and identification

As Eve was the only image set that had all the expressions, I overlaid all the feature points of each expression to create what I called a motion model as shown in Figure 5.12. Figure 5.13 displays the intensity graphs of Eve’s motion model with respect to each of her expressions. The intensity graphs are normalised by the maximum displacement range of each feature point.
5.6 Summary

From these graphs we can deduce the feature points that are significant to an expression. For instance the expansion and contraction of the mouth for happy and surprised, the raising of the eyelid for surprised and fearful, the lowering of the inner brows for angry and disgusted, and the asymmetric gesture of contemptuous expression. The motion model is a promising emotion signature of an individual – and if it could be shown to be unique amongst individuals then it is applicable for identification.

5.6 Summary

Current approaches to facial expression classification differ by the number of discrete expressions, choice of dataset and the number of different preprocessing techniques. This makes comparison of results between studies difficult. For these reasons, this study was initiated to find the best techniques to provide a comparative bridge (and suggest a standard) so that results can be compared. In the literature review I found that there was a lack of certified datasets publicly available and many did not cover the contemptuous expression (as well as the neutral face). My examination of the literature shows that the reported classification rates will be artificially higher for studies that exclude these expressions as well as for those that use separate training and testing sets or a cross validation fold lower than 10. Further, the literature review reveals the influence of holistic approaches for facial recognition over component approaches for facial expression classification.

A face model was proposed for aligning the faces in the dataset. The roles of each of the feature points in the face model were justified – this was not clear in the literature. For the classification task, my proposed pairwise adaptive multiclass SVM performed better than the standard multiclass SVM in many of the experiments conducted except for the holistic action approach, which involved the difference of face images. And finally, when I analysed the performance of expression pairs, the finding shows that the performance dynamics of paired expressions suggest a non-equidistant expression space – as I observed specific pairs of expressions performing better than others – and these observations correspond to psychological studies using real human subjects. The overall significance of this study is that it provides a thorough understanding of the choices that impact a classifier’s performance on the seven universal facial expressions and the performance dynamics between expression pairs. I foresee my future work to extend my face model into the areas of intensity analysis and identification.
5.6. Summary

Concluding the chapter summary, I have shown the possible variations that exist in classifying facial expressions. I considered Ekman’s Universal Facial Expressions, which consist of seven discrete expressions (the typical six six expression used in the literature – happy, sad, surprised, fearful, angry, disgusted – plus the contemptuous expression) where the expression space is unequally spaced as shown by my pairwise adaptive SVM results. The following chapters present face perception techniques from both a holistic and a component approach to realise my thesis question, ‘Can a machine describe the emotional response to a scene using perceived faces and facial expressions?’
Figure 5.7: Example grid search from the component action approach to show that $C$ and $\gamma$ are not unique to the classification rate.
5.6. Summary

Figure 5.8: Eve - neutral face.

Figure 5.9: Duplicate facial actions 1: Right is described as an intense expression of left.

Figure 5.10: Duplicate facial actions 2: Left is described as a determined anger while right is controlled sadness.
Figure 5.11: Duplicate facial actions 3: Left is described as surprised however it is described as fear on the right if the surprised expression is held too long.

Figure 5.12: A Motion Model of Eve, which has the potential to be an emotion signature to identify Eve.
Figure 5.13: Intensity graphs of Eve, which show the relevant feature points significant to an expression.
Chapter 6

Holistic Face Perception

The contents of this chapter have been published in [25, 26, 99, 223].

Contents

6.1 Introduction ........................................... 99
6.2 System overview ...................................... 102
6.3 Dataset and image preprocessing ...................... 104
  6.3.1 Dataset preprocessing ................................. 104
  6.3.2 Image preprocessing .................................. 106
6.4 Classification methods ................................ 106
  6.4.1 Support Vector Machines (SVM) and kernels ...... 106
  6.4.2 The SVM and kernel parameters to optimise ...... 109
  6.4.3 pα-SVM framework ................................. 109
  6.4.4 SVM and kernel search parameters .................. 109
6.5 Null test for discarding face detectors that see faces
  in nothingness ......................................... 109
6.6 Window search and a bounding box for estimating the
  true positives rate .................................... 110
6.7 Results and discussion ............................... 110
  6.7.1 Edge operators over histogram equalised greyscale
        produces artificial edges ............................ 112
  6.7.2 Cross validation of face detection shows non-equalised
        greyscale as the best and most stable performer .... 113
  6.7.3 Null test shows Canny detects faces in nothingness –
        which is undesirable ................................ 113
  6.7.4 Window search shows Sobel detected faces to be robust 114
  6.7.5 Cross validation of expression classification shows
        equalised greyscale as the best performer .......... 116
  6.7.6 Sobel expression classifiers over Sobel detected faces
        shows best performance ............................. 116
‘Can a machine describe the emotional response to a scene using perceived faces and facial expressions?’ At this point in my thesis I am ready to realise my perceptive machine. The previous chapters have provided the foundation necessary to establish the relevance of my thesis question by presenting literature from Cognitive Science and Psychology (Chapters 1 and 2), then a background theory of Support Vector Machines (Chapter 3) – which is my main machine learning method. Through my experiments of comparing holistic and component approaches for classifying discrete facial expressions (Chapter 5) I have contributed a refinement to SVM, known as a ‘Pairwise Adaptive SVM’ ($pa$-$SVM$). Because I have found this method to be generalisable and have begun using it in other areas of my thesis, it seems logical that $pa$-$SVM$ is presented prior other contributions (Chapter 4). What remains is integrating facial expression classification with face detection, which is the purpose of my final two chapters.

In this chapter, I present a study with the aim to devise a machine that can describe a scene using holistic faces and facial expressions. The significance of this study is to simulate the pareidolia capability of humans to produce an emotional response to a scene using the facial expressions of perceived faces. This novel process uses a holistic face detector and a facial expression classifier. The $\nu$ and SVDD One-Class Support Vector Machines (SVM) were evaluated for creating a holistic face detector, which looks for faces that can vary from natural faces to minimal face-like patterns. A Pairwise Adaptive $C$ and $\nu$-SVM ($pa$-$SVM$) were evaluated for creating the facial expression classifier. In both
scenarios, a dataset of human faces and facial expressions was used to produce a number of preprocessed images (greyscale, histogram equalised greyscale, and their respective Sobel and Canny edges) at a number of resolutions for analysis. A Gaussian and a degree 2 polynomial kernel were used with the SVM methods and the results were obtained using a 10 fold cross validation technique. A concern with the face detectors is verifying that they can look for minimal face-like patterns empirically. To address this concern, I created cartoon faces of the human face dataset and degraded these cartoon faces to produce an array of minimal face-like patterns. I then evaluated the face detectors and facial expression classifiers with the best model parameters on these cartoon faces. The outcome is a holistic system with the potential to describe a scene by producing an array of emotion scores corresponding to the seven Universal Facial Expressions of Emotion.

6.1 Introduction

Scene perception is defined as how we see a scene and perceive its meaning. From the literature this meaning is known as the gist and it refers mostly to our ability to categorise a scene. For instance, vague and hybrid images of a hallway, a city and a street are used to measure the speed that we can determine its category [158]. My interest with scene perception is the idea of using faces and facial expressions to produce an emotional response. So far, facial expression and scene perception literature analyses facial expressions in the context of emotional scenes [105, 122, 178, 179]. In a previous work we hypothesised that the expressions of faces perceived in house designs through pareidolia can produce an emotional response in the observer [25, 26]. In this study, I refine my software system for pareidolia and extend the work into scene perception where a broader application can be evaluated (i.e. faces in cars, clouds, trees, etc).

I begin by presenting the relevant literature from Cognitive Science and Psychology to show the brain regions that are dedicated to processing emotional scenes and faces – this dedicated neural basis for scenes and faces is an early indication of why faces and facial expressions are the appropriate features for describing a scene. I then present supporting studies to show the overlapping brain regions that are active in response to emotional scenes and faces. Next, I describe how my hypothesis relates to pareidolia and the subconscious, which then leads to my proposal.

Scene perception is a rapid visual process [78] where the initial phase of
rapid categorisation is suggested to be by global properties of structure and affordance [77]. This suggests a holistic approach. Using functional magnetic resonance imaging (fMRI) the brain region known as the parahippocampal gyrus has shown to be active when a scene is processed [55], and sensitive to the holistic properties of a scene [56]. When a scene is emotional, the brain region known as the amygdala is activated [186]. This is not surprising as the amygdala is known to be responsible for processing emotions, and responds to visual stimuli [149, 195].

Similar to scene perception, recognising faces and emotions from facial expressions is a rapid visual process [83]. Using fMRI, the brain region known as the fusiform gyrus has shown to be active for faces [115], and is considered to process faces holistically [234]. Likewise, when a perceived face is emotional, the amygdala has shown to be active, and this rapid activation has been studied in newborns [112] and adults [2, 191, 199]. A supporting study, which analysed fMRI results of brain regions on emotional scenes and emotional faces, showed that the activation of the amygdala has the greatest overlap [187], and the activation is stronger for facial expressions than for scenes [86, 195]. An illustration of where the fusiform gyrus, amygdala and parahippocampal gyrus are located in the brain is shown in Figure 2.1 of Chapter 2.

Returning to the study of newborns recognising faces and facial expressions [112], this implies that our brain may be hard-wired to look for faces from birth, and the use of schematic face-like patterns in the study suggests that newborns are capable of perceiving minimal face-like patterns. The idea of seeing faces from minimal face-like patterns is known as pareidolia. It is defined as the ability to perceive significance from random and vague stimuli. It is an innate ability that is commonly practised – especially for faces [222]. For instance, seeing the face of Jesus Christ on a potato chip [222] and on the turf of Crystal Palace FC at London [65]; Mother Teresa on a cinnamon bun [65]; Virgin Mary on a grilled cheese sandwich [145, 222]; Satan in the smoke of 9/11 [145]; the face on Mars [145]; and a devil in the Queen’s hair of a 1954 Canadian banknote [145]. There are indications that examples of pareidolia of faces from minimal face-like patterns activate the same brain regions as real faces [84, 148].

Supporting the pareidolia effect, a study on rapid scene categorisation suggests that the presence of the target category can be perceived outside the focus of spatial attention, even when the scenes were task-irrelevant [168]. And, a recent study using noisy images and fMRI suggests that faces can
be perceived in minimal face-like patterns even when these features are not centred in our visual focus [176].

The existence of brain regions that are particular for scenes and faces suggests that priority is given to processing these features. The pathways that link these brain regions and the speed at which face-like information is processed suggest a subconscious and a conscious process [41, 210], and where a stimuli is considered emotional, the amygdala may be involved in modulating the threshold of subconsciousness and consciousness [149]. ‘How does the subconscious relate to my hypothesis?’ I hypothesise that when we perceive a scene, we may lack the ability to fully describe our emotional response to it as the answers in our conscious mind are limited by our ability to question our subconscious. The rapid rate at which the subconscious is processed suggests that it is rich in information regarding the emotion of scene perception, and the ability for a machine to relay this information allows our emotional response to a scene to become transparent. The supporting studies on pareidolia for rapid face detection and scene perception mentioned previously [168, 176] suggest a complex interplay between the conscious and the subconscious.

Given that a neural basis is shown to exist for scenes and faces, it seems faces are the appropriate features to describe a scene. The idea of using emotion as a strong description of a scene is supported by studies that showed that the amygdala region shares the greatest overlap in activation when scenes and faces are processed. The use of a pareidolia effect of abstract faces and facial expressions to produce an emotion was first proposed in a previous work where the focus was on house designs [25, 26]. In this chapter, I extend the work into scene perception where a broad application can be evaluated.

Several machine learning methods are available to realise my perceptive machine. However, from the suggestion of our brain being hard-wired to look for faces from birth I hypothesised that the encoded knowledge would primarily be influenced by face data (of the positive class) rather than non-face data (of the negative class). My hypothesis directs our attention to one-class classification methods. Two comparative studies on general tasks have shown that one-class classification using SVMs has outperformed its competitors [73, 239]. In one study where the one-class SVM did not achieve the overall best performance, it did show more robustness than other methods [174]. Narrowing my review for face detection, there is literature of one-class SVMs being used [24, 111]; however, (with the exception of the previous work [25, 26]) those studies are specific to finding human faces.

In this study I investigate a number of image preprocessing techniques,
resolutions and different configurations of classifiers to understand which parameters and classifiers are best to detect a range of faces from natural faces to minimal face-like patterns. I propose a two stage solution. The first uses a 10 fold cross validation to isolate the parameters and classifiers that achieve the best one-class classification accuracy. Then I produce a set of cartoon faces using a human face training dataset, and degrade these cartoon faces across several resolutions to produce an array of minimal face-like patterns. A window search is then carried out where a degraded cartoon face is placed in the center of a white image to isolate those parameters and classifiers that produce the best detection rate. The purpose of the degraded cartoon faces and window search is to measure the robustness of the best parameters and classifiers empirically.

