Real-Time Recognition of Facial Expressions for Affective Computing Applications

by

Christopher Wang

A M. Eng. Project submitted in conformity with the requirements for the degree of Master of Engineering

Department of Mechanical and Industrial Engineering
University of Toronto

© Copyright by Christopher Wang 2010
Real-Time Recognition of Facial Expressions for Affective Computing Applications

Christopher Wang

Master of Engineering

Department of Mechanical and Industrial Engineering

University of Toronto

2010

Abstract

In this paper, a real-time facial expression recognition system is proposed to classify images of human facial expressions into a two dimensional model of emotion. The system is comprised of a facial feature detection system that uses constrained local models to locate the features, and a facial expression recognition system that utilizes multi-class support vector machines to classify the facial expressions. The outputs of the system are values of pleasure and arousal associated with an input image. Classifications rates achieved were 76% for the pleasure dimension and 62% for the arousal dimension. An analysis on facial action parameters also revealed that some parameters were more effective at determining pleasure and arousal values than others.
Acknowledgments

I am thankful to my supervisors, Goldie Nejat and Beno Benhabib, whose guidance and support throughout the year made the completion of this project possible.

I would also like to thank the members of the Computer Integrated Manufacturing Lab and Autonomous Systems and Biomechatronics Lab who assisted me in performing experiments for this project.
# Table of Contents

## Contents

Acknowledgments .................................................................................................................. iii  
Table of Contents .................................................................................................................. iv  
List of Tables ........................................................................................................................ vii  
List of Figures ........................................................................................................................ viii  
Chapter 1 ............................................................................................................................... 1  
1 Introduction ......................................................................................................................... 1  
1.1 Motivation ....................................................................................................................... 1  
1.2 Challenges ....................................................................................................................... 1  
1.3 Research Objective ......................................................................................................... 2  
1.4 Report Overview ............................................................................................................. 3  
Chapter 2 ............................................................................................................................... 4  
2 Literature Review ............................................................................................................... 4  
2.1 Feature Detection .......................................................................................................... 4  
2.1.1 Two Dimensional Techniques ................................................................................... 4  
2.1.2 Three Dimensional Techniques ................................................................................ 9  
2.2 Available Databases ...................................................................................................... 11  
2.2.1 BU-3DFE [33] ........................................................................................................ 11  
2.2.2 JAFFE ...................................................................................................................... 11  
2.2.3 FACS ....................................................................................................................... 12  
2.2.4 Cohn-Kanade .......................................................................................................... 12  
2.2.5 Yale .......................................................................................................................... 12  
2.2.6 AR ............................................................................................................................ 12
5.1 Facial Feature Training ........................................................................................................... 34
  5.1.1 Obtaining Images For Training.......................................................................................... 34
  5.1.2 Finding the True Mean....................................................................................................... 35
  5.1.3 Principal Components Analysis (PCA) ............................................................................. 36
  5.1.4 SVMs for Feature Detection ............................................................................................. 37
  5.1.5 Logistic Regression............................................................................................................ 38
  5.1.6 Constrained Local Model Fitting ...................................................................................... 40
  5.1.7 Convergence Results......................................................................................................... 42
  5.1.8 Iteration Duration.............................................................................................................. 43

5.2 Facial Expression Recognition ............................................................................................... 44
  5.2.1 Two Dimensional Model of Emotion .............................................................................. 44
  5.2.2 Experimental Method....................................................................................................... 44
  5.2.3 Inter-Coder Reliability ..................................................................................................... 45
  5.2.4 Facial Action Parameters (FAPs) .................................................................................... 47
  5.2.5 Emotion Classification..................................................................................................... 48
  5.2.6 Emotion Classification Discussion .................................................................................. 55
  5.2.7 Summary of Experiments and Results ............................................................................ 56

Chapter 6 ....................................................................................................................................... 58

6 Conclusion.................................................................................................................................... 58

Chapter 7 ....................................................................................................................................... 59

7 Bibliography............................................................................................................................... 59
List of Tables

Table 1: Inter-coder reliability $\alpha$ values for trial 1 and trial 2 for pleasure and arousal. .......... 45

Table 2: Comparison of inter-coder reliability $\alpha$ for rated data for pleasure and arousal with 5 and 4 coders .................................................................................. 46

Table 3: Inter-coder reliability $\alpha$ for ranked data for pleasure and arousal ......................... 46

Table 4: Facial Action Parameter Definitions ............................................................................. 48

Table 5: 7-class SVM Results for rated data in pleasure dimension with slack = 1 .................. 50

Table 6: 7-class SVM Results for rated data in arousal dimension with slack = 1................. 50

Table 7: 3-class SVM results for rated data in pleasure dimension with slack = 0 ............... 51

Table 8: 3-class SVM results for rated data in arousal dimension with slack = 0................. 51

Table 9: Binary SVM for ranked data in pleasure dimension with slack = 0 ......................... 51

Table 10: Binary SVM for ranked data in arousal dimension with slack = 0......................... 52

Table 11: Top Results for linear, quadratic and cubic kernel in the pleasure dimension ..... 53

Table 12: Top Results for linear, quadratic and cubic kernel in the arousal dimension .......... 53

Table 13: Correlation between FAPs and average classification accuracy for linear, quadratic and cubic kernels sorted in ascending order in the pleasure dimension ........................................ 54

Table 14: Correlation between FAPs and average classification accuracy for linear, quadratic and cubic kernels sorted in ascending order in the arousal dimension .......................... 54
List of Figures

Figure 2-1: Russell's Two Dimensional Model of Emotion [7]........................................................................ 14

Figure 2-2: a) 28 words on the pleasure-arousal emotional model based on word comparisons and b) 28 words on the pleasure-arousal emotional model based on self reported data [38]....... 15

Figure 2-3: 83 expressive words plotted on the pleasure-arousal model of emotion [39]........... 16

Figure 2-4: The six basic emotions in the pleasure-arousal emotional model [40]........................ 17

Figure 3-1: Facial Feature Detection Flow Chart ...................................................................................... 22

Figure 3-2: The locations of the 68 feature points .................................................................................... 23

Figure 3-3 The 68 facial feature point on a sample image (image courtesy of JAFFE database) [34].................................................................................................................................................. 23

Figure 3-4: Graphical depiction of hyperplane and sample points [61] .............................................. 26

Figure 4-1: Flowchart for proposed facial expression recognition system.................................. 31

Figure 4-2: Two dimensional model of emotion (Pleasure-Arousal) .................................................. 32

Figure 5-1: Feature points of unaligned training images ................................................................. 35

Figure 5-2: a) Average location of feature points and b) feature points of training images aligned to mean.................................................................................................................................................. 36

Figure 5-3: Variations of the shape model for a) first eigenvector b) second eigenvector c) third eigenvector, from -60λ to +60λ where λ is the corresponding eigenvalue................................. 37

Figure 5-4 a) SVM response and location of maximum response and b) corresponding point in actual image and c) Actual feature point location. ................................................................. 38

Figure 5-5: Logistic Regression of Feature 16 ....................................................................................... 39

Figure 5-6: a) Simulating a face and nose tracker, and b) superimposing an initial guess.............. 40
Figure 5-7: CLM fitting for the $i^{th}$ iteration and true value of feature point locations of a trained image, a) $i = 0$ b) $i = 1$ c) $i = 5$ d) $i = 10$ and e) true values. .......................................................... 41

Figure 5-8: CLM fitting for a non-trained image, a) $i = 0$ b) $i = 1$ and c) $i = 5$ ................................. 41

Figure 5-9: Convergence data for a) change in feature vector, and b) change in parameter vector corresponding to CLM fitting for Figure 5-7. ............................................................................ 43

Figure 5-10: Convergence data for a) change in feature vector, and b) change in parameter vector corresponding to CLM fitting for Figure 5-8. ............................................................................ 43

Figure 5-11: Two dimensional model of emotion for pleasure-arousal [7]................................. 44

Figure 5-12: Feature point locations ............................................................................................................ 47

Figure 5-13: a) histogram of rated data before re-categorization, and b) histogram of rated data after re-categorization ............................................................................................................. 49

Figure 5-14: Rated data re-categorized for 3-lass SVM ............................................................................. 50

Figure 5-15: System Overview .................................................................................................................. 57
Chapter 1

1 Introduction

1.1 Motivation

Human affect is the emotional and internal states of an individual [1], and can manifest itself in a variety of ways including facial expressions, body movements, gestures, voice behaviours and other physiological signals like heart rate and sweat [2]. A related research area that focuses on determining human affect is affective computing, which attempts to provide machines (e.g. computers, robots) with the ability to observe, interpret and generate affect features, specifically emotional states, of humans [3].

This work focuses on incorporating affective computing in human-robot interaction (HRI) scenarios with robots. By understanding the different emotional cues of humans, robots can more effectively respond and react to real-life situations. For example, a robot can be able to determine when an action it is performing is causing a negative or positive response a human user. Alternatively, a robot can sense that the user may be feeling anxious or fearful in a dangerous situation, and take the necessary precautionary measures without having to be explicitly told.

1.2 Challenges

This work focuses on the ability to recognize one particular affect features, namely facial expressions via camera, in order to provide a robot with the ability to perceive the emotional states of human users. The problem of emotion recognition through facial expressions has a number of challenges associated with it.

1. Firstly, it is necessary to have an appropriate classification scheme in which to classify images of facial expressions. Many studies use discrete facial expression categories [4,5,6] for their classification schemes, but this has some inherent problems. Using this type of classification scheme restricts the system to classifying only the selected classes (e.g. Anger, Sad, Happy, etc) and will be
unable to classify facial expressions that are not included in the classifier. Additionally, facial expressions that occur as a combination of multiple emotions, or transition expressions will be classified incorrectly as one of the selected classes.

2. Secondly, a facial expression recognition system must be able to accurately detect the communication feature it is using for classification. In a real environment, it is impractical for a system to be trained from images of a single individual, so the collection of images used for training of the feature detectors must include many different face shapes, sizes, and ethnicities. This will allow for the system to better cope with the large range of faces it is likely to see in its operating life.

3. Thirdly, similar to the facial feature detection, the emotion classification system must also work for a large number of people, and its accuracy cannot be restricted to a certain individual or small group of people. It must also work for real life situations. Rather than having a system that works for posed, exaggerated facial expressions, which may not manifest themselves in everyday interactions, it is necessary for the system to be designed for real facial expressions that occur in real life. Most of these facial expressions are quite subtle, and a major challenge is to create a system that works for these subtle facial expressions.

4. Finally, a functioning emotion recognition system for HRI needs to work in real time. HRI occur in real time, so it is impractical for an HRI system to be computationally complex in order to analyze a facial expression for emotion. The user's facial expression may have changed by the time the result is computed and the resulting delay will cause the robot to respond with a lag and decrease the quality of the interaction between the user and robot.

1.3 Research Objective

To address these concerns, a system that performs emotion recognition in real time is proposed in this work. The system is designed to map a human user's facial expressions into a two dimensional model of emotion. The two dimensional model of emotion categorizes emotional states into the two metrics of valence and arousal [7]. The dimension of valence represents how positive or negative the individual is feeling, while the dimension of arousal
represents the level of energy the individual is experiencing. For example, high valence and high arousal values on this scale could represent excitement, while low valence and low energy values could represent boredom. By using this scale of measurement, one can determine whether an individual is going from low arousal or valence to high arousal or valence and vice versa. With this information, it will be possible for a robot to associate actions with emotional responses, regardless of what the end state is. For example, if an action causes a user to go from very sad to a less sad emotional states, a system that uses this classification scheme will be able to capture this, while a discrete emotion recognition system will simply see that the individual is still sad, and not be able to draw any conclusions on the results of its actions. A database has been created for this work consisting of facial expression images of naturally occurring facial expressions to train the system effectively for real world HRI. For the system to work in real time, constrained local models will be used to detect the features on the face, with support vector machines to classify the emotional states.

1.4 Report Overview

The remainder of the thesis consists of the following: Section 2 consists of a literature review on feature extraction, emotion classification, facial expression recognition methods, and existing facial expression databases; Section 3 presents the theory behind the constrained local models used for feature detection; Section 4 presents the theory behind the emotion classification system; Section 5 discusses the experimental procedure and their results; Section 6 provides concluding remarks and plans for future work.
Chapter 2

2 Literature Review

2.1 Feature Detection

When detecting facial expressions from images taken of human faces, it is first necessary to determine what information to extract before classification can be performed. The following section is a literature review of the various techniques employed for various feature extraction techniques. The methods outlined encompass the different metrics derived from an image in facial expression recognition, and how they are detected and acquired.

There are three choices to make when selecting techniques to acquire features for facial expression recognition: feature selection, input modality and input dimension. A feature in this context is the input that the facial expression recognition system uses to classify the emotion. This feature can be considered either template based or feature point based. Template based methods take the image as a whole as a feature and process the image data accordingly, while feature point based methods specify particular points on the image and define them in terms of different metrics. The choices for input modality are between static images and image sequences. Static methods extract information from a single image, while image sequence methods try to exploit the information found in the differences or changes between subsequent images. Finally, the dimension of the input can be either two dimensional or three dimensional. Two dimensional methods utilize perspective projection information, while three dimensional methods aim to take advantage of the extra information provided in higher dimensional views.

2.1.1 Two Dimensional Techniques

2.1.1.1 Template Based Methods

Local binary patterns (LBPs) [8] are used in 2D template based methods and applied to static images. To obtain an LBP value for a particular pixel, the surrounding neighborhood of pixels is thresholded to the value of the center pixel and output to a binary number. It is usually done in 3x3 neighbourhoods to obtain an 8 bit value for each pixel, but can be extended to larger
circular neighbourhoods to provide larger scale information, since 3x3 neighbourhoods are somewhat limited in the information they provide. The output values of the LBP operator can yield information about micro patterns, identifying features like edges, spots, flat areas, etc. Also, uniform patterns comprise 90% of all patterns in the 3x3 neighbourhoods, so features of interest can be limited to those that have more than two bitwise transitions. These values are outputted to a 256 bin histogram and used as metrics for comparison and classification. The image can be divided into smaller regions to incorporate spatial information into the histograms as well. LBPs are gray scale invariant, work well with low resolution images and are extremely simple to compute. However, LBPs may not be able to handle occlusion and head pose variations. In order to use LBPs to compare images, it is necessary to have the images to be normalized to the same sizes. In [8], Shan et al. used LBPs in conjunction with support vector machines (SVMs), linear discriminant analysis (LDA), and linear programming methods to classify facial expressions.

Spatially Maximum Occurrence Models (SMOM) [9] is a 2D template based technique used on static images that is based on the statistical properties of a training set, specifically on the probability that a pixel location will be a certain intensity value. For each pixel position, a histogram is generated with N bins, the number of intensity levels. The peak values are ranked and form a set of k images, where the k\textsuperscript{th} image contains the value of the k\textsuperscript{th} peak of each pixel histogram. The 1\textsuperscript{st} image will contain an image with pixel intensities which occur the most often in the training set for that particular expression. It uses the intensity distribution at each pixel for each expression, but does not provide any information about spatial relationships between neighbouring pixels.

Elastic Shape Texture Matching (ESTM) [9] is a 2D appearance based technique applied to static images, that measures the similarity between images based on shape with an edge map, and texture with Gabor and angle maps. The edge map contain locations of edge, Gabor maps are determined by concatenating magnitudes of Gabor wavelet representations at different frequencies and wavelengths applied to the edge map, and the angle map contains the direction of each edge. Gabor wavelets are gradient filters of different frequencies and directions. A collection of Gabor filters are usually applied to images in order to extract the different edges and gradients existing in an image, and a widely used technique in facial image processing.
However, it is extremely computationally expensive to run a large number of filters at different frequencies and orientations.

Principal component analysis (PCA) [10,11,12,13] is a popular mathematical tool that reduces the dimensions of a problem by subtracting the average face from a collection of images and breaking down the dissimilarities into different independent components. When applied to the intensity values of images, these independent components can be determined and used for facial expression recognition.

2.1.1.2 Feature Based Methods and Tracking

Active Appearance Models (AAMs) [14,15] are parametric models that describe the shape and appearance variations of an image. A 2D AAM is a triangular 2D mesh that is initially overlaid on an image with its vertices located at specific feature points, which may be manually or automatically selected, usually on an initialization frame in which subsequent fitting attempts will rely on. Before AAMs can be fitted, a set of training images must be used to train shape models using PCA to obtain the average and basis vectors for the vertices locations. After the shape models are obtained, each image is warped to the mean shape with an affine warp between the produced triangles, creating a normalized image shape. Again, PCA is performed, yielding a mean appearance and a set of appearance basis vectors. The goal of AAM fitting is to minimize the difference between the warped images and appearance images, in essence determining the locations of the vertices based on the appearance inside each triangle. AAMs can represent any face image that contains variations found in the training sample, but cannot express faces that are not present. Computational complexity is based on the initial guess as well as the convergence algorithm used. In order to utilize AAMs for classification of facial expressions, it is necessary to have a neutral expression defined in order to achieve person independence, as different people have different expressions for happiness/neutrality. Cheon et al. [14] used Differential AAM Features (DAFs) along with the above formulations to exploit the assumption that differences from neutral expressions to a specific expression are similar among different people. The DAF vector is the difference between the current image and reference neutral image of the shape, appearance and pose vectors. It uses pre-computed probability distributions to determine the DAF vector values for specific facial expressions. Jaeckel et al. [16] used AAMs to track facial landmarks at 25 fps in a 60s video to map an individual’s facial motions to a robotic face. This
method had only been trained for a single individual for a specific video, and is not usable in an unstructured environment. The facial landmarks are labeled manually for the training sequences and the most extreme facial movements are used to train the algorithm. Park et al. [17] used AAMs to track 70 feature points, using the inner points of both eyes and middle point of both nostrils for image alignment, and 27 dynamic points to estimate motion: 4 around each eye, 3 around the nostrils, 2 on inside of lip center, 8 around mouth contour and 3 around each eyebrow. The method uses dynamic points to warp the image and cause a subtle facial expression to become an exaggerated one, which is easier to classify. The advantage of motion magnification [17] is that it makes important facial features easier to extract and more evident.