Following a previous publication [25, 26], I use SVMs for creating and evaluating the expression classifiers. A Pairwise Adaptive SVM (pa-SVM) is used to evaluate a number of resolutions and image preprocessing techniques, and is compared to standard multiclass SVMs [97]. For face detection, a one-class SVDD and \( \nu \) SVM is evaluated. A Gaussian and a degree 2 polynomial kernel are used for the one-class and multiclass SVMs.

A system overview is presented in the next section, followed by a description of the dataset and image preprocessing used in Section 6.3. Section 6.4 presents the classification method, and a null test is presented in Section 6.5 to discard from further consideration the face detectors that see faces in nothingness. Then in Section 6.6 a window search is presented to evaluate the robustness of the face detectors – detected faces are then used to measure the robustness of the expression classifiers. Lastly, the experimental results and discussion are presented in Section 6.7 and concluding remarks in Section 6.8.

6.2 System overview

My system consists of several steps as illustrated in Figure 6.1. The first step was to align the face dataset using manually inserted feature points. Each face image is preprocessed into greyscale, histogram equalised greyscale, and their respective Sobel and Canny edges. Once the datasets are normalised between the values of -1 and +1 they are split into two processes. One process uses the dataset to create face detectors for detecting a range of faces, from natural faces to minimal face-like patterns. The other process uses the dataset to create expression classifiers, which are later evaluated on the detected faces.

For both processes the datasets are partitioned into classes and divided into
Figure 6.1: System overview showing the flow of the stages.
training and validation sets using a $k$-fold cross validation algorithm (where $k$ is 10) – seven classes for the expression classifiers and 1 class for the face detectors. This is then followed by obtaining the SVM and kernel grid parameters for the experiments.

Once the best parameters are determined for the face detectors, I discard from further consideration the face detectors that see faces in nothingness. Then I perform a window search on a set of test images where a degraded cartoon face is placed in the center of a white image. The cartoon faces are generated using the manually inserted feature points, and are degraded across several resolutions to produce an array of minimal face-like patterns. The detected faces from the window search are evaluated with the best expression classifiers to determine which one is optimal. Details of further steps are explained in subsequent sections.

6.3 Dataset and image preprocessing

In this section I describe the face dataset, how the images were aligned, how the degraded cartoon faces were generated and the image preprocessing techniques that were used.

6.3.1 Dataset preprocessing

The face dataset was generated using the JACFEE and JACNEUF datasets, which are certified by the Paul Ekman Group LLC [167]. I specifically chose these datasets because they have discrete expression classes (the seven universal facial expressions of emotion as well as neutral faces). The generated dataset consists of 280 coloured face images, including 140 neutral faces and 20 images of seven different facial expressions: happy, sad, surprised, fearful, disgusted, angry and contemptuous. I consider the neutral face separate from the seven universal facial expressions of emotion, and because of this, it was not used for expression classification; however, I used it for face detection.

A total of 60 feature points were manually inserted for each face image along the brows, eyes and mouth – as shown in Figure 6.2. I use 2 feature points of the inner eye corners (a.k.a. lacrimal caruncle) for aligning the faces in the dataset. Once aligned the images are cropped and resized across 10 resolutions with a height and width of 15, 19, 24, 30, 38, 47, 59, 73, 92 and 114 pixels – roughly a 1.25 step up using 24 as the base.
My face model defined by the manually inserted feature points was proposed in previous publications [96, 98], which showed a component approach performing better than a holistic approach for classifying discrete facial expressions. As mentioned, the feature points of the face model are used for aligning the faces in the dataset and for creating cartoon faces, which are then degraded across a number of resolutions – i.e. I take an image of \((m, m)\) pixels, resize it to a lower \((n, n)\) resolution, and then resize it back to the original resolution (this normalisation step is necessary to set a standard size).

Specifically, 16 degraded cartoon faces of seven facial expressions were used for the array of minimal face-like patterns. At one end I start with a clear and sharp cartoon face, which is generated by averaging the \(xy\) positions of the manually inserted feature points of the 20 faces per expression, and drawing lines between these averaged \(xy\) positions. The resolution of the clear and sharp cartoon face is set at \(114 \times 114\) pixels. The array of degraded cartoon faces is produced by resizing each face to a height and width of 92, 73, 59, 47, 38, 30, 24, 19, 15, 10, 8, 6, 5, 4 and 3 pixels, then back to \(114 \times 114\) – and then placing each in the center of a white image for assessing the best parameters and best classifiers empirically.

![Figure 6.2: Face model of [96, 98]. The feature points shown in red and numbered ‘1’ are used for aligning the faces in the training dataset. The complete face model is shown in Figure 5.2 of Chapter 5.](image-url)
6.3.2 Image preprocessing

I describe a number of preprocessing techniques used: the greyscale, histogram equalised greyscale, and their respective Sobel and Canny edges.

The greyscale uses an ITU-R 601-2 luma transform on RGB (red green blue) colours [1]:

\[
greyscale = \text{red} \times 0.299 + \text{green} \times 0.587 + \text{blue} \times 0.114
\]

The purpose of histogram equalisation is to stretch the intensity values of the greyscale to the full range of available intensities. It is a popular technique for improving contrast in an image [20]. The edge operators are then performed on the non-equalised and equalised greyscale.

The Sobel and Canny edge operators have several parameters that impact the final edge image. Selection of the Sobel and Canny parameters was by visual evaluation for ideal face edges. The chosen parameters are as follows: aperture size of 3 and blur of 1 for Sobel; aperture size of 3, lower threshold of 127 and upper threshold of 255 for Canny.

Figure 6.3 shows sample images of the seven universal face expressions of emotion in greyscale, Sobel and Canny. Corresponding cartoon images are also shown alongside, and Figure 6.4 shows a subset of the seven cartoon faces used for the window search. We can see that at low resolutions the facial expressions become difficult to distinguish.

6.4 Classification methods

Following the conclusions of Sections 3.1.3 and 3.1.2 I have chosen SVM as my preferred learning method for creating the face detectors and expression classifiers. Specifically, I have chosen to evaluate the $C$-SVM and $\nu$-SVM for my study.

6.4.1 Support Vector Machines (SVM) and kernels

From the literature review of kernels with SVMs in Sections 3.2.3 and 3.2.2 I have chosen to conduct my experiments using a Gaussian and a degree 2 polynomial kernel.
6.4. Classification methods

Figure 6.3: Sample images of the seven universal face expressions of emotion in greyscale, Sobel and Canny. Corresponding cartoon images are also shown alongside. The face dataset was generated from the JACFEE and JACNEUF datasets [167].
Figure 6.4: A subset of the seven cartoon faces used for the window search. They are produced by the average of the feature points corresponding to an expression, and resized to produce the minimal face-like patterns. We can see that at low resolutions the facial expressions become difficult to distinguish.
6.5. Null test for discarding face detectors that see faces in nothingness

6.4.2 The SVM and kernel parameters to optimise

From the theories of SVM and kernels in Chapter 3 and specifically Sections 3.7-3.8, my choice of the \( C \)-SVM and \( \nu \)-SVM with a Gaussian and a degree 2 polynomial kernel has some parameters to tune. They are \((C, \gamma)\) for the \( C \)-SVM and \((\nu, \gamma)\) for the \( \nu \)-SVM – where \( C \) is a parameter for \( C \)-SVM, \( \nu \) is a parameter for \( \nu \)-SVM, and \( \gamma \) is a parameter for the Gaussian and polynomial kernel.

6.4.3 \( pa \)-SVM framework

The \( pa \)-SVM framework of Chapter 4 is used for my expression classifiers. Compared to the holistic facial experiments conducted in the previous chapter (5), I have expanded to a number of resolutions. One reason is to understand the impact of the classifier’s performance across resolutions and whether my results agree with similar studies exploring this effect.

6.4.4 SVM and kernel search parameters

I use the same \((C, \gamma)\) search parameters as [97] (Chapter 4) in my multiclass SVM experiments:

\[
C \in \{2^{-5}, 2^{-4}, ..., 2^{14}\}
\]

and

\[
\gamma \in \{2^{-17}, 2^{-16}, ..., 2^2\}
\]

And likewise, for the sequences of \((\nu, \gamma)\) I grow \( \nu \) by a linear step of 0.05 up to 1.0, while \( \gamma \) remains the same. i.e.

\[
\nu \in \{0.05, 0.10, ..., 0.95, 1.00\}
\]

For my one-class SVM experiments that use only \( \nu \) and \( \gamma \), I use the same \((\nu, \gamma)\) values as for the multiclass SVM experiments.

6.5 Null test for discarding face detectors that see faces in nothingness

A concern with one-class classification arises when the boundary decision encompasses a known negative. In my case, a known negative is the detection of
6.6 Window search and a bounding box for estimating the true positives rate

Two instances of this are black or white blank images, and any face detector that sees a positive face in these instances is considered undesirable and should be discarded from further processing. To discover which face detectors to discard from further processing I ran the face detectors over a series of black and white blank images. The face detectors that pass this test are used in the window search described next.

6.6 Window search and a bounding box for estimating the true positives rate

A concern with one-class face detectors is verifying that they can look for minimal face-like patterns empirically. To address this concern, a window search is performed on the 16 degraded cartoon faces with the seven expressions, where each degraded cartoon face is placed at the center of a $342 \times 342$ test image – which is 3 times the size of the centred image. An evaluation procedure is undertaken by sliding a $114 \times 114$ sized window across the test image. At each step the window image is resized and preprocessed to reflect the dimension and the image preprocessing technique used for training the candidate face detector. Not all possible positions in the test image need to be searched – this is because the null test would have thrown out those face detectors that see faces in nothingness.

To measure the true positives rate I devised a number of bounding boxes around the center of the test image. This can best be explained using Figure 6.5. If a face is detected with its center within the red box it is counted as a true positive; if not, it is a false positive. I define the red box as a 25% bound from the center of the image – it is a good balance for the true positives rate after a pilot study comparing a 50% and 10% bound (see Section 6.7.8). The 100% bound is the dark green box, which considers all candidate face positive. The light coloured boxes represent the outer edges of faces detected on the boundary of the dark coloured boxes. In the same figure, a 50% bound is shown by a dark blue box.

6.7 Results and discussion

Because this investigation covers a number of image preprocessing techniques, resolutions and different configurations of classifiers, the interpretation of the results can be complex. However, as we progress through this section we do
Figure 6.5: Bounding boxes for estimating the true positives rate of a window search. To measure the true positives rate I devised a number of bounding boxes around the center of the test image. If a face is detected with its center within the red box it is counted as a true positive; if not, it is a false positive. I define the red box as a 25% bound from the center of the image. The 100% bound is the dark green box, which considers all candidate faces positive. The light coloured boxes represent the outer edges of faces detected on the boundary of the dark coloured boxes. A 50% bound defined by a dark blue box is also shown.
6.7. Results and discussion

arrive with a set of best parameters and best classifiers, which have shown to be robust and have passed my empirical tests of degraded cartoon faces. The use of degraded cartoon faces was one of many possible options. However, in using them I am able to show empirically that the face detectors are able to perceive faces and minimal face-like patterns that are different to their training data. These differences (notably sharp lines to blurred tones against a white backdrop) mean that particular image preprocessing techniques used for training the candidate face detector would produce better results.

Later in this section, I present a criticism of the edge operators and whether my chosen 25% bounding box for reporting the true positives rate is appropriate. Additionally, I discuss whether the decision values reflect confidence of faces detected, and whether increasing the resolution of classifiers increases performance. The final section shows how the collection of optimal one-class face detectors can be used to detect and categorise different types of faces. Essentially, I have contributed a framework where other image preprocessing techniques can be tested, along with a different set of testing images using a window search to show empirically the robustness of face detectors and expression classifiers.

My presentation of results and discussion follows the structure of my system overview diagram in Figure 6.1. I begin by presenting the results and provide appropriate discussions with regard to face detection, followed by facial expression classification, and I then evaluate their robustness when both are integrated. However, the foremost issue to address is the artificial edges, which are produced when the edge operators operate over histogram equalised greyscale.

6.7.1 Edge operators over histogram equalised greyscale produces artificial edges

As described in Section 6.3, the purpose of histogram equalisation is to increase the contrast of an image by stretching the intensity values of the greyscale to the full range of available intensities. By stretching the intensity values artificial edges appear, which is an undesirable effect. An illustration of these artificial edges can be seen in Figure 6.6 when comparing the non-equalised version of the first row with the equalised version of the second row. The edge operator parameter values remained the same in both cases.
6.7. Results and discussion

Figure 6.6: The artificial edges can be seen in the second row where the edge operators are performed over a histogram equalised greyscale. The edge operator parameter values remained the same in both cases.

6.7.2 Cross validation of face detection shows non-equalised greyscale as the best and most stable performer

The one-class cross validation results of Figure 6.7 show non-equalised greyscale as the best and most stable performer, followed by Sobel and then equalised or non-equalised Canny. To clarify, when I say equalised Canny or Sobel I mean the edge operators over histogram equalised greyscale, while non-equalised Canny or Sobel means the edge operators over non-equalised greyscale.

The accuracy rates may be high in both graphs of Figure 6.7, which is usually the case. However, these accuracies do not reflect robustness. This is the reason for my window search.

6.7.3 Null test shows Canny detects faces in nothingness – which is undesirable

Before I present the results of my window search, I present the results of my null test, which uses all the best face detectors from the one-class cross
6.7. Results and discussion

Figure 6.7: The face detector results on the left side show that non-equalised greyscale is the best and most stable performer. On the right side Sobel is performing best.

validation method in the previous section and tests them on black and white blank images. The graphs in Figure 6.8 show that all SVM types and kernels for Canny do detect faces in nothingness. To a lesser extent, a number of non-equalised greyscale classifiers at low resolutions and a number of non-equalised Sobel classifiers at high resolutions are affected – both these classifiers use a \( \nu \)-SVM with the degree 2 polynomial kernel.

6.7.4 Window search shows Sobel detected faces to be robust

As described in Section 6.6, the purpose of a window search is to assess the robustness of the one-class face detectors. The graphs in Figure 6.9 show only Sobel (green) to be robust. To determine which particular classifier resolution was robust I examined 2 scenarios. The first was the average detection rate, followed by the number of degraded cartoon faces that were successfully detected by the classifiers. Examining the two scenarios, I isolated for non-equalised Sobel the \( 30 \times 30 \) and \( 38 \times 38 \) classifier as the best; for equalised Sobel the \( 114 \times 114 \) classifier was identified as the best.

The poor performance of greyscale does not necessarily mean that it is a bad choice for a perceptive machine. Firstly, it passed the null test, and secondly (as I have mentioned) the cartoon faces are set against a white back-
6.7. Results and discussion

Figure 6.8: Null test shows Canny sees faces from nothingness – which is undesirable, and is discarded from further processing. The graphs show that all SVM types and kernels are affected for Canny. To a lesser extent, a number of non-equalised greyscale classifiers at low resolutions and a number of non-equalised Sobel classifiers at high resolutions are affected – both of these classifiers use a ν-SVM with the degree 2 polynomial kernel. In these plots the y axis is simply the sorted name of the classifier.
drop where there are no tonal shades between the eyes and mouth, which is evident in the greyscale training data; the white backdrop is also not part of the greyscale training data. Once the tonal shades are made less relevant using edge operators such as Sobel, good performance is seen as expected. In a later section (6.7.11) I show further uses of greyscale along with Sobel for detecting other face types.

In the next section I present the cross validation results of the expression classifiers. Then I run the best expression classifiers over the Sobel detected faces of my window search in Section 6.7.6 to show which expression classifiers are robust.

6.7.5 Cross validation of expression classification shows equalised greyscale as the best performer

The multiclass cross validation results of Figure 6.10 show equalised greyscale as the best performer, followed by Sobel then Canny. The plotted rates are based on my new \textit{pa}-SVM method as they are higher than the standard multiclass SVM [97]. The clear gap between the greyscale and Sobel rates in the non-equalised face detectors of Figure 6.7 in Section 6.7.2 is now reflected here in the equalised results of the expression classifiers.

Similar to the point expressed in Section 6.7.2, the accuracy rates may be high in both graphs of Figure 6.10; however, these accuracies do not reflect robustness. To show robustness I run the best expression classifiers over the Sobel detected faces from my window search, which is described below.

6.7.6 Sobel expression classifiers over Sobel detected faces shows best performance

In this section I show two sets of expression classifier results over the Sobel detected faces from my window search. The first set in Figure 6.11 iterates over the Sobel detected faces from my window search – the rate displayed is a count of the number of candidate faces within the 25\% bound where the classifier chooses the correct expression. The second set in Figure 6.11 serves as a relative comparison to the ground truth of how well the expression classifiers could perform if they classified over the exact face location of the Sobel detected faces (i.e. the face located on the exact centre of the test image).

An examination of these graphs shows that Sobel expression classifiers return the best performance in my test configuration. In particular, the graphs
6.7. Results and discussion

Figure 6.9: Window search shows only Sobel to be robust. Using a cut off average detection rate of 80% for the top graphs I show that for non-equalised Sobel detected faces there are two best Sobel classifier resolutions: 30 × 30 and 38 × 38; for equalised Sobel detected faces there is only one Sobel classifier resolution that seems best: 114 × 114 – this is depicted in the graph below, which shows the classifier detecting most of the degraded cartoon faces (identified by the vertical scale). The average detection rate in the top graphs is averaged over the rates from the number of degraded cartoon faces and the seven expressions. For the bottom graphs, a more saturated colour tone means a higher number of expressions (of the seven) that passed our chosen cut off rate.
6.7. Results and discussion

Figure 6.10: Results of expression classifiers show equalised greyscale as best performer. The rates plotted use my new pa-SVM method and thus they are higher than the standard multiclass SVM.

show that certain expression classifiers are sensitive to the exact face location and the resolution used for detection. This suggests that a well chosen face detector resolution is required for robust classification of expression.

In the remaining subsections I present criticisms of the edge operators and discuss whether my chosen 25% bounding box for reporting the true positives rate is appropriate. Additionally, I discuss whether the decision values reflect the confidence of faces detected, and whether increasing the resolution of classifiers increases performance. The final section shows how the collection of optimal one-class face detectors can be used to detect and categorise different types of faces.

6.7.7 Were the edge operator parameter values appropriate over histogram equalised greyscale?

I mentioned in Section 6.7.1 that the edge operator parameter values were the same for non-equalised and histogram equalised greyscale. As a result artificial edges and over-emphasised edges appear. To compensate for this undesirable effect I explored an alternate set of different parameter values. For equalised Canny I changed the lower threshold to be the same as the upper threshold and left the aperture size unchanged – this is because at lower thresholds more
Figure 6.11: Expression classifiers over Sobel detected faces using the 25% bound. When considering the results of Figure 6.12, the Sobel expression classifiers shows best performance.
6.7. Results and discussion

Figure 6.12: Expression classifiers over the exact face location of the Sobel detected faces. The above results offer a relative comparison to the ground truth for the expression classifiers over the candidate Sobel detected faces of Figure 6.11.
edges appear and other aperture sizes would drastically increase the gain of
the edges or make them hardly visible. For equalised Sobel I reduced the
gain from 3 to 1. Figure 6.13 shows these new changes, and we can see that
they produce fewer artificial and emphasised edges than my current set of
parameters.

Figure 6.13: Alternative edge operators over histogram equalised greyscale.
The alternate edge operator produces fewer artificial edges and edges that are
not as over-emphasised than the current edge operator, but suffers in my null
test and window search of degraded cartoon faces.

Next, I show the cross validation results in Figure 6.14 comparing the edges
produced by the alternate edge operator to those produced by the current
operator. The results of the new changes seem good. However, when it comes
to my null test and window search, the returned performance starts to pale.
The null test in Figure 6.15 shows that every resolution of \( \nu \)-SVM and degree
2 polynomial sees faces in nothingness – except for the \( 15 \times 15 \) face detector.
With only the \( 15 \times 15 \) face detector passing the null test, the result of the
window search for the new changes was poor.

Due to the poor performance of the window search, it seems that the
current edges are preferred over the alternate edges. Further, we can observe
in Figure 6.13 that the \( 15 \times 15 \) face of Sobel using the current edge operators
Figure 6.14: Alternate edges perform better in cross validation results.
Figure 6.15: Alternate edges suffer under null test and window search.
shows more of a face-like appeal – as though the machine has preference to perceive significance from these over-emphasised edges - which is pareidolia.

6.7.8 Was my 25% bounding box an appropriate estimation of true positives?

My next question of interest was whether my 25% bounding box (for my window search) was an appropriate estimation of the true positives rate. What about a 50% or a 10% bounding box? If I went with a lower bounding box, say 10%, I would expect the averaged detection rate to reduce, and vice versa – as seen in Figure 6.16.

Figure 6.16: Comparing a 50%, 25% and 10% bounding box method for reporting true positives rate.
6.7.9 Low resolution for detection and slightly higher resolution for expression classification

My window search results using non-equalised images in Section 6.7.4 show that for face detection there were no clear benefits from having a higher resolution, since the best face detectors were a $30 \times 30$ and $38 \times 38$ classifier. From the literature review, most studies show improved detection rates at higher resolutions. However their higher resolutions can still be considered as very low. For instance, [237] showed an improved detection rate from $6 \times 6$ to $24 \times 24$ resolution using Modified Census Transform (MCT) and Boosting on colour images (MCT behaves similar to histogram equalisation where the aim is to compensate for lighting variations); [87] showed an improved performance from $6 \times 6$ to $24 \times 24$ resolution using an AdaBoost based face detector.

Compared to studies in the literature, my study investigated resolutions much greater than those reported. It shows that moving from very low to low resolutions (e.g. $15 \times 15$ to $24 \times 24$) improves the detection rate, however moving from low to high resolutions (e.g. $24 \times 24$ to $114 \times 114$) reduces the detection rate. The idea of returning a better performance at higher resolutions can only be seen for Canny in both equalised and non-equalised face detector results (Figure 6.7 of Section 6.7.2).

Similarly, the results of my expression classifiers (see Section 6.7.5) show that there was no clear benefit for having a high resolution expression classifier. This is because most of the low resolution classifiers produce the same results as the high resolution classifiers. In the literature review, most results show a reduced detection rate when compared to a higher resolution classifier. For instance, [132], [196], and [131] returned reduced performance when the pixel resolutions fell below $64 \times 64$, $110 \times 150$ and $72 \times 96$ respectively. Looking at the cross validation results for the expression classifiers (Figure 6.10 of Section 6.7.5), some specific SVM types and image preprocessing techniques do exhibit this trend. However, the highest cross validation rate can be seen just at the start of the resolution axis.

6.7.10 Do the decision values reflect confidence of faces detected?

A common consensus in the SVM learning community assumes that the decision values reflect the confidence of the classifier. To verify this consensus, I gathered all the decision values from the Sobel detected faces and averaged them over the number of samples. For the assumption to be true, the deci-
sion values near the centre of the test image should be the highest. The plots shown in Figure 6.17 agree with this assumption.

Figure 6.17: Decision values of Sobel detected faces do reflect the confidence of faces detected. A more saturated colour tone is a higher decision value. The very tiny circles are decision values that are very close to zero when averaged over the number of Sobel detected faces.

6.7.11 A framework for finding the face detectors of different face types

I gathered a large sample of different types of non-human faces from the creative commons flickr community, and tested the face detectors to see which face detectors are tuned for a particular face type. A sample of these faces is shown in Figure 6.18; the labels indicate the face detectors that returned a positive for that image - ‘G’ = greyscale, ‘S’ = Sobel. The Canny face detectors were not tested as they failed my null test in Section 6.7.3.

6.7.12 What if the cartoon faces were trained?

The experimental results leading to this point have shown robust performance for Sobel face detectors and expression classifiers. They were trained using human faces and returned good results for finding non-human face types – indicating that Sobel is generalisable. The cartoon faces were not trained because it is not a common technique used in the face detection and expression classification literature.

A question still sparks my attention, and perhaps that of the reader. ‘What if the cartoon faces were trained?’ Figure 6.19 shows the cross validation results for face detection and expression classification – they reflect a similar
Figure 6.18: Found faces of non-human face types. Labels below show the face detectors that returned a positive for that image - ‘G’ = greyscale, ‘S’ = Sobel. [Flickr images under a creative commons license - flickr IDs from the left: Alexandre Hamada Possi, aacckk, joseanavas, Marcus Meissner, nicocrisafulli and nottsexminer]
trend to the non-equalised Sobel results in Figure 6.7 (on face detection) and 6.10 (on expression classification).

Figure 6.19: One-class face detection and multiclass expression classification using cartoon faces.

My initial expectation was that the cartoon faces would exhibit results similar to Canny. This was simply because both are in binary form. However, my null test showed only the $114 \times 114$ classifier trained using a $\nu$-SVM, and the polynomial kernel failed (i.e. the classifier considered both black and white blank images as candidate faces). The window search for the cartoon faces in Figure 6.20 showed the $15 \times 15$ face detector to be the best.

Next, I compared two sets of results where the expression classifiers operated over the cartoon faces returned from the window search (which is similar to Section 6.7.6 with Sobel detected faces from the window search). Shown in Figure 6.21, the left set of results is where the expression classifiers operated over the cartoon faces detected from the window search (using the 25% bounding box method), while the right set of results is where the expression classifiers operated over the exact face location of the cartoon faces detected from the window search.

The poor performance in the left set of results of Figure 6.21 suggests that binary expression classifiers are particularly sensitive to the exact location of a candidate face – and not as robust as non-binary expression classifiers when compared to the results of Section 6.7.6. Further, the substantially higher number of detected cartoon faces over Sobel detected faces does account for
6.7. Results and discussion

Figure 6.20: The window search showed the $15 \times 15$ face detector to be the best, and it shows that the classifier was able to successfully detect the whole range of degraded cartoon faces. The reported true positive rates uses the 25% bounding box method.