Wang et al. used a direct combined model [18] as an adapted feature point location estimation method to automatically locate feature points on a frontal view facial image. Their method tracks 5 rigid points (tip of the nose and two corners of each eye) that are affected by rigid head motion only and 16 non rigid points (around the eyebrows and the mouth) that are affected by both rigid head motion. The 21 feature points are warped from the 2d facial image coordinates to 2D virtual coordinates using affine transformations. Lucas Kanade Optical flow [18] is then used to track the trajectories of each of these points in a 13x13 pixel flow window. Using the fact that some features are subjected to only rigid head motion, 3D head rotation can be estimated, along with the other features independent of the rigid head motion. However, inaccurate tracking results occur in textureless regions of the face or at the boundary of the mouth.

A Multichannel Gradient Model (MCGM) [5] is an optical flow method that is based on a model that operates in two spatial dimensions and one temporal one, allowing it to recover the velocity field of the image at all locations in image sequences. After applying a range of spatial and temporal filters to an image, ratios can be used to recover the speed and direction of each pixel. The MCGM is robust to changes in scene luminance when calculating speed, and was implemented at 18fps by Anderson et al. using a hardware solution. It is able to effectively operate in cluttered and dynamic scenes. As a metric for facial expression classification, different regions of the face can be empirically or automatically determined, and test image/training images averaged the motion generated by the MCGM to reduce the amount of information entered into the classifiers. Ratios of average motion were taken to remove the effects of rigid head motion in the facial expression action. This method used SVM to act as a classifier and ran
at 4fps. The system is restricted by the speed at which it operates, as well as the averaging of data which results in a loss of information.

Active Shape Models (ASM) [19] is a feature based approach that consists of a collection of points located on the face that are tracked throughout image sequences. Because the method is tracking individual points, the location of these points can be subject to poor performance if the images are noisy. Song et al. [19] used a GPU based ASM that uses hardware to detect and enhance the edges in the images, which are normally computationally expensive), as well as perform tone mapping and model matching procedures. These enhanced images also cause the tracking algorithms to converge more quickly. The group uses 128 feature points to define the face configuration, 56 of which are used to compute Facial Action Parameters (FAPs). FAPs are metrics that define the shape and orientations of the facial features in terms of lengths and angles. For example, the distance between the upper and lower eyelid, eye separation distance, eye-nose separation, mouth-nose separation, and mouth width distance. The classification method used in this study is a Triple Hidden Markov Model. The hardware also increases the computational speed from 24fps to 48 fps

The method of Eigen points [20] searches for image features that are associated with a group of control points and estimates a detailed spatial distribution of control points around that feature. Training images for this method include feature images and x-y locations of these control points. A coupled manifold is formed by performing SVD decomposition on a matrix containing image data and control point locations. To locate a feature (lips, eyes, etc), a template or model based matcher can be used, which defines the sub image and image plane used to locate the control points. To place the control points, the feature sub image is projected onto the couple manifold model, which is then projected onto the control point subspace to give locations for the control points. This method is non iterative, accounts for noise, and attempts to retrieve shape from appearance. However, it is dependent on example data, and a large set of training images is required for it to run effectively.

Constrained Local Models (CLM) [21] are 2D static methods that merge shape and texture information by coupling the local patch of feature detectors at a global shape level (i.e. a set of feature detectors in a particular orientation). It has been proven to work more effectively than AAMs because it is more robust to occlusion and appearance changes, as well as the lack of
need for texture warps. The Bayesian Constrained Local Model shows shape information in a prior distribution, that models all the faces the face detector can detect, and texture information is stored in log likelihood functions. The algorithm is iterated to shift probability distributions over the face, removing false responses at each iteration to more accurately find the feature points. Ryan et al. [22] used CLMs with an Exhaustive Local Search (ELS) algorithm for fitting that operated at 35 fps. In [23], convex quadratic fitting was utilized to update the CLMs.

The Facial Action Coding System (FACS [24]) is a set of action units (AUs) that describe the different movements that can occur on a human face during facial expressions. There are 46 action units in total that range from the lifting of eyebrows, to tightening of lips, or dropping of the jaw, etc. AUs are not independent of each other, and various overlaps can occur, leading to some problems in classification algorithms.

2.1.2 Three Dimensional Techniques

Three dimensional models inherently provide more information than 2D models due to the presence of depth information, and are more robust than 2D models. Many 3D model extraction solutions are subject to expensive computational complexity, or over simplified models that do not accurately represent the object.

The acquisition of 3D data can also produce image artifacts that may affect the rendered model [25]. The camera can receive light at intensities that saturate the detector or receive light levels too low to produce high quality images. This can occur in areas where there is specular reflection in stereo systems. Stereo based systems also have trouble getting true dense sampling of the face surface, and spare sampling points in regions where there is too much natural texture, leading to the exclusion of certain features (too smooth). Multimodal analysis with 3D and 2D data may be able to provide better data for classification (of face recognition) than single modalities, but compared to multiple 2D images (without 3D rendering), it does not show significant improvement, leading to a possible optimization problem in determining the best ways to use the acquired data [25]

A process completed by Chaumont et al. [26] breaks this problem into two steps, which first formulates an estimation of the 3D model, followed by model refinement. In the estimation section, a CANDIDE wireframe model (3D wireframe of an average face) is projected onto the
2D space from the 3D space under the assumption that all feature points are coplanar. This approximation is realistic because the differences in depth between features are very small compared to the distance to the camera. Making this assumption results in a projection of a 2D image on a 2D plane, which is a problem much easier solved. Also, since few 2D-3D correspondence points are available for use, the matrix is very sparse, and can be solved very quickly. After this approximation is determined, the wireframe is refined by perturbing the 3D points separately to match with the 2D points. This method is a fast method for face tracking and 3D face model extraction, can predict feature positions due to rotations and translations and model recovery in the presence of occultation because 3D information is known about the object.

Soyel et al. [27] used 3D distance vectors to obtain 3D FAPs between feature points to measure quantities like openness of eyes, height of eyebrows, openness of mouth, etc. to obtain distance vectors for test and training data for different expressions. They use only 23 facial features that are associated with the selected measurements and classify with a neural network. Tang et al. [28] utilizes the same approach, but performs an algorithm on the set of distances between the 83 points to determine the measurements that contain the most variation and are the most discriminatory, allowing for better recognition than empirically determined measurements.

Shape information is located in geometric features like ridges, ravines, peaks, pits, saddles, etc. local surface fitting is done, by centering the coordinate system at the vertex of interest (for ease of computation). The patch can expressed in local coordinates and a cubic approximation \((x^3, x^2y, xy^2, etc)\) can be used to fit the surface locally, yielding two principle vectors that describe the maximum and minimum curvature at that point, and two corresponding eigenvalues. Along with the normal direction at that point, the surface properties can be classified into labels (flat, peak, ridge, etc) and a Primitive Surface Feature Distribution (PSFD) [29] can be generated as feature.

Other methods attempt to fit surface models onto point clouds of 3D sensor data. Mpiperis et al. [30,31] used a neutral face with an average identity and deformed it to the appropriate expression/identity. A triangular 3D mesh is placed on the face and subdivided into sub-triangles to increase the density. First a set of landmarks is associated with vertices on the mesh, which remain unchanged during the fitting process. Fitting is done as an energy minimization problem that consists of terms describing opposing forces between the landmarks.
and mesh points, the distance between the surface and the mesh, and a smoothness constraint, which is solved by setting partial derivatives to 0 and solved using SVD. Asymmetric Bilinear models are used for facial expression recognition in which models identity in one dimension and expression in another. 3D Facial shapes obtained through finding the difference between neutral and expressive faces in 3D can also be used to classify facial expressions [32].

Venkatesh et al. employed principal component analysis on 3D mesh datasets to attempt to classify facial expressions [10]. PCA is a popular mathematical technique that allows the dimensions of the problem to be reduced, making it easier to solve. For the training set, 68 feature points, which have been known to effectively represent facial expressions, have been manually selected around the eyes, mouth and eyebrows. PCA is done on the x, y, and z locations of these feature points to determine eigenvalues that can be used to find matrix projections on a given matrix A. This method automatically extracts features after they are divided into bounding boxes using anthropomorphic properties. This method achieves the automatic selection of points; however it is very computationally expensive.

2.2 Available Databases

2.2.1 BU-3DFE [33]

The BU-3DFE database consists of 2500 data sets of feature points in 3D space and 2500 data sets of frontal texture data from 100 subjects in which 60% were female, 40% were male, with varying ethnic ancestries. Each subject performed seven expressions (neutral, happiness, disgust, fear, anger, surprise and sadness) on cue.

2.2.2 JAFFE

The JAFFE [34] database consists of 213 images of 10 Japanese females displaying seven expressions multiple times (neutral, sad, surprise, happy, angry, disgust and fear).
2.2.3 FACS

The FACS [24] database contains frontal images of men and woman activating singular or multiple AUs. It is primarily used for teaching individuals how to recognize and score the action units that produce changes in facial appearance.

2.2.4 Cohn-Kanade

The Cohn-Kanade [35] database, also known as the CMU-Pittsburgh AU-Coded Facial Expression database consists of 1917 images of 182 adults with ages between 18 and 50 years. 69% are female, 31% are male, 81% are Euro American, 13% are Afro-American and 6% belong to other ethnicities. It includes frontal views and 30 degree views. Facial expressions displayed include expressions where single or multiple action units are activated.

2.2.5 Yale

The Yale [9] database consists of 15 people, 14 male and 1 female, displaying five expressions: neutral, smile, surprise, blink and grimace.

2.2.6 AR

The AR [9] database is made up of 121 people, 70 males and 51 females, with images for three facial expressions: neutral, smile and scream.

2.2.7 Korean Expression

The Korean Expression [11] database consists images of six males and females displaying facial expressions for 83 expressive words. The images are divided into the pleasure-arousal dimension.

2.2.8 Self built

Some facial expression techniques are tested on databases created by the researchers themselves. They will be referred to as self built databases from herein.
2.3 Emotional Models

There are two main schools of thought when discussing the classification of models. On the one hand, there are discrete models that classify emotions into mutually exclusive groups, and distinguish themselves from each other in important ways [36]. On the other hand, there are those who favour the continuous models of emotion, in which emotions can be located at points on a two or three dimensional plot [7].

2.3.1 Discrete Models

2.3.1.1 Ekman’s Basic Emotions

In 1984, Ekman first postulated a set of six basic emotions that included fear, happiness, anger, sadness, disgust and surprise as universal to all humans [36]. As a result, a large amount of research has been conducted on facial expression recognition for these six discrete emotions [6,5,4]. Although Ekman has expanded these original six emotions to include nine more emotions, most of the literature focuses on the classification of facial expressions in these six categories. He documents that basic emotions differ from each other and other affective phenomena by having the following characteristics:

1. Distinctive Universal Signals
2. Distinctive Physiology
3. Automatic appraisal, tuned to
4. Distinctive Universals in Antecedent Events
5. Distinctive Appearance developmentally
6. Presence in other primates
7. Quick onset
8. Brief duration
9. Unbidden occurrence
10. Distinctive thoughts, memories images
11. Distinctive subjective experience
2.3.1.2 Emotions from Facial Expression Recognition

Much of the literature on facial expression recognition focuses mainly on the six basic emotions identified by Ekman [36], due to the distinguishing factors in facial expressions between the basic six. With more facial expressions, classifications become more difficult, and any similarities between classes would yield to increased difficulties in classification.

2.3.2 Continuous Emotion Models

2.3.2.1 Pleasure-Arousal Circumplex Model of Affect

The second school of thought treats emotions not as if they are discrete, but rather occur in a multi dimensional space and as ambiguous and overlapping experiences. Emotions, herein, are defined in two dimensional models using the following categories: positive and negative affect; tension and energy, approach and withdrawal, or valence and arousal [37].

In 1980, Russell introduced the circumplex model of affect [7]. In this model of emotion, he suggests that rather than dimensions like displeasure, distress, depression, excitement, etc being independent factors as his predecessors had imagined, they are actually dependent and interrelated in a systematic fashion, and can be represented in a spatial model as seen in Figure 2-1

![Figure 2-1: Russell's Two Dimensional Model of Emotion](image)

**Figure 2-1:** Russell's Two Dimensional Model of Emotion [7].
The eight terms can be considered as labels for a fuzzy set, in which there is a gradual transition from membership to non-membership [7]. By plotting 28 affective words such as anger, sad, content, excited, etc with experimental data on a two dimensional model of pleasure and arousal, Figure 2-2a was created by Russell [38]. It shows that antonyms fall approximately 180 degrees apart (happy and miserable), horizontal and vertical axes of pleasure and arousal are easily interpretable by individuals and emotional words are spread out along the circumplex.

A similar study performed by Russell involved investigating the structure of self-reported affect, as it is experienced internally [38] rather than using the aforementioned simple subjective categorization of affective terms. The experiment consisted of individuals responded to affect scales of pleasure-displeasure, followed by rating how accurately 518 adjectives described how they felt that day from a scale of 1 (extremely inaccurate) to 8 (extremely accurate). The data was analyzed and yielded Figure 2-2b. Compared to Figure 2-2a, they are quite similar, though the self report data has different clusters than the categorized data. Self report data clusters occur mainly because those affective terms are likely to occur at the same time. When an individual is depressed, they may also feel gloomy, sad or miserable. The categorized data clusters reflect more the conceptual overlapping in meaning of the affective terms, but nonetheless, both models show that pleasure and arousal are effective dimensions for modeling emotion.

![Diagram](image)

**Figure 2-2:** a) 28 words on the pleasure-arousal emotional model based on word comparisons and b) 28 words on the pleasure-arousal emotional model based on self reported data [38].
The pleasure-arousal model of emotion also yields different point locations of affective terms for different cultures, but overall, provides similar results that cause the terms to lie on a circumplex [38]. The interpretation of facial expressions lying in this emotional space also differs between cultures, but a study of ten facial expressions plotted in the pleasure-arousal dimensional model from different cultures yielded high redundancy in its results. [38]

2.3.2.2 Emotional Models Determined From Facial Expressions

In facial expression recognition, some methods have attempted to use the continuous model of pleasure and arousal to represent emotion. For example, in [39, 11], 83 emotional words were displayed by people through facial expressions and subsequently rated on the two-dimensional model of emotion, yielding a model as seen in Figure 2-3. In [40], Figure 2-4 was used to map surprise, joy, fear, anger, disgust and sadness to three different quadrants.

![Figure 2-3: 83 expressive words plotted on the pleasure-arousal model of emotion [39]](image-url)
Figure 2-4: The six basic emotions in the pleasure-arousal emotional model [40].

2.3.2.3 Higher Dimensions

Alternative to the 2D model, studies have been done on higher dimensional models that include factors such as potency, dominance, aggressiveness, affiliativeness and locus of causation [41]. Though higher dimensional models would be able to give more information relating to a specific emotion, these are more concerned with the reaction of an individual rather than an emotion. Also, they are much more difficult to conceptualize [41].

2.3.3 Comparison

The strength in Ekman’s model of discrete emotions lies in its ability to distinguish one emotion clearly from the others by highlighting key differences. However, it cannot accurately model any emotions that are not included in the defined categories and has trouble dealing with the experiential components of emotion [37].

The strength in the circumplex model is the flexibility in which it can effectively model the transitions between different emotions and situations when emotions are blended together, because of its continuity and treatment of emotions as classes with fuzzy boundaries [38].
2.4 Feature Expression Recognition

A majority of the classification methods are very similar, many relying on the use of support vector machines, [4,42,43,44,45,46] with the main differences being the features that are being used (Gabor wavelet response, pixel intensities, FAPs, etc). Some features perform better with different classifiers, and some perform worse, so it can be difficult at times to draw conclusions about new combinations of classifiers and feature sets. Nonetheless, a review of different techniques used to classify facial expressions is provided.

2.4.1 2D Methods for Recognition of Discrete Emotions

In [42], feature point locations detected in a test image and a neutral image were compared using the JAFFE and Yale databases. Support vector machines were used on global information and local Gabor wavelet features to classify facial expression images into six discrete categories of emotions (happy, sad, surprise, fear, anger disgust). In [4], SVMs with the radial basis function Gaussian kernel on geometric features and k-Nearest Neighbour (KNN) methods, which classifies an example based on the majority vote of the k-nearest neighbours, on flow features to classify facial expressions into five discrete facial expressions (happy, sad, surprise, anger, disgust) using a self built database consisting of 4500 images from ten subjects. In [43], SVM classifiers were used on pixel pattern based texture features to classify images into six discrete emotions using the Cohn-Kanade database. In [44], SVMs were used to classify facial action units on a Candide grid into six discrete emotions (happy, sad, surprise, fear, anger disgust) using the Cohn-Kanade database. Here, a polynomial kernel function is used for the SVMs. In [5], SVMs were used to classify facial velocity information into one of six basic emotions or as a non-expressive face, at a rate of 81/82% using he CMU AU coded facial expression database and at a speed of 4 fps. In [47], fifteen two-class SVM classifiers were used on local binary pattern features to classify near infrared images into the six basic facial expressions at an accuracy of 79% using a facial expression database comprised of two to six posed expressions from 50 people.