Figure 6.21: Cartoon expression classifiers over the detected cartoon faces of the window search. The left is where the expression classifiers operated over the cartoon faces from the window search (using the 25% bounding box method) and the right is where the expression classifiers operated over the exact face location of the cartoon faces from the window search.
6.8. Summary

the poor performance.

6.7.13 Excluding the contemptuous expression returns better results

In the literature the contemptuous expression is often not considered. This expression is special because it is the only facial expression with an asymmetrical property. By excluding this expression the experiments return improved results as shown by Figure 6.22.

![Figure 6.22: Excluding the contemptuous expression returns better results.](image)

A similar finding was shown for face components in Section 7.8.7 – i.e. when face components such the eyes, brows and mouth were trained.

6.8 Summary

In this chapter I have introduced a new method for perceiving a scene using pareidolia of faces and expressions of emotion. The use of pareidolia of abstract faces and facial expressions to produce an emotion was first proposed in a previous work [25, 26] where this technique was applied to house designs. A concern with this idea was showing that we can empirically perceive faces and expressions. By using a set of feature points in the training data, I was able to develop a set of cartoon faces, which I degraded to provide a set of test images.
Using a number of image preprocessing techniques, resolutions and different configurations of classifiers for face detection and expression classification, I am able to show a set of parameters and classifiers that work best for my test case.

Essentially, I have contributed a framework where other image preprocessing techniques and a different set of test images can be substituted such that the robustness of the face detectors and expression classifiers can be empirically shown using a null test and a window search. With this simple procedure, you can pass in an image of a scene, and using the set of best parameters and classifiers, a description of this scene is produced as an array of emotional scores corresponding to a set of facial expressions of emotion.

A reason for experimenting with greyscale, histogram equalised greyscale, and their respective Sobel and Canny edges, is because they are common image preprocessing techniques found in the face detection and expression classification literature. However, the face detection literature is specific to finding human faces. I wanted the face detectors to be able to generalise to finding other face types, and testing upon degraded cartoon faces is one such instance.

In the next chapter I investigate component face perception techniques to further my discovery of a generalisable face detector and expression classifier.
Chapter 7

Component Face Perception

Contents

7.1 Introduction .................................................. 134
7.2 System overview ........................................... 137
7.3 Dataset and image preprocessing ......................... 138
7.4 Classification methods .................................... 141
  7.4.1 Support Vector Machines (SVM) and kernels ........ 141
  7.4.2 The SVM and kernel parameters to optimise .......... 141
  7.4.3 \(p\)-SVM framework .................................. 143
  7.4.4 SVM and kernel search parameters .................. 143
7.5 Component face detector ................................. 143
  7.5.1 Computing the Component Image ...................... 144
  7.5.2 Building the Component Height Map ................ 144
  7.5.3 Computing the Integral Rectangle .................... 144
  7.5.4 Merge conditions for overlapping rectangles ....... 144
  7.5.5 Matching rules to collate three shape components .. 145
7.6 Scale invariant feature vectors for semblance classifiers 146
  7.6.1 Geometry feature vectors .......................... 146
  7.6.2 Size feature vectors ................................ 147
  7.6.3 Shape feature vectors ................................ 147
7.7 Example walkthrough of my component face detection process ........................................... 148
7.8 Results and discussion .................................... 151
  7.8.1 Successful detection of different face types ........ 151
  7.8.2 Cases of unsuccessful detection of different face types 156
  7.8.3 Semblance classifiers using joined feature vectors return best results .................................. 158
  7.8.4 Eye faces are easily differentiated from brow faces using two-class SVMs ................................. 158
In the previous chapter I realised a perceptive machine with the aim to simulate the pareidolia capability of humans to produce an emotional response to a scene using facial expressions of holistically perceived faces. I described that these perceived faces can vary from natural faces to minimal face-like patterns. The term *holistic* means a system that processes full face images for training, detection and expression classification. Consequently, the performance is constrained by the type of image preprocessing, resolution and classifier configurations used on the training data.

Even with a good generalisable face detector (which I have shown in the previous chapter), only faces with similar configurations (i.e. geometry, size and shape of mouth and eyes) to that of the training data would be found. What about cartoon or caricature faces where the eyes and mouth are distorted beyond that of a natural human face? For a perceptive machine to be creative in perceiving these kinds of distorted abstract faces, a direction beyond a holistic path is explored.

In this chapter, I present a component approach designed to perceive faces where the eyes and mouth are distorted beyond that of a natural human face. The aim of this study is to simulate the pareidolia capability of humans to produce an emotional response to a scene using the expressions of the perceived face components. This novel process uses a component face detector and a
7.1 Introduction

In the previous chapter I presented to the reader the meaning of a scene (known as the ‘gist’) and specific brain regions that are active for processing scenes (the parahippocampal gyrus), faces (the fusiform gyrus) and emotions (the amygdala). The foundation of the previous chapter was inspired by the fusiform gyrus, which is thought to process faces holistically [234]. However, a limitation of a holistic system is that only faces with components that bear similar configurations (i.e. geometry, size and shape of mouth and eyes) to that of the training data would be found. For distorted face components to be detected (i.e. cartoons and caricatures), a direction beyond a holistic path is explored.

I begin by presenting the direction I explored. Next, I present the current literature on component based face detection – as well as my criticisms. Then finally, I present my proposed component based face detector.

To explore an alternative direction beyond a holistic path, I investigated component expression classifier. My face detector creates a number of component images from the original image at different resolutions. A component image is a binary edge image (using a Canny edge operator) where the edges are segmented using a connected components labelling method with a border-following technique – I call the segmented and labeled edges ‘components’. The component images are then overlaid to produce a component height map where large and notable components across all resolutions have high values, while specular and noisy components have low values. I retain three shape components that have values in the height map that are higher than the noise cut-off value. The three shape components represent two eyes and a mouth.

One-class \( \nu \) and SVDD SVMs are used to rank the three shape components by geometry, size and shape semblance to the natural human faces of the training data, via scale invariant feature vectors. Following the previous chapter, I use a pairwise adaptive SVM (\( pa \)-SVM) to assess the performance of using face components in classifying expressions. For each SVM experiment, a Gaussian and a degree 2 polynomial kernel were used with a 10 fold cross validation technique. The outcome is a component system with the potential to describe a scene by producing an array of emotion scores corresponding to the seven Universal Facial Expressions of Emotion. A particular advantage of this technique technique, when to a holistic method, is that the face components are explicitly isolated.
7.1. Introduction

Further literature from the Cognitive Science and Psychology community, and my investigation of the literature reveals that the inferior occipital gyrus is known to structurally process faces by components in a hierarchical order, giving preference to the eyes followed by the mouth and then the nose ([6] using fMRI with blood-oxygen-level dependence ‘BOLD’ responses). There are some studies that argued that the fusiform gyrus is also sensitive to face components [6, 108], but further studies have supported the theory of the fusiform gyrus processing faces holistically ([234] using fMRI with multi-voxel pattern analysis). To be technically in line with the neuroscience terminology, the specific region of the fusiform gyrus involved in processing faces is known by cognitive neuroscientists as the ‘fusiform face area (FFA)’ and the specific region of the inferior occipital gyrus involved in processing faces is known as the ‘occipital face area (OFA)’ [70]. There are debates as to whether a hierarchical order between the OFA and FFA exists – as currently a more complex interplay between the OFA and FFA is assumed [7]. An illustration of where the parahippocampal gyrus, amygdala, inferior occipital and fusiform gyri are located in the human brain is shown in Figure 2.1 of Chapter 2.

Inspired by the existence of a specific brain region known to structurally process faces by components, I investigated the literature from the machine learning community for candidate component based face detectors. A minimal requirement in my investigation was to look for face detectors that can detect at least the eyes and mouth. This is because I consider them to be the minimal set of features required to elicit a pareidolia recognition of faces from minimal face-like patterns. An example of this is the use of schematic face-like patterns in experiments with newborns that suggest that they are capable of perceiving such patterns [112]; implying that the brain may be hard-wired to look for faces from birth. Another requirement in my investigation was to avoid studies that use skin colour. This is because I do not want to limit the face detection to specific skin colours.

From my investigation, only a few component based face detection studies are relevant. They can be categorised as having either a top-down or bottom-up approach. An example of a top-down approach is a method that uses found faces, then isolates the face components using facial feature detectors – i.e. a Voila & Jones Haar-like face and facial feature detector for finding a face and then its mouth, nose, eyes and ears using expected regions of interest (ROI) for the search [21] (the Viola & Jones Haar-like feature detector detects objects using the Integral Image and has gained popularity because of its computation speed for finding faces [215]). Moving the focus to the studies that assume
7.1. Introduction

A face is found, a common technique for finding face components is using manually inserted feature points to train (or construct) a statistical model, and then searching for or aligning each feature point in the model over the found face. Several studies share this procedure – i.e. an Active Shape Model (ASM) to look for eyes, brows, nose, mouth and jawline [100], a Bayesian model with a consensus of exemplars to look for brows, eyes, nose, mouth and chin [16], a Conditional Regression Forest model to look for eyes, nose and mouth [40], and a multiple of Support Vector Machines (SVMs) and statistical Point Distribution Models (PDMs) to look for brows, eyes, nose and mouth [173]. An example of a bottom-up approach is a method that finds the face components first and then assesses their configuration using statistical models – i.e. a boosted cascade of classifiers to look for eyes, nose and mouth, and assess their topology using a PDM of face component centres [45], and similarly, an AdaBoost trained classifier cascade of Haar-like features to look for eyes, nose and mouth and assess their topology using graph matching techniques [75]. Lastly, [133] uses a weighted mask function and matching rules to look for faces in the form of isosceles or right-angled triangles where the corners correspond to face components, and the face components are extracted using a binarised greyscale image.

The methods currently available in literature for component based face detection do not fully meet my requirements. Firstly, (besides [133]) they are not aimed at looking for minimal face-like patterns since face components such as nose, brows, jawline and chin are considered. Secondly, the literature is specific to finding real human faces. And finally, the classifiers are trained using both positive and negative data. All cases except [133] do not appear to use negative data, and [133]’s detection depends on two threshold values - one at the initial step for creating a binarised greyscale image and the other at the decision making, which can be cumbersome. From the suggestion in the previous chapter that our brains are hard-wired to look for faces from birth [112], I hypothesised that the encoded knowledge would primarily be influenced by face data (of the positive class) rather than non-face data (of the negative class). Additionally, the performance of a classifier is strongly affected by the characteristics of the negative data. My hypothesis directs our attention to one-class classification methods, and from the literature review in the previous chapter, one-class SVM is my preferred choice [194].

Because the literature presented for component based face detection methods does not meet my needs, a new method is proposed. My proposed method works on the idea of allowing the raw features of the image to reveal the faces
within. I achieve this by using a connected components labelling method to segment and label the edges of an image into components. Then by analysing grouped components, distorted abstract faces can be perceived, as desired for simulating the pareidolia phenomenon. Using this approach creates a very generalisable component based face detector – it allows the eyes and mouth to be captured and explicitly isolated. The aim of this study is the description and an experimental evaluation of my proposed method using one-class SVMs and multiclass pa-SVMs [97]. The one-class SVMs are used for estimating the semblance of the detected face components to the faces in the training data via scale invariant feature vectors, and the multiclass pa-SVMs are used for classifying the expressions. The idea of my method appears similar to [133], which used triangles of face components to detect human faces; however, my approach offers several advantages – which I discuss in Section 7.8.

A system overview is presented in the next section, followed by a description of the dataset in Section 7.3. Section 7.4 presents the classification method, then my component face detector and the scale invariant feature vectors used for the semblance classifiers are described in Sections 7.5 and 7.6. An example walkthrough of my component face detection process is provided in Section 7.7. The experimental results and discussion are presented in Section 7.8, and in this section extensions of my component face detector is presented. Section 7.9 presents the literature revealing a component induced form of pareidolia (conveyed by simple line drawings and illusory contours), and the specific brain regions involved. Finally, future works in Section 7.10 and concluding remarks are made in Section 7.11.

## 7.2 System overview

My system consists of several steps as illustrated in Figure 7.1 for the face detection procedure, and Figure 7.2 for the classification process. Beginning with Figure 7.1, the first step downsamples an image into a number of low resolutions (by resizing it to a lower $\begin{pmatrix} n \times n \end{pmatrix}$ resolution, and then resizing it to a maximum resolution). A binary edge image is created using a Canny edge operator for each downsampled image. Then a component image is created using a connected components labelling method, and the most relevant components are then selected, representing respectively two eyes and a mouth, and are ranked by geometry, size and shape semblance classifiers trained using the natural human faces of the training data. Section 7.5 discusses this in detail.