Hidden Markov Models (HMMs) in [48] to achieve a 75% recognition rate at 15 frames per second for four discrete emotions (surprise, happiness, sadness, anger on eyebrow, eye, pupil, nostril and mouth feature point data. A two step classification scheme with linear
classifiers and discrete HMMs was used in [49] to classify image sequences into the six universal facial expressions at a recognition rate of 90.8% using the Cohn Kanade database. In [50], linear discriminant analysis (LDA) on a self built database of thermal images to achieve real time, lighting, occlusion independent recognition rates of 66.3% for five facial expressions (normal, happy, sad, disgust, fear). In [51], a dynamic clustering method in conjunction with PCA for classification of facial expression images to three facial expressions (unhappy, neutral, happy) at an accuracy of 95.1% on the Frey facial expression database. In [52], the analytical hierarchy process to classify facial expression images from the JAFFE database into five facial expressions (neutral, happy, anger, sad, surprise) at an accuracy rate of 85.2%. A similarity measurement based on elastic shape texture matching (ESTM) using the spatially maximum occurrence model as an input feature in [9]. The recognition rates on the AR database and Yale database for this method were 94.5 and 94.7% respectively in 0.10 s and 0.17 s of computation time. The categories of recognition were neutral, smile, surprise, blink, grimace, and scream. In [53], a double layer GDA classifier on 2D + 3D AAMs in conjunction with KNN method to classify 2D images into four emotions (neutral, happy, surprise, angry) using a self constructed database containing 1200 images of 20 persons annotated with expression and pose data. The first level of GDAs separates the data a shape and an appearance vector, and the second level of GDAs separate the data into their own features spaces, so that they are grouped together with their class labels, allowing for a KNN approach to be used. In [54], spatio-temporal motion energy templates of the whole face were used for five expressions (smile, surprise, anger, disgust, and raise eyebrow) to achieve a recognition rate of 98% on a self generated 2D facial expression database. In [55], Naïve Bayesian classifiers were used to recognize six expressions (anger, disgust, fear, happiness, sadness, and surprise) from images in the Cohn–Kanade database. It used features detected by Gabor filters and achieved a recognition rate of 79.5%. In 1.71 seconds. Primitive surface feature distribution (PSFD) was used on manually selected points in [29] to classify facial expressions using Quadratic Discriminant Classifiers, Linear Discriminant Analysis, Naïve Bayesian Classifiers and Support Vector Classifiers. The above method has only been done on static images, and is limited due to the manual selection process of fiducial points.
2.4.2 3D Methods for Recognition of Discrete Emotions

In [46], Hu et al. used the linear Bayes normal classifier (LDC), quadratic Bayes normal classifier, Parzen classifier and SVM classifier on displacements vectors (using the neutral face as a reference) with the BU-3DFE database to classify facial expressions into six discrete emotions (angry, disgust, fear, happy, sad and surprise). They showed that recognition from 45 degrees to the frontal view is better than the recognition rate of the frontal view and SVM classifiers produced the best results. In [32], support vector machines were used on 3D shape models to classify data from the BU-3DFE database into six discrete emotions. In [45], SVM was used on data extracted from reconstructed 3D facial expressions (from 2.5D data) to recognize four facial expressions (neutral, anger, surprise, smiling) to achieve a recognition rate of 60.9% without the use of a reference face and from any viewpoint ranging from -45 to 45 degrees around the y axis. It uses the BU-3DFE database and performs the recognition at a speed of 65.33 seconds per set of data. Ant colony and particle swarm optimization was used in [31] to create a set of classification rules which were used to classify 3D point clouds from the BU-3DFE into six facial expression classes (anger, disgust, fear, happiness, sadness and surprise) at a rate of 92.3%. In [28], a regularized Adaboost classifier, a collection of subsequent weak classifiers that focuses on the hard to classify examples, was used to classify MPEG-4 facial action units into the six basic emotions with a recognition rate of 95.1% using the BU-3DFE database. In [30] bilinear models were used on an elastically deformable model to classify 3D facial expression data from the BU-3DFE database into six discrete emotions.

2.4.3 2D Methods for Recognition of Emotion in Two Dimensions

In [40], neurofuzzy networks were used to generate classification rules based on MPEG-4 FAPs in order to classify user specific emotions of face images into one of four quadrants in the pleasure-arousal emotion model. It used a self-generated database naturalistic database that contains both audio and visual cues. In [39], back propagation on dynamically linked feature points detected by Gabor wavelets was used to create a three layer neural network that classifies facial expression images in the pleasure and arousal dimensions. In [56], similarity measurements on independent components analysis parameters for pixel intensity values was used to classify images from the Korean facial expression database into measurements on the pleasure/arousal dimensional model of emotion achieving 90.9% on the pleasure dimension, and 66.6% on the arousal dimension. Manifold learning and local linear embedding representations
of pixel intensities was used in [57] to classify images from the Korean database in the pleasure/arousal dimensional model of emotion to achieve a recognition rate of 90.9% in the pleasure dimension and 56.1% in the arousal dimension.
Chapter 3

3 Facial Feature Detection

One objective of this work is to develop a system that is able to detect and locate a collection of feature points in a facial expression image accurately and in real time. The proposed facial feature detection system accepts a grey-scale image as input and outputs a vector that describes the location of the feature points. Figure 3-1 outlines the process for the facial feature detection system. This section will discuss the theory behind the feature tracking algorithm and SVMs.

![Facial Feature Detection Flow Chart]

**Figure 3-1:** Facial Feature Detection Flow Chart
3.1 Selected Facial Features

Following the feature selection process in [21], the facial feature points used are as follows:

1. Five points on the top of each of the left and right eyebrow (10 points)
2. Six points around each of the two eyes (12 points)
3. Four points along the bridge of the nose (4 points)
4. Five points along the nostrils (5 points)
5. One point on each corner of the mouth (2 points)
6. Five points on the outer and inner lip for the top and bottom lips (20 points)
7. Fifteen points from ear to ear along the outline of the face (15 points)

Figure 3-2 and Figure 3-3 show the locations of all the feature points.

**Figure 3-2:** The locations of the 68 feature points

**Figure 3-3** The 68 facial feature point on a sample image (image courtesy of JAFFE database)
3.2 Constrained Local Models (CLMs)

In the facial feature detection system, a CLM is used to align a face shape to a facial expression image. The CLM utilizes a collection of patch experts (in this work, SVM feature detectors) that are learned for each feature points, in conjunction with a shape model that describes how the face shape can deform [58]. The patch experts search the image for their corresponding features, and the algorithm fits these feature point locations to a shape model, restricting the final output to the allowable spatial variations.

3.2.1 Learning the Shape Model

A shape model that describes the shape variations of different facial expression images consists of the average positions of the feature points and a set of parameters that control the modes of variation [59]. Algebraically, this can be represented as:

\[ z' = z + Vp \quad (3-1) \]

where \( z = [x_1^T, ..., x_N^T] \), \( x \) is vector containing the coordinates of a feature point, \( V \) is the set of eigenvectors associated with the variation in the training data, \( p \) is a parametric vector that describes the non-rigid warp from \( z \) to \( z' \), and \( N \) is the number of feature points [21]. Typically, \( z \) in the above equation is the average location of the feature points, but the warp function itself can accept any value of \( z \) and warp it with respect to the parameter \( p \).

To create the shape model, a training set containing images of different facial expressions of different individuals was used and is described in section 5.1.1. Since training images are not necessarily aligned in scale or orientation, it is necessary to apply Procrustes analysis [60] to remove the similarity between the images and align them so that the mean of the aligned training images can be found. The generalized orthogonal Procrustes analysis algorithm iteratively aligns all the images to a mean shape, after which a new mean is calculated from the newly aligned images. The iterations stop when the difference between the mean and calculated mean are less than a certain threshold value. The algorithm is as follows [60]:

1. Select the set of feature points from one image to be the approximate mean shape of the face (i.e. the first shape in the set).
2. Align the shapes of the other faces to the approximate mean shape:
   a. Calculate the centroid of each shape.
   b. Align the shape centroids to the origin, (0,0) in 2D Cartesian coordinates.
c. Normalize each shape with respect to the centroid size.

d. Rotate each shape to align with the approximate mean shape.

3. Calculate the new approximate mean from the set of aligned shapes.

4. Repeat steps 2 to 3 until the difference between the approximate means are negligible

After completing the shape analysis, Principal Components Analysis [15] is performed on the aligned feature points to determine the eigenvectors $V$, which are the modes that define the allowable variations in the face shape. With the calculated mean as $\bar{z}$, the covariance of the $n$ aligned sets of feature points is given by:

$$S = \frac{1}{n-1} \sum_{i=1}^{n} (z_i - \bar{z})(z_i - \bar{z})^T.$$  \hspace{1cm} (3-2)

The eigenvectors of $S$ are calculated, yielding $n$ eigenvectors $v_i$ and corresponding eigenvalues $\lambda_i$. After sorting the eigenvectors from increasing to decreasing powers (values of $\lambda_i$), the first $t$ eigenvectors are chosen to represent $f_v$ of the total variation such that:

$$\sum_{i=1}^{t} \lambda_i \geq f_v \sum_{i=1}^{n} \lambda_i$$  \hspace{1cm} (3-3)

where $f_v$ represents the proportion of the total variation that the eigenvectors can describe and the eigenvectors in the warp function can be described as:

$$V = [\lambda_1 \lambda_2 ... \lambda_t].$$  \hspace{1cm} (3-4)

### 3.2.2 Support Vector Machine Feature Detectors

In order for the facial feature recognition system to detect all the features on the face, specialized feature detectors for each feature point are required to be trained. In the proposed systems, the detectors are linear support vector machines (SVMs).

A support vector machine is a mathematical tool that can be used to classify data into groups. For the proposed system, given a current pixel location, it is desired to determine the likelihood that it is the location of a feature point. As an input vector to the SVM, the $15^2 \times 1$ feature vector formed by concatenating the column vectors of the $15 \times 15$ image patch centered at $x_k$ is used.

For an arbitrary vector $x_i$, a SVM seeks to construct a decision function $E_k(x_i)$ that classifies $x_i$ as either a positive or negative member of a particular class. Given a set of $N$ vectors with known class memberships, an SVM system attempts to create a hyperplane that will separate the members and non-members of a class, which are expressed as inequality constraints,
while maximizing the distance between the hyperplane and the nearest vector. This is achieved by minimizing the following equation [61]:

$$\Phi = \frac{||w||^2}{2} + C \left( \sum_i \xi_i \right)$$

subject to:

$$g_i \cdot w + b \geq +1 - \xi_i \quad \text{for } y_i = +1$$

$$g_i \cdot w + b \leq +1 - \xi_i \quad \text{for } y_i = +1$$

$$\xi_i \geq 0 \; \forall \; i$$

where $g_i$ is the $i^{th}$ vector from the set of training vectors, $w$ is a vector perpendicular to the separating hyperplane, $b$ is the bias such that $|b|/||w||$ is the perpendicular distance from the hyperplane to the origin, $||w||$ is the Euclidian norm of $w$, $\xi_i$ is a slack variable such that $|\xi_i|/||w||$ is the perpendicular distance from the hyperplane to the vector $x_i$. $y_i$ is the membership of the vector $g_i$ and $C$ is a user defined parameter to specify the penalty associated with a feature vector that is on the wrong side of the hyperplane. The slack variables and penalty variable are necessary in the general case for situations in which data is non-separable, but a separating hyperplane is still desired. Figure 3-4 shows a graphical depiction of the hyperplane and example points.

![Graphical depiction of hyperplane and sample points](image)

**Figure 3-4:** Graphical depiction of hyperplane and sample points [61]
This simplifies into the Wolfe dual problem giving the following maximization objective function $L_D[59]$: 

$$
L_D(\alpha_i) \equiv \sum_i \alpha_i - \sum_{ij} \alpha_i \alpha_j y_i y_j g_i \cdot g_j
$$

that is constrained by 

$$
0 \leq \alpha_i \leq C,
\sum_i \alpha_i y_i = 0
$$

where $\alpha_i$ is the Lagrange multiplier for the $i^{th}$ support vector. A support vector is a training point that defines the hyperplane. If all other training points are removed, the support vectors themselves will yield the same hyperplane. This gives the solution 

$$
w = \sum_{i=1}^{N_s} \alpha_i y_i g_i = \sum_{i=1}^{N_s} \alpha_i T_i(x)
$$

where $N_s$ is the number of support vectors and defines a vector perpendicular to the hyperplane, pointing in the direction of positive classification, $\alpha_i$ is the weight for $i^{th}$ the support vectors $T_i(x)$. By projecting the input vector on to $w$, the distance from the hyperplane the input vector can be determined. For the proposed feature classifier, $x_i \cdot w$ is used to evaluate the likelihood that a particular pixel $x_i$ is a feature point, with higher values representing a higher likelihood.

### 3.2.3 Exhaustive Local Search (ELS) [21]

Given a set of SVM feature detectors, an Exhaustive Local Search (ELS) in the neighbourhood of each feature points is used to determine a new estimate of the feature point location. Since the SVM feature detectors are linear and involve only a dot product of two vectors, it is possible to do an exhaustive search without a penalty in computational time. The ELS method in the feature detection system involves solving for 68 local translation updates $\Delta x$. The local updates are chosen by:
where $\mathbb{R}$ is a 2D search window centered at $\mathbf{x}_K$ of size $15 \times 15$, $E_k$ is the SVM classification function described in Section 3.2.2, and $\Delta \mathbf{x}$ is the column vector of the horizontal and vertical distance from $\mathbf{x}_K$ in image coordinates. These local updates can be concatenated to form a vector

$$\Delta \mathbf{z} = [\Delta x_1^T \Delta x_2^T \ldots \Delta x_n^T]^T.$$  \hfill (3-9)

To determine the local translation $\Delta \mathbf{x}_K$, the linear SVM is used and evaluated as:

$$E_k[I(\mathbf{x} + \Delta \mathbf{x})] = \sum_{i=1}^{NS} \alpha_i T_i(x)^T I(\mathbf{x} + \Delta \mathbf{x}) = I(\mathbf{x} + \Delta \mathbf{x}) \sum_{i=1}^{NS} \alpha_i T_i(x)$$  \hfill (3-10)

where $\alpha_i$ is the $i$th support weight, $T_i$ is the $i$th support vector, and $NS$ is the number of support vectors, and. $\sum_{i=1}^{NS} \alpha_i T_i(x)$ is a constant that can be computed prior to run-time and $I(\mathbf{x} + \Delta \mathbf{x})$ is an image patch reshaped as a column vector and normalized to have zero mean and unit norm.

3.2.4 CLM Fitting

The constrained local model algorithm attempts to use the results of the SVM detectors in the ELS search with the shape model to iteratively find set of parameters $\mathbf{p}$ that warps the feature point locations guesses in the new image $\mathbf{z}$ so that it aligns with the template (average face). The warp function, defined as:

$$W(\mathbf{z}; \mathbf{p}) = \mathbf{z}' = \mathbf{z} + V\mathbf{p}$$  \hfill (3-11)

warp the current guess $\mathbf{z}$ to $\mathbf{z}'$. Aligning the source image $I$ to the template image $T$ by solving for $\mathbf{p}$ is done iteratively and a variety of methods are available to update $\mathbf{p}$ with $\Delta \mathbf{p}$ [62]. The inverse composition algorithm solves for the increment $\Delta \mathbf{p}$ by minimizing:

$$\sum_x [T(W(z; \Delta \mathbf{p})) - I(W(z; \mathbf{p}))]^2$$  \hfill (3-12)

to solve for $\Delta \mathbf{p}$ and updating $W(z; \mathbf{p})$ as $W(z; \mathbf{p}) \circ W(z; \Delta \mathbf{p})^{-1}$ where $\circ$ is the composition operator defined as $W(z; \mathbf{p}) \circ W(z; \Delta \mathbf{p}) = W(W(z; \Delta \mathbf{p}); \mathbf{p})$. The inverse compositional algorithm involves warping the Image $I$ to the template model $T$, and solving for the warp parameter $\Delta \mathbf{p}$ that will warp the new image to match the location of the feature points estimates.

By solving for $\Delta \mathbf{p}$ in this way, only the Hessian and Jacobian evaluated at $\mathbf{p} = 0$ is required to be calculated, both of which can be computed prior to run-time.
To determine the warp update $\Delta p$ based on $\Delta z$, a weighted least squares optimization of the form:

$$\Delta p = (JWJ^T)^{-1}JW\Delta z$$

(3-13)

can be used where $J$ is the Jacobian defined as Equation 3-14 and $W$ is the weighting matrix defined as Equation 3-15:

$$J = \frac{\partial W(z; 0)}{\partial p} = \frac{\partial W(z; p)}{\partial p} |_{p=0}$$

(3-14)

$$W = \text{diag}(w_{x1}, w_{y1}, ..., w_{xN}, w_{yN}).$$

(3-15)

To select the weights, a probability function is created by fitting a logistic regression function to the output of the SVM for a collection of test images given by:

$$P(y = 1|E_k[I(x + \Delta x)]) = \frac{1}{1 + e^{a_kE_k[I(x+\Delta x)]+b_k}}$$

(3-16)

where $P(y = 1|E_k[I(x + \Delta x)])$ is the probability that pixel $x + \Delta x$ is a feature given the pixel intensity value of $I$ centered at that point. $a_k$ and $b_k$ are logistic parameters determined experimentally for the $k^{th}$ feature. These probabilistic values are determined at runtime populate the weight matrix in Equation 3-15 such that:

$$w_{xk} = w_{yk} = w_k = P(y = 1|E_k[I(x_k + \Delta x_k)]).$$

(3-17)