The process for training the semblance and expression classifiers is shown
in Figure 7.2. The first step of Figure 7.2 was to align the face dataset using manually inserted feature points (purely for training). Then, using the feature points I created cartoon faces (described in the next section) and extracted the face components for the semblance classifiers (to assemble the scale invariant feature vectors reflecting the geometry, size and shape semblance of the faces in the training data) and the expression classifiers (by normalising the input of the face component classifiers between the values of -1 and +1 for training). Next, for both processes the dataset is partitioned into classes (seven classes for the expression classifiers and one class for the face detectors), and divided into training and validation sets using a $k$-fold cross validation algorithm (where $k$ is 10). The last step is to obtain the SVM and kernel grid parameters before running the experiments. Details of further steps are explained in subsequent sections.

7.3 Dataset and image preprocessing

In this section I describe further the face dataset, how the images were aligned and how the cartoon faces were generated, and the preprocessing step for preparing the face components for training the component expression classifiers.

Similar to Section 6.3, the face dataset was generated using the JACFEE and JACNEUF datasets, which are certified by the Paul Ekman Group LLC [167]. I specifically chose these datasets because they have discrete expression classes (the seven universal facial expressions of emotion as well as neutral faces). The generated dataset consists of 280 face images, including 140 neutral faces and 20 images of seven different facial expressions: happy, sad, surprised, fearful, disgusted, angry and contemptuous. I consider the neutral face separate from the seven universal facial expressions of emotion, and because of this, it was not used for expression classification; however, I used it for creating the semblance classifiers.

The face model defined by the manually inserted feature points was proposed in my previous publications ([96, 98] and Chapter 6) which showed a component approach performing better than a holistic approach for classifying discrete facial expressions. An illustration of my face model is shown in Figure 6.2 of Chapter 6, where a total of 60 feature points were manually inserted for each face image along the brows, eyes and mouth; the 2 feature points of the inner eye corners (a.k.a lacrimal caruncle) were used for aligning the faces in the dataset.
7.3. Dataset and image preprocessing

Figure 7.1: Overview of the component face detection process showing the flow of the stages and states.
7.3. Dataset and image preprocessing

Figure 7.2: Overview of the classification process showing the flow of the stages and states.
7.4 Classification methods

The cartoon faces for each image are created by drawing lines between the manually inserted feature points, and the face components (eyes, brows and mouth) are cropped using these feature points. Because the sizes of the cropped face components are different from one another, I found for each set of face components the maximum width and height of the cropped image, then padded the images to that maximum size.

Figure 7.3 shows sample images of the seven universal face expressions of emotion in greyscale, the corresponding cartoon images, and the cropped face components. You may notice that the width and height of the cropped face components between expressions are the same. This is because they have been padded to the maximum width and height found within that set.

I conducted a pilot study (as a sanity check) which showed that the performance of classifying white edges on a black background is equivalent to classifying black edges on a white background. In my experiments for classifying the expressions of face components, I use white edges on a black background because this is the standard output of the Canny edge operator.

7.4 Classification methods

Following the conclusions of Sections 3.1.3 and 3.1.2 I have chosen pa-SVM as my preferred classifier for creating the semblance classifiers and face component expression classifiers. Specifically, I have chosen to evaluate the C-SVM and ν-SVM for my study.

7.4.1 Support Vector Machines (SVM) and kernels

From the literature review of kernels with SVMs in Sections 3.2.3 and 3.2.2 I have chosen to conduct my experiments using a Gaussian and a degree 2 polynomial kernel.

7.4.2 The SVM and kernel parameters to optimise

From the theories of SVM and kernels in Chapter 3 and specifically Sections 3.7-3.8, my choice of the C-SVM and ν-SVM with a Gaussian and a degree 2 polynomial kernel has some parameters to tune. They are \((C, \gamma)\) for the C-SVM and \((\nu, \gamma)\) for the \(\nu\)-SVM – where \(C\) is a parameter for C-SVM, \(\nu\) is a parameter for \(\nu\)-SVM, and \(\gamma\) is a parameter for the Gaussian and polynomial kernel.
7.4. Classification methods

<table>
<thead>
<tr>
<th>greyscale</th>
<th>Cartoon</th>
<th>Cropped components</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="image" /></td>
<td><img src="image2" alt="image" /></td>
<td><img src="image3" alt="image" /></td>
</tr>
<tr>
<td><img src="image4" alt="image" /></td>
<td><img src="image5" alt="image" /></td>
<td><img src="image6" alt="image" /></td>
</tr>
<tr>
<td><img src="image7" alt="image" /></td>
<td><img src="image8" alt="image" /></td>
<td><img src="image9" alt="image" /></td>
</tr>
<tr>
<td><img src="image10" alt="image" /></td>
<td><img src="image11" alt="image" /></td>
<td><img src="image12" alt="image" /></td>
</tr>
<tr>
<td><img src="image13" alt="image" /></td>
<td><img src="image14" alt="image" /></td>
<td><img src="image15" alt="image" /></td>
</tr>
<tr>
<td><img src="image16" alt="image" /></td>
<td><img src="image17" alt="image" /></td>
<td><img src="image18" alt="image" /></td>
</tr>
<tr>
<td><img src="image19" alt="image" /></td>
<td><img src="image20" alt="image" /></td>
<td><img src="image21" alt="image" /></td>
</tr>
</tbody>
</table>

Figure 7.3: Sample images of the seven universal face expressions of emotion in greyscale, Cartoon and cropped face components. You may notice that the width and height of the cropped face components between expressions are the same. This is because they have been padded to the maximums found within that set. The face dataset was generated from the JACFEE and JACNEUF datasets [167].
7.5. Component face detector

7.4.3 pa-SVM framework
The pa-SVM framework of Chapter 4 is used for the face component expression classification.

7.4.4 SVM and kernel search parameters
I use the same \((C, \gamma)\) search parameters as [97] (Chapter 4) in my multiclass SVM experiments:

\[
C \in \{2^{-5}, 2^{-4}, \ldots, 2^{14}\}
\]

and

\[
\gamma \in \{2^{-17}, 2^{-16}, \ldots, 2^{2}\}
\]

And likewise, for the sequences of \((\nu, \gamma)\) I grow \(\nu\) by a linear step of 0.05 up to 1.0, while \(\gamma\) remains the same. i.e.

\[
\nu \in \{0.05, 0.10, \ldots, 0.95, 1.00\}
\]

For my one-class SVM experiments, which use only \(\nu\) and \(\gamma\), I use the same \((\nu, \gamma)\) values as described for the multiclass SVM experiments.

7.5 Component face detector

In this section I describe the detailed steps of my component face detection process, which was presented in the system overview section in Figure 7.1. For the initial step of downsampling, I have chosen 10 resolutions with a height and width of 15, 19, 24, 30, 38, 47, 59, 73, 92 and 114 pixels (a scaling up of 1.25). Then I have chosen the Canny edge operator to compute the binary edges. The Canny edge operator had several parameters that impact the final edge image. Selection of the parameters was by visual evaluation for ideal face edges. The chosen parameters were: aperture size of 3, lower threshold of 127 and upper threshold of 255. Next, I obtained the component images using a connected components labelling algorithm with a border-following technique [200].
7.5. Component face detector

7.5.1 Computing the Component Image

I segmented the edges into ‘components’. For each component, I construct a rectangle around those pixels – defined by $x_{\text{min}}, x_{\text{max}}, y_{\text{min}}, y_{\text{max}}$. Because the components are described by pixels and rectangles, there are two possible methods for building the height map. One using the pixels, ‘pixel-based’, and the other using the rectangles, ‘block-based’ – an equal weight was chosen for overlaying the components of the ‘pixel-based’ and ‘block-based’ approach.

7.5.2 Building the Component Height Map

The main purpose of the height maps is to provide regions of interest (ROI) defined by global and local peaks. I consider these peaks to hold relevant candidate face components because they are prominent in all the component images. Bear in mind, these peaks are made up of a number of components from each component image; where each component is a rectangle described by four values ($x_{\text{min}}, x_{\text{max}}, y_{\text{min}}, y_{\text{max}}$).

7.5.3 Computing the Integral Rectangle

To model the sum of the components contributing to each peak I construct a new rectangle, which I call an ‘integral rectangle’. This process is by averaging each of the four corner points over the selected rectangles, and masking the area to be ignored from further processing. To begin computing the integral rectangles I start from the highest point of the height map down to a noise cut-off (which is defined by an average of non-zero height values). For each operating point, I selected all the rectangles (derived from the component images) containing that point. A heuristic is used to reject large selected rectangles by downsampling. This heuristic averages the pixel height value for each component, and if it falls below the noise cut-off value it is rejected. The purpose of this is to allow the computed integral rectangle with the tightest fit to the peak region.

7.5.4 Merge conditions for overlapping rectangles

The next step is to decide whether to merge overlapping integral rectangles (they usually overlap due to high height values residing next to masked regions). Two rectangles are merged if they satisfy two conditions. The first
condition is that the two rectangles have the same aspect ratio. The second condition is that the height values (from the component height map) are within a threshold:

\[
\text{threshold} = \frac{\sum_{i=k}^{m-1} (y_{i+1} - y_i)}{m - k}
\]  

(7.1)

where \(y_i\) is the count of height values along the \(x\) axis from low to high, \(m\) is the number of height values, and \(k\) is the height value along the \(x\) axis that is above the noise cut-off. The purpose for the threshold is to prevent the merging of components with largely different height values. A walkthrough example is given in Section 7.7, which shows the plots of the height values versus count with the noise cut-off.

### 7.5.5 Matching rules to collate three shape components

Once the merge operation is complete I use a set of matching rules to produce a set of three shape components, where the shapes are the integral rectangles from the previous step – representing (respectively) two eyes and a mouth. The current matching rules are:

1. mouth rectangle center below and between left and right eye rectangles.
2. top of one eye is not below the bottom of the other eye.

The final step is to determine a confidence rating for the candidate faces. I evaluated 2 metrics for this purpose. The first metric sums the height values of each face component using the component height map; a higher sum reflects a higher confidence rating. This is then compared with the decision values of the one-class semblance classifiers. The decision value is considered a confidence rating in relation to the training data, i.e. if the decision value is positive and it is very high, it means that the classifier sees this as a face and the components are very similar to the face components of the training data. Then, the expressions of the face components are determined. This is by classifying the face components after they have been resized, preprocessed and padded to the maximum widths and heights used for training the component expression classifiers (see Section 7.3). Details of the feature vectors used for the semblance classifiers are described next.
7.6 Scale invariant feature vectors for semblance classifiers

In this section I describe the feature vectors used for the semblance classifiers that were presented in my system overview section in Figure 7.2. As described earlier, my goal is to create scale invariant feature vectors that model the geometry, size and shape semblance from the natural faces of the training data. To begin, I construct rectangles around the face components in the training data using the manually inserted feature points – i.e. brows, eyes and mouth. I use angles, area and aspect ratios to describe the geometry, size and shape of the face components. Figure 7.4 is used to assist with the description.

\[ \angle \text{mid eyes} \quad (7.2) \]
\[ \angle \text{mid mouth} \quad (7.3) \]

where \( \angle \text{mid mouth} \) is shown as \( m \) in Figure 7.4 and \( \angle \text{mid eyes} \) is the opposite angle. Both angles use radians as input to the SVM.
7.6. Scale invariant feature vectors for semblance classifiers

7.6.2 Size feature vectors

For size, I proposed 2 feature vectors:

\[
\frac{(L + R)}{2M} \quad (7.4) \\
\frac{(L - R)}{M} \quad (7.5)
\]

where \(L, R, M\) represent (respectively) the areas of the left eye, right eye and mouth; defined by the width multiplied by the height of the rectangle. The meaning of the above feature vectors correlates to:

- Average area of the eyes divided by the area of the mouth (7.4).
- Eye area asymmetry divided by the area of the mouth (7.5).

7.6.3 Shape feature vectors

Lastly, for shape, I proposed 3 feature vectors:

\[
\frac{(L_y/L_x + R_y/R_x)}{2} \quad (7.6) \\
2\frac{(L_y/L_x - R_y/R_x)}{(L_y/L_x + R_y/R_x)} \quad (7.7) \\
\frac{M_y/M_x}{M_y/M_x} \quad (7.8)
\]

where \((L_y, L_x)\), \((R_y, R_x)\) and \((M_y, M_x)\) are the aspect ratios of the left eyes \((L)\), right eyes \((R)\), and mouth \((M)\) with an aspect height defined by \(y\) and an aspect width defined by \(x\). The meaning of the above feature vectors correlates to:

- Average aspect ratios of the eyes (7.6).
- Scaled eye aspect ratio asymmetry (7.7).
- Mouth aspect ratio (7.8).

In my experiments I assessed the cross validation performance when I joined the feature vectors together and when more data was added from different resolutions. As an extension, the above procedure was repeated for brows instead of eyes; this is because the three shape configuration is applicable to both.
7.7. Example walkthrough of my component face detection process

I use Figure 7.5 to provide a walkthrough of the steps of my component face detection and expression classification method that was described in Section 7.5.