In the proposed system, the Jacobian $J = V^T$ due to the fact that the warp function $W(z; p)$ that we have defined is linear. In fact, the following holds true, and simplifies our problem:

$$W(z; \Delta p)^{-1} = z - V\Delta p = W(z; -\Delta p)$$

(3-18)

$$W(z; p) \circ W(z; \Delta p)^{-1} = z + Vp - V\Delta p$$

(3-19)

$z_{i+1}$ becomes $z_i + Vp_i - V\Delta p_i$, and $p_{i+1}$ as $p_i - \Delta p_i$. The process of detecting the features through the ELS, and inverse composition update step for CLM fitting is repeated until $\Delta p < \epsilon$, where $\epsilon$ is a threshold value determined experimentally. When this condition is reached, the solution has converged and the new feature points are detected. Their locations $z_f$ are given by:

$$z_f = \bar{z} + Vp.$$ 

(3-19)
3.3 Chapter Summary

Given an input facial expression image, the proposed system utilized a constrained local model to detect a set of facial feature points in real time. The CLM is characterized by having linear SVM as patch experts, an exhaustive local search pattern and an inverse compositional warp update. The shape model is trained on a set of labeled facial expression images and only the eigenvectors with the strongest representational power are selected.
Chapter 4

4 Facial Expression Recognition

The proposed system for facial expression system follows the flow chart as shown in Figure 4-1.

Figure 4-1: Flowchart for proposed facial expression recognition system

The proposed system for facial expression recognition will convert labeled feature points from classified facial expression images into FAPs and train two multi-class SVMs with a vector of FAP measurements as the input (one for pleasure and one for arousal). The feature point detection system will determine the located feature points. The facial expression recognition system will then convert the feature points to FAPs and input them into the multi-class SVMs to output a value for pleasure and arousal of the image.
The emotional model used for recognition is the two dimensional model of pleasure and arousal as shown in Figure 4-2.

![Figure 4-2: Two dimensional model of emotion (Pleasure-Arousal)](image)

This model is very flexible and will allow for the proposed system to detect the transitions between different emotions and how the pleasure and arousal states of the subject in the image change over time.

### 4.1 Multi-Class SVMs

Binary SVMs as described in Section 3.2.2 classify an input vector into either a member or non-member of particular class. To utilize SVMs for classification of multiple classes, a collection of binary SVMs can be combined to form a multi-class SVM. For N classes, N binary SVMs can be trained with the members of that class as positive samples, and members of all other classes as negative samples. These SVMs together form a multi-class SVM. For the proposed system, the facial feature points in the facial expression images will be converted to FAPs. The corresponding images to these FAPs will determine the group membership. Depending on the experiment type, the groups may be high, medium and low pleasure or arousal types. They could be a set of bins with minimum and maximum range of values. An example of the latter type would be groups that divide arousal values into seven bins where membership is determined by the value of the arousal rounded to the nearest integer. The seven-class SVM consists of seven SVMs and all must be applied with the input vectors independently to determine whether the input is a member of that class. For a successful classification, the output of all SVMs should be negative with the exception of the class that it belongs to.
4.2 Kernel Functions

It is not guaranteed that the training data for the SVMs is separable by a linear SVM. However, a kernel function can be defined that transforms the input data into a higher dimensional space and attempts to classify it.

In Equation 3-6, the optimization problem involves the dot product of $g_i . g_j$. If each of these vectors was mapped into a different space with an operator called $\phi$ that maps from $\mathbb{R}^d$ to $\mathbb{H}$, $g_i . g_j$ in $\mathbb{R}^d$ becomes $\phi(g_i) \cdot \phi(g_j)$ in $\mathbb{H}$. If there exists a function $K$ such that:

$$K(\phi(g_i), \phi(g_j)) = \phi(g_i) \cdot \phi(g_j)$$

(4-1)

the optimization to solve Equation 3-6 would only require $K$ to solve for the SVM and $\phi$ would not be required. The kernel functions investigated in this work are polynomial kernels, which are defined by:

$$K(g_i, g_j) = (g_i^T \cdot g_j + 1)^d$$

(4-2)

where $d = 1, 2$ or $3$.

4.3 Chapter Summary

A facial expression recognition system has been introduced which will convert a set of feature points into facial action parameters. These parameters will be input into a multi-class SVM that will classify the different facial expression images into values for pleasure and arousal in the two dimensional model of emotion.
Chapter 5

5 Experiments

5.1 Facial Feature Training

5.1.1 Obtaining Images For Training

A training set is used herein for training the proposed method to recognize facial features. The set consists of a combination of images from (i) the Japanese Female Facial Expression database (JAFFE), (ii) the Facial Action Coding System database (FACS) and (iii) a collection of images extracted from videos in our own experiments. The JAFFE database contains images of ten different Japanese women posing seven different facial expressions (neutral, happy, sad, angry, surprise, disgust and fear). The FACS database contains images of Caucasian males and females of varying age, with each image displaying a face activating a specific set of action units (AUs), not necessarily corresponding to one specific emotion. Since the JAFFE database only contains images of Japanese females, and the FACS database contains only images of Caucasian males and females, more images are required to increase the variation in appearance (sex, age, skin tone) of the data set. Also, the JAFFE database contains posed and forced facial expressions, which rarely occur in real situations. Thus, a database of images was generated using video clips of various activities to elicit facial expressions from individuals. This database served to increase the appearance variation in the training set by adding ten more individuals of different age, ethnicities, gender and facial features. It also provided facial expressions that occur naturally in everyday interactions to help train the facial expression classifier.

Namely, film clips were used to elicit emotions with a camera to capture any facial expressions that arose during watching of clips. The clips that were chosen were based on a study conducted in [63]. Subjects were individually sitted in a room and asked to watch a series of seven clips, each designed to elicit fear, happiness, surprise, sadness, anger or disgust. Between each clip viewing, they were asked to fill out a questionnaire to select what emotions they felt while watching the clip that contained questions about the state of arousal, as well as what discrete emotions best described their feelings during the film. This also served as a tool to
restore the user to a more neutral baseline, as watching the clips immediately could create lingering effects and affect the subject’s emotions in the clips following. Video data was gathered for ten subjects and was segmented into jpg images at a rate of 5 frames per second (fps). The extracted images were analyzed and a set of 71 images were selected to be used as part of the training data. The selected images contained facial expressions that differed from the neutral facial expression throughout the watching of the clips, and exhibited emotions during moments in the films.

To train the SVM feature detectors, 38 images from the JAFFE, 13 images from the FACS and 71 images from our own database were used. However, to train the facial expression recognition system, the 71 images from our own database were only used to promote the recognition of natural facial expressions.

5.1.2 Finding the True Mean

Figure 5-1 shows the unaligned locations of the labeled feature points from the raw training images. Since the images are different sizes and the faces are not all centered in the same position, the feature points are scattered and it would be impossible to perform Principal Component Analysis on the data and obtain usable results. Thus, Procrustes Analysis was used to find the true mean of the data set as shown in Figure 5-2a. With the true mean, all the existing images were normalized and aligned to the true mean as seen in Figure 5-2b.

![Figure 5-1: Feature points of unaligned training images](image-url)
5.1.3 Principal Components Analysis (PCA)

PCA was applied to the aligned images and true mean to obtain a set of 136 Eigenvectors with size $136 \times 1$, with 136 eigenvalues. The eigenvalues were sorted in descending order, and the top twelve eigenvectors were selected based on their eigenvalues so that 95% of the data could be represented by the selected eigenvectors. This reduces the number of parameters required to describe the feature point locations, reducing the computational time required for other algorithms in the method. Figure 5-3 shows the true average face under the influence of the first three eigenvectors in both the positive and negative directions.
Figure 5-3: Variations of the shape model for: a) first eigenvector, b) second eigenvector and c) third eigenvector, from $-60\lambda_i$ to $+60\lambda_i$ where $\lambda_i$ is the corresponding eigenvalues.

5.1.4 SVMs for Feature Detection

The next step is to train the SVM feature detectors. Each of the 68 feature points requires an SVM so that the program has a means to identify whether a pixel is a candidate for a feature point.

To train each SVM, positive samples and negative samples are required as input to the training algorithm. To obtain the positive samples for the $i^{th}$ feature point, the intensity values of the 15 by 15 pixel image patch centered at the true value of the feature point was extracted from the grey-scale images. To obtain the negative samples, identically sized image patches were extracted near the true location of the feature point with a minimum distance of 4 pixels away, and a maximum distance of 16 pixels away. Since the emotion classification input feature is dependent on the location of these points, focus was placed on selecting negative image patches closer to the true location of the feature point to improve the SVM’s ability to distinguish between feature points and non-feature points near the true feature point location.