Figure 7.5: Example image for describing the steps of my component face detection and expression classification method. [Flickr image under a creative commons license, taken by flickr ID: Alexandre Hamada Possi].

The component images of the initial downsampling step using the 10 chosen resolutions are shown in Figure 7.6.

The two approaches (‘pixel-based’ and ‘block-based’) for building the component height maps are shown in Figure 7.7, with an equal weight for overlaying the components.

Figure 7.8 shows the plots of the height values versus the count, and the noise cut-off, which separates noisy and relevant regions. The height maps of the relevant regions are shown below in the same figure. I use the height value counts above the noise cut-off shown in the top graphs of the figure for calculating the threshold used to determine a merge condition, as described in Section 7.5.4.

The top face candidates with the highest summed height values (from each face component) and colour coded component expressions are shown in Figure 7.9; on the left is the result of the ‘pixel-based’ approach where the left eye is fear (grey) and the right eye and mouth is surprised (violet); on
7.7. Example walkthrough of my component face detection process

Figure 7.6: Component images of my example walkthrough

Figure 7.7: A ‘pixel based’ and a ‘block based’ component height map with an equal weight for overlaying the components.
7.7. Example walkthrough of my component face detection process

Figure 7.8: Finding the noise cut-off value and showing the relevant regions of the component height maps
7.8. Results and discussion

I begin this section by presenting a number of different face types detected and expressions classified, followed by cases of unsuccessful detection. Then I go into the individual classification results, starting with the best cross validation rates of the semblance classifiers using the scale invariant feature vectors. I then show that the feature vectors can easily distinguish three shape eye faces from three shape brow faces (but are inadequate for expression classification). Next, I show that the mouth expression alone is almost equivalent to that of the whole face.

Later in this section, I present further experiments to show the impact of excluding the contemptuous expression (which is common in the literature), followed by a close comparison of my method with the literature.

7.8.1 Successful detection of different face types

A large sample of pareidolia images were randomly selected from flickr under the creative commons license and manually cropped. In this section I show the different face types detected and the expressions classified in Figures 7.10 and 7.11. The faces shown have the highest summed height values (from each face component) of all the face candidates. The colour code for each component expression is as follows: sad (blue), angry (red), surprised (violet), fearful (grey) and the right eye and mouth is surprised (violet).

The decision values of the one-class classifiers for both approaches are also shown in Figure 7.9. In both situation, the one-class classifier using a $\nu$ SVM and the polynomial kernel returns a positive value. The positive value indicates a positive face candidate and the number is considered a general indication of the classifier’s confidence rating ([99]), but is not comparable across kernels. In the same figure the classified expressions are shown below. The four multiclass expression classifiers show a unanimous decision for the expression of the left eye, right eye and mouth.

Given the above example, it is possible to arrive with conflicting expressions between face components. One question springs to mind, ‘Which face component expression is a reliable indicator?’ My results in Section 7.8.6 show that the mouth expression returns a cross validation rate almost equivalent to that of the whole face.
7.8. Results and discussion

Figure 7.9: Top candidate faces detected and classification of expressions. In both situations the one-class classifier using a $\nu$ SVM and the polynomial kernel returned a positive value. The positive value indicates a positive face candidate and the number is considered a general indication of the classifier’s confidence rating ([99]), but is not comparable across kernels. The four multiclass expression classifiers show a unanimous decision for the expression of the left eye, right eye and mouth.
(grey), disgusted (green), contemptuous (orange) and happy (yellow). The SVM type and the kernel of the one-class classifiers that returned a positive decision value are shown directly below the image. As mentioned earlier, for conflicting face component expressions I show in Section 7.8.6 that the mouth expression returns a cross validation rate almost equivalent to that of the whole face.

Examining the two figures, the first observation is that the expressions of face components are sensitive to the size and location of the bounding rectangle. A larger bounding rectangle would prevent proper alignment of face components to the trained images; for instance, the first image in Figure 7.10 by ‘joseanavas’ and the yellow image by ‘thentoff’ (fifth row), where the larger bounding rectangle shifts the alignment of the mouth to prevent a surprised expression being properly classified. Similarly, if a bounding rectangle is small enough to clip important edge features, a different expression would be seen. For example, the clock face image in Figure 7.10 by ‘Philipp C.J.’ and the image by ‘nottsexminer’, where in both cases, a smaller bounding rectangle returns a disgusted expression.

The bounding rectangle is too large in cases where at too many resolutions the feature of interest is joined with another component (for example, see the mouth component in Figure 7.6 of Section 7.7). In this situation, the size and location of expected ‘integral rectangles’ are shifted. For instance, in the house image in Figure 7.10 by ‘1541’ and the earphone image by ‘thentoff directly below, the shift in location of face components shows that the ‘block-based’ approach is reliable. This shift can be seen in the component height maps of Figure 7.12, where the heights below the noise cut-off have been suppressed (i.e. the regions described in the walkthrough of Section 7.7).

The larger than expected components obtained by the border-following connected components labelling method ([200]) can increase the size of face components, and in my current experiments I have rejected face candidates where face components overlap. Because of this, there are several examples in Figure 7.10 where no face candidates were returned by the ‘pixel-based’ approach.

Lastly, as I was gathering the decision values of the one-class semblance classifiers, I noted that in all cases the \((\nu, poly)\) one-class classifier returned positive decision values, followed by the \((\nu, rbf)\) one-class classifier. The \((svdd, poly)\) and \((svdd, rbf)\) one-class classifiers produced mostly negative decision values, which means they did not see the top candidate faces as positive faces.
### 7.8. Results and discussion

<table>
<thead>
<tr>
<th>Image</th>
<th>Block-based</th>
<th>Pixel-based</th>
<th>Image</th>
<th>Block-based</th>
<th>Pixel-based</th>
</tr>
</thead>
<tbody>
<tr>
<td>joseanavas</td>
<td>(ν, poly)</td>
<td>(ν, poly)</td>
<td>rorrorr</td>
<td>(ν, poly)</td>
<td>(ν, poly)</td>
</tr>
<tr>
<td>Philipp C. J.</td>
<td>(ν, poly)</td>
<td>(ν, poly)</td>
<td>phil dokas</td>
<td>(ν, poly)</td>
<td>(ν, poly)</td>
</tr>
<tr>
<td>1541</td>
<td>(ν, poly)</td>
<td>(ν, poly)</td>
<td>notsexminer</td>
<td>(ν, poly)</td>
<td>(ν, poly)</td>
</tr>
<tr>
<td>thentoff</td>
<td>(ν, poly)</td>
<td>(ν, poly)</td>
<td>thentoff</td>
<td>(ν, poly)</td>
<td>(ν, poly)</td>
</tr>
<tr>
<td>thentoff</td>
<td>(ν, poly)</td>
<td>(ν, poly)</td>
<td>xenmate</td>
<td>(ν, poly)</td>
<td>(ν, poly)</td>
</tr>
<tr>
<td>chechar</td>
<td>(ν, poly)</td>
<td>(ν, poly)</td>
<td>Jordiet</td>
<td>(ν, poly)</td>
<td>(ν, poly)</td>
</tr>
</tbody>
</table>

Figure 7.10: Different face types detected and the expressions classified. The faces shown have the highest summed height values (from each face component) of all the face candidates. The colour code for each component expression is as follows: sad (blue), angry (red), surprised (violet), fearful (grey), disgusted (green), contemptuous (orange) and happy (yellow). The SVM type and the kernel of the one-class classifiers that returned a positive decision value are shown directly below the image. [Flickr images under a creative commons license - the flickr IDs are shown below the first image]
<table>
<thead>
<tr>
<th>image</th>
<th>block-based</th>
<th>pixel-based</th>
<th>image</th>
<th>block-based</th>
<th>pixel-based</th>
</tr>
</thead>
<tbody>
<tr>
<td>duncan</td>
<td>(ν, poly)</td>
<td>(ν, poly)</td>
<td>eighttrees</td>
<td>(ν, poly)</td>
<td>None</td>
</tr>
<tr>
<td>ugglan</td>
<td>(ν, poly)</td>
<td>(ν, poly)</td>
<td>kencf0618</td>
<td>(ν, poly)</td>
<td>(ν, poly)</td>
</tr>
<tr>
<td>kencf0618</td>
<td>(ν, poly)</td>
<td>None</td>
<td>aacckk</td>
<td>(ν, poly)</td>
<td>None</td>
</tr>
<tr>
<td>Aidan-Sally</td>
<td>(ν, poly)</td>
<td>(ν, poly)</td>
<td>plindberg</td>
<td>(ν, poly)</td>
<td>None</td>
</tr>
<tr>
<td>Dru!</td>
<td>(ν, poly)</td>
<td>(ν, poly)</td>
<td>Dru!</td>
<td>(ν, poly)</td>
<td>(ν, poly)</td>
</tr>
<tr>
<td>ms4jah</td>
<td>(ν, poly)</td>
<td>(ν, poly)</td>
<td>grendelkhan</td>
<td>(ν, poly)</td>
<td>(ν, poly)</td>
</tr>
</tbody>
</table>

Figure 7.11: Different face types detected and the expressions classified. The faces shown have the highest summed height values (from each face component) of all the face candidates. The colour code for each component expression is as follows: sad (blue), angry (red), surprised (violet), fearful (grey), disgusted (green), contemptuous (orange) and happy (yellow). The SVM type and the kernel of the one-class classifiers that returned a positive decision value are shown directly below the image. [Flickr images under a creative commons license - the flickr IDs are shown below the first image]
7.8. Results and discussion

Figure 7.12: Example of component height maps shifting location of face components. The reason for the shifts in the ‘pixel-based’ approach can be deduced by examining the cluster of heights in the ‘block-based’ height map, and the highest height values of the ‘pixel-based’ height map.

7.8.2 Cases of unsuccessful detection of different face types

In Figure 7.13, I have selected a few examples of pareidolia images where no face candidates were produced. We can see that the ‘integral rectangles’ are present for each face component; however, the rectangles are either overlapping each other or they have been merged with other rectangles that cause the face components to overlap, thus preventing them being a face candidate. The images shown are from the ‘block-based’ approach, and only the height values in the height map that are above the noise cut-off are shown. See Section 7.10 for ideas as future work to improve these cases.
Figure 7.13: Example of unsuccessful detection of different face types. We can see that the ‘integral rectangles’ are present for each face component; however, the rectangles are either overlapping each other or they have been merged with other rectangles that cause the face components to overlap, thus preventing them being a face candidate. The images shown are from the ‘block-based’ approach, and only the height values in the height map that are above the noise cut-off are shown.
Next I present the individual classification results of the scale invariant feature vectors used for the semblance classifiers, the face components used for the expression classifiers, followed by the impact when the contemptuous expression has been excluded in the classification process.

7.8.3 Semblance classifiers using joined feature vectors return best results

The one-class cross validation results of Figure 7.14 show best rates when the geometry, size and shape feature vectors are joined using the $\nu$-SVM and the polynomial kernel, and this applies to either an eyes-mouth three shape or a brows-mouth three shape configuration. In particular, the results are also good when the feature vectors are combined from a number of different face resolutions – as seen by the graphs in the right column of Figure 7.14. The reason for combining the feature vectors from different face resolutions is to provide more data for training.

7.8.4 Eye faces are easily differentiated from brow faces using two-class SVMs

The two-class cross validation results of Figure 7.15 show that the feature vectors using eyes and mouth can be easily differentiated from the feature vectors using the brows and mouth. The highest results are when the geometry, size and shape features are joined (i.e. seven feature vectors); followed by shape (3 feature vectors using aspect ratios), geometry (2 feature vectors using angles) then size (2 feature vectors using areas).

7.8.5 Feature vectors are inadequate for expression classification

The multiclass cross validation results of Figure 7.16 show poor results for the classification of expressions using feature vectors. The feature vectors are derived from rectangles of face components. Because the rectangles do not contain sufficient discriminatory information these results are expected.
Figure 7.14: Semblance classifiers using joined feature vectors, with the $\nu$-SVM and the polynomial kernel returning best rates. This applies to either an eyes-mouth three shape or a brows-mouth three shape configuration – as well as combining the feature vectors from a number of different face resolutions – where ‘1sthalf’ is combined data of face resolutions with a width and height of 15, 19, 24, 30, 38 pixels; ‘2ndhalf’ is combined data of face resolutions with a width and height of 47, 59, 73, 92, 114 pixels, and ‘allres’ is combined data of all 10 face resolutions.
Figure 7.15: Best results for separating eye faces from brow faces when geometry, size and shape feature vectors are joined. This applies to combining the feature vectors from a number of different face resolutions – where ‘1sthalf’ is combined data of face resolutions with a width and height of 15, 19, 24, 30, 38 pixels; ‘2ndhalf’ is combined data of face resolutions with a width and height of 47, 59, 73, 92, 114 pixels, and ‘allres’ is combined data of all 10 face resolutions.
7.8. Results and discussion

Individual resolutions

Combined resolutions

Figure 7.16: The feature vectors are derived from rectangles of face components. Because the rectangles do not contain sufficient discriminatory information these results are expected. This applies to either an eyes-mouth three shape or a brows-mouth three shape configuration – as well as combining the feature vectors from a number of different face resolutions – where ‘1sthalf’ is combined data of face resolutions with a width and height of 15, 19, 24, 30, 38 pixels; ‘2ndhalf’ is combined data of face resolutions with a width and height of 47, 59, 73, 92, 114 pixels, and ‘allres’ is combined data of all 10 face resolutions.
7.8. Results and discussion

7.8.6 Expression analysis of face components shows mouth with the best result

The multiclass cross validation results of Figure 7.17 show that the mouth is the best feature for discriminating between expressions. When compared with the holistic results (in the same figure), the mouth appears to be equivalent to the results involving the complete face.

![Figure 7.17](image)

Figure 7.17: Expression analysis of face components shows that the mouth is the best feature for discriminating between expressions. The left graph shows the results of face components extracted from whole face resolutions, while the right graph presents complete faces classified by holistic expression classifiers. The darker colours of the left graph represent left face components while the lighter colours represent the right face components.

7.8.7 Excluding the contemptuous expression returns better results

In the literature the contemptuous expression is often not considered. This expression is special because it is the only facial expression with an asymmetrical property. By excluding this expression the experiments return improved results as shown by Figure 7.18.

A similar finding was shown for holistic faces in Section 6.7.13 – i.e. when the entire face image was trained. Next, I present the comparison of my
7.8. Results and discussion

Excluding contemptuous expression returns better results.

Figure 7.18: Excluding the contemptuous expression returns better results.

method to the literature, then further literature from Cognitive Science and Psychology is presented to reveal a component induced form of pareidolia and the specific brain regions involved. Finally, I present the possible improvements for future work and my concluding remarks.

7.8.8 Comparison to the literature

As I have mentioned in the introduction, my method appears to follow a similar idea to [133], which used triangles of face components to detect human faces. My approach differs in the following ways. Firstly, I use binary edges as the initial step rather than a binarised greyscale – this should be more robust than a manually tuned threshold. Secondly, instead of using weighted masked functions and providing a threshold value for decision making, I construct a height map and use the combined peak heights of each face component to quantify the confidence of a face. This is then complemented by decision values (which reflect confidence) from the one-class classifiers. To clarify, in [133] it was not clear whether the threshold value for tuning the binarised greyscale image (and for the decision making in the face verification stage) is constant or varies with each test image, or whether it was an automated process.
7.9 Literature revealing a component induced form of pareidolia and the specific brain regions involved

Using functional magnetic resonance imaging (fMRI) and dynamic causal modelling (DCM), a study has shown that a triangular network exists between the ‘occipital face area’ (OFA), ‘fusiform face area’ (FFA) and the lateral occipital cortex (LO) [153]. The results of the study suggest that the LO has a bidirectional interconnection with both the FFA and the OFA – “supposing that the LO plays a general and important input region role [153]”. The main role of the lateral occipital cortex (LO), however, is known to be the processing of objects ([151] using Transcranial magnetic stimulation (TMS)); in particular, shapes – either visually ([118] using fMRI, [150] using TMS) or from sound ([109] using fMRI). The LO is part of the lateral occipital complex (LOC) which includes the posterior fusiform gyrus (pFs) involved in object recognition [79]. Similar to the LO, the LOC is sensitive to shapes visually ([119] using fMRI and BOLD responses), and through sound ([192] using positron emission tomography ‘PET’), as well as touch ([120] using fMRI).

I found in the literature review that the number of studies involving the LOC is generally greater than the number of studies focussing on the LO. In particular, some of those LOC studies are relevant to scene perception and pareidolia. Relevant to scene perception, a study on the neural coding of natural scenes ([165]) suggested that distributed and complementary roles are assigned to the LOC and PPA (‘parahippocampal place area’ the specific region in the parahippocampal gyrus for processing scenes [55]) – where spatial boundary information is primarily processed in the PPA while the scene content is primarily processed in the LOC. Relevant to pareidolia, there are studies that show that the LOC responds to illusory contours and contour integration [4, 198]. Contour integration is “the linking of collinear but disconnected visual elements across space, [which] is an essential facet of object and scene perception [198]”. Illusory contours are black and white images of ordered shapes, which are used to show our capability in manifesting missing information to see illusions of other shapes [4] – mostly in the form of partial silhouettes. I consider this technique a component induced form of pareidolia, processed without our awareness [170]. This concept bears similarity to a scene perception study where simple line drawings suffice to show our capability in categorising natural scenes of beaches, city streets, forests, highways,
7.10 Future work

In this section I describe possible improvements to my component face detector and the expression classifiers as future work.

7.10.1 Improve connected components labelling algorithm

We have seen in Section 7.8.1 (and 7.8.2) that my component face detector is capable of detecting many different face types without the need for SVMs. There were two cases in my discussion that showed the location shifts of face components between the ‘block-based’ and ‘pixel-based’ approach, and this shift is due to the way the border-following technique segments the edges into components. A suggested improvement is for the connected components labelling algorithm to take into account several resolutions of an image and use this new information to guide the allocation of pixels to components. My assessment is that at higher resolutions, good edges break up and become separate components (the component images in Figure 7.6 of Section 7.7 are a good example). But the problem lies at lower resolutions where distinct edges are merged and the large components skew the expected size of ‘integral rectangles’. A possible scenario is to give the edges at the lower resolution a smaller weight when integral rectangles are computed; as currently an equal weight is given for overlaying the components of the height maps from each resolution, and when the four corners of the integral rectangle are computed, the average of the corners of large components at lower resolutions are treated equally.

7.10.2 Improve merge conditions

The current merge conditions work by using the same aspect ratio and the idea of not merging two rectangles with largely different height values. However, in some cases, there are large integral rectangles with the same aspect ratio and height values, which allow them to merge. This may be due to the issue described above regarding large components at lower resolutions. In my experiments, I could have introduced further conditions to tackle this problem; however, I am more interested in obtaining and presenting the results based
7.10. Future work

on first order conditions which can be built upon, and from my assessment
the current merge conditions have performed well.

7.10.3 Correcting size of rectangles for aligning face components to the trained images

I discussed in Section 7.8.1 that the size of the rectangles prevent proper alignment of face components to the trained images, which can skew the classified expression. The binary edges shown in the figures of Section 7.8.1 are passed through the Canny edge operator at the maximum resolution. A possible path is to examine the component image at the maximum resolution and correct the size of the bounding rectangle. The solution would usually be easy if the face component fits within the current bounding rectangle, as there would exist a finer component at the maximum resolution. However, it is possible that at the maximum resolution, there can be multiple face component images that extend beyond the bounding rectangle, necessitating further analysis.

7.10.4 Incorporating colour information

Let’s start with an example. Suppose that we look for face components in a yellow house with a gable roof where the windows are the eyes and the door as the mouth. The house is set along a clear blue sky where the edge of the gable roof is distinct. Because of the distinct edge, a component is formed where the rectangle surrounds the gable roof, or perhaps at high resolutions two components are formed. In either case, they cover the windows, and when the height map is constructed, the area around the windows would still have higher values. However, if I use the colour information of the sky, I could discard the component along the roofline without affecting the end result. Using colour information can essentially localise the perception of face components, however, supposing global perception gives two clouds as the eyes and a door as the mouth, this would not be considered as a face candidate.

7.10.5 Clustering algorithm for multiple face detection

The current problem domain is to look for face components in non-human face types where a close up image is shown with a pareidolia effect of a face. In the results there are pareidolia images with many face candidates; however, only the one with the highest summed height values (from each face component) is considered and evaluated against one-class classifiers. To cater for multiple
pareidolia faces, a clustering algorithm would be required to calibrate the precision of this technique. Currently, multiple face detection in the literature uses colour based segmentation; however, this may not necessarily be useful for perceiving pareidolia faces.

7.11 Summary

In this chapter I have extended my work of producing an emotional response to a scene from holistically perceived faces, to perceiving face components such as the eyes and mouth, and classifying the expressions of emotions. The idea is to detect abstract faces where the eyes and mouth are distorted beyond that of a natural human face. The significance of this study is to simulate the pareidolia capability of humans to produce an emotional response to a scene using expressions of perceived face components.

From the literature review there were only a few relevant studies that were aimed at detecting faces by face components; however, they were all specific to finding human faces, and they have shortcomings that meant they did not fit my requirements. Due to these reasons I have developed a new component face detector, which is capable of detecting many different face types. A particular advantage of this technique, when compared to a holistic method, is that the face components are explicitly isolated. This was achieved by using heuristics to define a suitable metric for evaluating a range of candidate faces.

To complement the component face detector I trained several one-class classifiers using scale invariant feature vectors derived from the face components (i.e. eyes and mouth) to capture the geometry, size and shape of the face components. The feature vectors were then evaluated using four one-class classifiers (a combination of $\nu$ and SVDD SVM type with the Gaussian and the polynomial kernel). My assessment of the output of the classifiers showed that in all cases the ($\nu$, poly) one-class classifier agreed with the faces detected, followed by the ($\nu$, rbf) one-class classifier.

Further experiments were conducted to show how the feature vectors (which consist of only seven dimensions) are capable of distinguishing three shape faces of two eyes with a mouth from those made up of two brows with a mouth; however, these feature vectors were not adequate enough to distinguish expressions. To create the component expression classifiers I sliced the dataset of cartoon faces into face components – brows, eyes and mouth. This was possible because of the feature points I manually inserted to photographed human faces, which is part of my face model (see Chapter 5). My evaluation
of the component expression classifiers shows that the classification of the mouth expression is almost equivalent to that of the whole face, while the eyes and brows were harder to determine across the seven discrete expressions of emotions.

Concluding the chapter summary, the strengths as well as the limitations of my component face detector have been identified with possible improvements as future work. This chapter concludes the contributions achieved to addressing my thesis question, ‘Can a machine describe the emotional response to a scene using perceived faces and facial expressions?’ In the final chapter I refer to the materials presented in this and previous chapters to affirm the importance of my efforts.
8.1 Motivation

In Chapter 1, I started with a simple story:

You are currently standing in the middle of a suburban area and observing a row of houses. A feeling in your gut says one house seems not to belong, or perhaps one house stands out to your liking. When asked whether you can see a face in this particular house, to some this may not be obvious at first, however to many they can almost instantly perceive areas of the house where an eye would be, and then a mouth. By taking a step back, you see another face and this time the neatly trimmed garden hedge forms a mouth. You begin to ask yourself, are these faces happy or sad? Or maybe angry? You run them by your gut feeling, and with a rush of excitement you become aware.

The above story served a number of purposes. Firstly, it introduced the term ‘pareidolia’ – the ability to see significance in random and vague stimuli (i.e. windows as eyes and a door as a mouth). Secondly, it describes my motivation and subsequently leads to my thesis question, which is asking whether a machine can see a face or a collection of faces in a scene where a face does not exist, and whether the machine is capable of assessing their expressions – and if so, what would they be? In a simplified form, ‘Can a machine describe the emotional response to a scene?’

8.2 Research to support the relevance of my thesis question

Investigating the thesis question led to specific questions, ‘How can a machine describe a scene?’ and ‘What features would it use?’ The literature shows
that faces are a useful feature for my machine to describe a scene, and thus satisfy the definition of pareidolia. However, I believe that the significance is beyond just perceiving abstract faces in a scene, it is the ability to produce an emotional response to it – and this emotion is drawn from facial expressions. With this notion, my simplified thesis question become more specific, ‘Can a machine describe the emotional response to a scene using perceived faces and facial expressions?’

To understand whether my question was relevant I undertook an extensive literature review into Cognitive Science and Psychology. This was the core work of Chapter 2. From that chapter, important physiological connections in the brain were revealed to answer questions such as ‘Why faces?’, ‘What is it that makes the face a unique feature for scene perception?’ and ‘What about facial expressions? What purpose do they serve?’ The revealed connections are specific regions and neural pathways in the brain that are dedicated for processing face perception (the inferior occipital and fusiform gyri), scenes (the parahippocampal gyrus) and emotions (the amygdala).

The revelation of specific brain regions and neural pathways suggests that information is rapidly processed beyond our consciousness, i.e. within the subconscious. This raises another question, ‘How does the subconscious relate to my thesis question?’ When we perceive a scene, we may lack the ability to fully describe our emotional response to it as the answers in our conscious mind are limited by our ability to question our subconscious. The rapid rate at which the subconscious is processed suggests that it is rich in information, and the ability of a machine to relay this information allows our emotional response to a scene to become transparent.

Additionally, regarding pareidolia, a recent study suggested that faces can be perceived in minimal face-like patterns even when these features are not centred in our visual focus. This finding is relevant as it justifies that a machine does not need to exhibit any special attention to searching for faces at or near the centre of an image (under the assumption that the centre of the image is the centre of visual focus). Another study was presented to show that the ranking of face likeness to genuine faces is modulated by the left fusiform gyrus – this was replicated in the experiments of the final chapters.

The research on pareidolia was important because the literature showed that it is an ability that exists in the general population, and that it is linked to creativity. The link to creativity led us to question whether a machine that is created to perceive a scene through pareidolia is replicating the creative process of an individual, and if so, would it be considered creative? A connection was
then shown that links the issue of creativity to the cultural survival of historical artwork by our propensity for seeing and appreciating the visual creativity of faces and animals – leaving a question to the reader, ‘Could a machine be used to appreciate creative artworks? And would it rank the work based on faces and expressions?’

Chapter 2 presented a number of emotional models from prominent theorists of basic emotions – Ekman and Cordaro; Izard; Levenson; and Panksepp and Watt. Ekman and Cordaro’s model is the most relevant for my thesis question because it seems that some of the emotions in the other models could not be robustly expressed by facial expressions. In particular, a lot of work was performed by Ekman to support his model as being universally recognised. This raises a point to the realisation of my thesis question – if my machine is able to recognise the universal facial expressions then the emotions they produce should be recognised universally.

8.3 Research into the theory behind Support Vector Machines (SVMs) and why they were chosen

After Chapter 2, the remainder of my thesis concentrated on machine learning techniques, with particular focus on Support Vector Machines (SVM), which are my primary learning tool and the content of Chapter 3. The initial section of Chapter 3 focused on the literature regarding general classification tasks, evaluating facial expressions and face detection methods. And it was because of the robust application to facial expressions and face detection that SVM was chosen.

A total of three SVM types (C-SVM, $\nu$-SVM, SVDD) and three kernels (linear, polynomial, Gaussian) were explored and mathematically described using a two-class, one-class and multiclass scenario. The main focus behind the research into the theory of SVMs was to clearly describe the parameters that require tuning, and how the support vectors are used to define the decision boundaries for the different SVM types and kernels, as well as how the number of support vectors varies under different parameter configurations.
8.4 Contribution to multiclass classification using SVMs

The description of the two different frameworks (one-vs-all and one-vs-one) for multiclass classification in Chapter 3 eases into the content of Chapter 4, Pairwise Adaptive Support Vector Machines (pa-SVM), which is one of my main contributions. The chapter began by outlining how parameters are currently selected in popular software to be the same for each binary classifier in a multiclass configuration. Because of this design, it may not arrive at an optimal solution. My hypothesis is that performance can be improved by optimising the parameters for each binary classifier. To compare with other recent methods, a number of datasets from the UCI Machine Learning Repository were chosen for benchmarking, and to maintain a consistent low biased classification accuracy, a standard 10-fold cross validation method was used throughout my experiments.

Overall, the experimental results of Chapter 4 supported my hypothesis because there were no correct classification rates in the standard multiclass SVM method that returned a rate greater than the rates of my proposed method. And, because of the way in which the best parameters for each binary classifier are evaluated, multiple pairs of parameters with the same best rate frequently occur. To evaluate multiple parameters of binary classifiers with the same best correct classification rate, I expanded my experiments to cover four scenarios of min/max $C$ and min/max $\nu$ with min/max $\gamma$ (where $C$ and $\gamma$ are the parameters for the pa-SVM based on $C$-SVM with a polynomial or Gaussian kernel, while $\nu$ and $\gamma$ are the parameters for the pa-SVM based on $\nu$-SVM with a polynomial or Gaussian kernel).

The explored scenarios showed that the total number of support vectors are minimised by selecting the max $C$ min $\gamma$ of $C$-SVM and min $\nu$ min $\gamma$ of $\nu$-SVM when the best classification rate of a binary classifier has more than one parameter pair. However, for some datasets the $\nu$-SVM has regions of infeasible parameters where no solution can be found. This is because there was at least one fold where there were no feasible parameters that satisfy a particular pair of classes during the training stage of a binary classifier. The act of tracking infeasible parameters has been taken into account by my experiments.

My comparison with other recent methods shows that the new method is very competitive. The main concern with the comparison was the inclusion of studies that used a separate training and test set instead of a cross validation
method, and others that used a different cross validation fold altogether. The number of datasets from the very few studies that returned a better result was small, and those with a marginal increase did use a separate training and test set.

8.5 Contribution to facial expression classification

The outcome of Chapter 4 was a proven improvement when compared to the standard multiclass SVM. This refined framework was subsequently explored in the rest of the experiments of my thesis to validate its robust performance, starting with facial expression classification in Chapter 5.

The aim of Chapter 5 was to show that a component approach does perform better than a holistic approach for classifying discrete facial expressions. The content of Chapter 5 began with the literature review showing method disparities among current classification studies of discrete facial expressions, making the process for comparison difficult. Because of these reasons, I delivered the best techniques (using a number of image preprocessing techniques and feature points) to provide a comparative bridge (and suggested a standard) so that results can be compared.

To conceptualise a component approach, I proposed a face model consisting of feature points around the brows, eyes and mouth. The purpose of the face model was to align the datasets of photographed human faces that have various expressions. In particular, the roles for each of the feature points in the face model were justified – this was not clear in other studies. Also, when I analysed the performance of expression pairs, the finding shows that the performance dynamics of paired expressions suggest a non-equidistant expression space, as I observed specific pairs of expressions performing better than others. These observations correspond to psychological studies using real human subjects.

A notable mention from the literature review in Chapter 5 was the lack of certified datasets publicly available, and many studies did not cover the contemptuous expression (as well as the neutral face). During the comparison of the classification rates of my results with the literature, it should be noted that the reported classification rates (of other studies) will be artificially higher if they exclude these expressions; in particular, the reported classification rates will be artificially higher if they use separate training and testing sets or a cross validation fold lower than 10. Another mention from the literature review was the influences of holistic approaches for facial recognition over component approaches for facial expression classification, which perhaps explains the high
8.6 Contribution to holistic face perception

The overall purpose of the study conducted in Chapter 5 was to provide a thorough understanding of the choices that impact a classifier’s performance on the seven universal facial expressions and the performance dynamics between expression pairs. The efforts of my face model are further utilised and tested in Chapter 6 and Chapter 7, where I constructed cartoon faces using the feature points of the face model for my experiments. Starting with Chapter 6, I used the cartoon faces to create a number of minimal face-like patterns (i.e. degraded cartoon faces) as an empirical platform to show the viability of holistic perception of pareidolia faces for describing a scene. I wanted my face detectors to be able to generalise to finding non-human face types (since the current face recognition literature is specific to finding and classifying human faces), and testing upon degraded cartoon faces was one such instance.

To complete my aim, a number of image preprocessing techniques, resolutions and different configurations of classifiers were assessed for detecting and classifying the expressions of faces holistically. From the analysis of my results, I was able to show a set of parameters and classifiers that work best for my cartoon faces. Essentially, I have contributed a framework where other image preprocessing techniques and a different set of test images can be substituted such that the robustness of the face detectors and expression classifiers can be empirically shown using my proposed null test and window search. The outcome is a simple procedure where you can pass in an image of a scene, and by using the set of best parameters and classifiers, a description of this scene is produced as an array of emotional scores corresponding to a set of facial expressions of emotion.

8.7 Contribution to component face perception

Referring back to the questions I asked in the introduction (of Chapter 1), ‘But is that enough? Are holistic based methods good enough for detecting faces for my purpose?’, probably not. This is because when I visualise a house with two windows for the eyes and a door for a mouth, I consider the house to be a face even if I move the windows further away horizontally, or move both windows vertically higher. And because of this possibility, holistic based
8.7. Contribution to component face perception

Methods become less effective – providing the motivation for Chapter 7.

The idea of Chapter 7 is to detect abstract faces where the eyes and mouth are distorted beyond that of a natural human face. The purpose of the study was to simulate the pareidolia capability of humans to produce an emotional response to a scene using expressions of perceived face components. In the literature review I found only a small number of relevant studies that were aimed at detecting faces by face components. Besides the fact that they were all specific to finding human faces, they also have shortcomings that meant they did not fit my requirements.

Due to these reasons I have developed a new method that is capable of detecting many different face types without the need for SVMs, and a particular advantage of this technique when compared to a holistic method is that the face components are explicitly isolated. To achieve this, heuristics were used to define a suitable metric for evaluating a range of candidate faces. And to complement the component face detector I trained several one-class semblance classifiers using scale invariant feature vectors derived from the face components (i.e. eyes and mouth) of the training data to capture the geometry, size and shape. My assessment of the outputs from the semblance classifiers showed that not all combinations of $\nu$ and SVDD SVM types (with the Gaussian and the polynomial kernel) agree with the faces detected; only the $(\nu,\text{poly})$ one-class classifier.

Amongst the many experiments I conducted with the feature vectors, I showed how they are capable of distinguishing three shape faces of two eyes with a mouth from those of two brows with a mouth; however, they are not adequate enough to distinguish expressions. And to create the component expression classifiers (for the experiments) I sliced the dataset of cartoon faces into face components – brows, eyes and mouth. This was possible because of the feature points I manually inserted into photographed human faces, which is part of my face model (from Chapter 5).

Further, I showed that the classification of the mouth expression is almost equivalent to that of the whole face, while the eyes and brows were harder to determine across the seven discrete expressions of emotion. Lastly, the final experiment I conducted in Chapter 7 was an evaluation of the expression classifiers where the contemptuous expression was excluded. This experiment was also conducted at the end of Chapter 6. The reason behind this was to support my finding in Chapter 5, which was that the reported classification rates of other studies will be artificially higher where the contemptuous expression was excluded. And the reason I extend the issues and justify the
use of the contemptuous expression is because I consider it to be universally present across human cultures, and its asymmetric nature is not shared with other expressions.

8.8 What next?

The future work for each of my contributions has been presented in their respective chapters. I iterate these notable endeavours as follows:

- Chapter 4 – pa-SVM: to include a parallel analysis to assess the improvements in speed as a result of this current refinement, as well as the extensibility of this study to a different framework (such as one-vs-all or a weighted strategy), and the generalisability of this classification technique using a different learning classifier.

- Chapter 5 – extending the face model: using the feature points to capture all possible facial expressions; creating a motion model which could be a promising emotion signature of an individual – and if this could be shown to be unique amongst individuals then it is applicable for identification.

- Chapter 6 – other test cases for holistic perception: The use of degraded cartoon faces is one such instance of empirically testing the viability of holistic perception of pareidolia faces for describing a scene. My framework can be substituted with other image preprocessing techniques and a different set of test images such that the robustness of the face detectors and expression classifiers can be empirically shown using the null test and the window search.

- Chapter 7 – several modules in my method have been identified that can be improved upon. They include improving the connected labelling algorithm so that components are better segmented by using information from multiple resolutions of the image; improving merge conditions so that expected face components (based on the integral rectangles) do not fail the matching rules; correcting the size of rectangles so that the boundary fits around expected face components and aligns it properly to the trained images; incorporating colour information to tune local and global perception of face components; and finally, a clustering algorithm for multiple face detection based on face components – as currently multiple face detection in the literature uses colour based segmentation, which may not necessarily be useful for perceiving pareidolia faces.
8.9 Summary

I have presented my thesis question and delivered a machine that is capable of describing the emotional response to a scene using faces and facial expressions. My machine is capable of seeing pareidolia faces, both natural and abstract – where abstract faces are detected using a newly developed component face detector (Chapter 7). My machine is capable of evaluating facial expressions using holistically perceived faces and perceived face components – this is possible because of my face model where the feature points were manually inserted for each photographed human face of the training data (Chapters 6 and 7). My machine has improved classification of facial expressions using a refined parameter selection process for training the facial expression classifiers (Chapter 5), which also works for other real world multiclass datasets (Chapter 4). Further, the math and theory of my primary machine learning tool was described (Chapter 3) and I have shown the relevance of my thesis question using research in the disciplines of Cognitive Science and Psychology (Chapter 2). Finally, all the above was driven by the motivation behind my thesis question (Chapter 1).
Bibliography


[73] Daniela Godoy. Comparing one-class classification algorithms for finding interesting resources in social bookmarking systems. In *Proceedings of the Third international conference on Resource Discovery*, RED’10,
Bibliography


Kenny Hong, Stephan K. Chalup, and Robert A.R. King. A component based approach for classifying the seven universal facial expressions of emotion. In IEEE Symposium on Computational Intelligence for Creativity and Affective Computing (CICAC), 2013. Manuscript accepted for publication. (Cited on pages 67, 106 and 139.)


[118] Jiye G. Kim and Irving Biederman. Greater sensitivity to nonaccidental than metric changes in the relations between simple shapes in the


[184] Luis Rueda, B. John Oommen, and Claudio Henríquez. Multi-class pairwise linear dimensionality reduction using heteroscedastic schemes. Pat-