After obtaining the positive and negative samples, the image patches were converted to a single 225 x 1 sized vector and normalized to zero mean and unit norm. This normalization technique is used to reduce the effect that lighting and skin tone in the training images would have on the SVM classifier. For example, an SVM for a feature point that lies on the border of the lips will be less affected by images from two different people with different skin or lip colours if the data is normalized before training, than if the intensities of the image patch were directly used.

With a set of positive and normalized image patches for each SVM, the average positive image patch as calculated was determined, and the top 1500 negative patches with the largest
Euclidian distance from the average positive patch were selected as inputs to the SVM for each feature point. This operation reduced the total number of input vectors for the SVM training algorithm, which in turn reduces the amount of time required for training of the SVM.

The SVM training algorithm used on the input data produced a linear support vector and a set of weights to define the parameters of the SVM. The SVM was tested on image patches of the training data and the results obtained are shown in Figure 5-4. Figure 5-4a shows the SVM response as a color map. The brighter spots represent a higher likelihood that a particular pixel is a feature, and the dark spots represent a very low likelihood that a particular pixel is a feature. The blue cross represents the highest value for the SVM response, and is the point that the SVM believes is the best candidate for a feature point. Figure 5-4b shows the corresponding point in the original image that the blue cross represents, and Figure 5-4c shows the true value of the feature point from the labeled training data.

**Figure 5-4** a) SVM response and location of maximum response and b) corresponding point in actual image and c) Actual feature point location.

5.1.5 Logistic Regression

To determine the weights as defined in Equation 3-17, the SVM responses of positive and negative patches were analyzed in the form of logistic regression. The SVM responses were
ranked in ascending order and a metric was based on Equation 3-16 to represent the likelihood that a particular image patch is a feature given a particular SVM response. For example, if the SVM response was 2.5, the y-value on the curve where x = 2.5 would represent the probability that an image patch with a score of 2.5 or higher would be a feature point.

A new set of 31 images were selected and labeled to be used as the test data for the estimation of a and b. Positive samples were selected as the labeled points and the eight adjacent pixels. Negative samples were selected as random points in the local 40 x 40 patch of pixels surrounding the labeled point. Such as before, the image patches were normalized to zero mean and unit norm so that consistency for the SVM inputs were preserved.

The log parameters a and b were selected manually and superimposed onto the probabilities until the desired fit was obtained to give Figure 5-5. Since the data did not perfectly fit to all the probability curves, it was necessary to select portions of the data that were more important than the rest. Since the algorithm is only interested in the highest response, the logistic fitting was focused more on the larger SVM response values. As seen in the figure, the fitted curve does not represent the data well until the SVM Response = 6. The lower SVM responses are not as likely to be features, and the logistic curve will likely not be evaluated at those points. During fitting, care was also made to be more conservative, so that the fitted probabilities never overestimated the likelihood that a pixel is a certain feature. This resulted in a 68 x 2 vector that contained the log parameters a and b to be input into the CLM fitting algorithm.

![Logistic Regression of Feature 16](image)

**Figure 5-5:** Logistic Regression of Feature 16
5.1.6 Constrained Local Model Fitting

To test the CLM algorithm, a Windows application was developed in C++. The interface shown in Figure 5-6a below allows the user to open an image and select a box that simulates an eye/nose tracker. Based on the parameters of the box, initialization of the CLM fitting procedure causes the average face, scaled to the size of the box, is drawn on the image as an initial guess as shown in Figure 5-6b. For initialization, the parameters are as follows:

\[ z_0 = \bar{z} + Vp_0 \]  \hspace{1cm} (5-1)

\( \bar{z} \) is the average face from Procustes analysis scaled to the box parameters, \( V \) is the matrix containing the column of eigenvalues determined in Equation 3-4 to represent 95\% of the variation, \( p_0 \) is the initial warp guess (in the case, it is the zero vector), and \( z_0 \) is the initial locations of all the feature points.

![Figure 5-6](image)

**Figure 5-6:** a) Simulating a face and nose tracker, and b) superimposing an initial guess
Figure 5-7 shows the CLM results for the $i^{th}$ iteration as it iterates for $i = 0, 1, 5$ and 10 next to the true labeled data for training and Figure 5-8 show the CLM results for the $i^{th}$ iteration as it iterates for $i = 0, 1, 5$ for an image not present in the training set.

**Figure 5-7**: CLM fitting for the $i^{th}$ iteration and true value of feature point locations of a trained image, a) $i = 0$ b) $i = 1$ c) $i = 5$ d) $i = 10$ and e) true values.

**Figure 5-8**: CLM fitting for a non-trained image, a) $i = 0$ b) $i = 1$ and c) $i = 5$

The results show that the CLM does not accurately detect the feature point locations. Possible reasons for this could be:

1. The SVM feature detectors for each feature point may not be accurate enough. More positive and negative samples are required for training. At the cost of more computational time required during the training process, it may help to not select the top 1500 negative
image patches from the total set of image patches for training. Alternatively, a different metric other than the Euclidian norm could be used to select the features best suited as negative patches.

2. The labeled positive sample points may not be accurate enough because there is a high level of difficulty in determining the correct location of a defined feature point. Feature points on the borders around the lips and eyes are not always clear, and there is also a large amount of difficulty to match feature points between different individuals due to variation in facial appearance. To fix this, the number of feature points could be decreased and only the strong feature points could be selected.

3. The search area for the feature points may be too small due to the resolution of the image. The individual SVM feature detectors could be stuck at a local maximum and the influence from the other feature detectors may not be strong enough to move the CLM in range of the true feature point location. Methods to improve on this could be to change to a lower resolution of camera (or warp the face to a smaller patch of pixels), to develop a scheme that searches a larger area for smaller sub patches that are likely to contain a feature, or to simply increase the size of the search area.

5.1.7 Convergence Results

To analyze the convergence of the CLM fitting algorithm, the Euclidian norms of the change in feature point locations $\Delta z_i$ and parameters $\Delta p_i$ between iterations were plotted in Figure 5-9 and Figure 5-10 below for the two different cases shown in Figure 5-7 and Figure 5-8 above. It is evident that the convergence rate is quite fast, occurring in 3 or 4 iterations.
Comparing the Figures for $\Delta z_i$ and $\Delta p_i$, it is possible to experimentally determine a threshold value $\epsilon$ in which the CLM iterations will stop. This condition was calculated to be
\[ \| \Delta p_i \| < \epsilon = 20 \] (5-2)

5.1.8 Iteration Duration

The duration of each iteration was measured 30 times and the average duration per iteration was 0.32s with a standard deviation of 0.2s. Smarter search patterns (normal directions for face border) or reduced feature points could be implemented to minimize computational time.
Reducing the number of iterations for convergence will also improve the overall speed of the operation.

5.2 Facial Expression Recognition

5.2.1 Two Dimensional Model of Emotion

The model of emotion used for classification is the two dimensional model of pleasure and arousal as described by Russell [7] is shown in Figure 5-11.

![Two dimensional model of emotion for pleasure-arousal](image)

**Figure 5-11**: Two dimensional model of emotion for pleasure-arousal [7]

5.2.2 Experimental Method

5.2.2.1 Method 1: Numeric Rated Scale

To assign a value for pleasure and arousal for each image, two methods were utilized. The first method was to have coders to assign a value of pleasure and arousal from a scale of 1 to 10 for each image. In the case of pleasure, 1 represents low pleasure and 10 represents high pleasure. For arousal, 1 represents low arousal and 10 represents high arousal. The procedure was repeated twice, with the images sorted into different orders for the second trial. The data maintained for the rest of the experiments were those from trial 2. For the purpose of further discussion, this data collected using this method will be herein referred to as the numeric or rated data.
5.2.2.2  Method 2: Ranks

The second method required the coders to sort subsets of images, with each subset containing images of only one person. Each coder then ranked the images in these subsets from lowest level of arousal to highest level of arousal, and from lowest level of pleasure to highest level of pleasure. Then, each subset was further divided into three smaller subsets based on the ranked order provided by a particular coder. The smaller subsets were grouped together, forming six total sets. For the pleasure dimension, a set of images containing all the lowest ranked images for each individual was created. Similarly, a set of images containing all the high pleasure images and a set of images containing all the medium pleasure images was created. Each coder received three sets of images for pleasure and three sets of images for arousal to rank. These sets of data are specifically created based on the first level of rankings. Similar to method one, the average ranks of each image was calculated as the rank for that image. For the purpose of further discussion, this data collected using this method will be herein referred to as the ranked data.

5.2.3  Inter-Coder Reliability

Before any analysis can be further conducted on the coded images, it is necessary to provide some level of inter-coder reliability to verify that the different coders are rating the images in a similar way. To do this, we employed the use of Krippendorff's Alpha $\alpha$ [64], a metric that is used to measure the agreement between any number of observers for ordinal or interval data. The reliability data for ranked and rated data for both pleasure and arousal are shown in Table 1 below.

<table>
<thead>
<tr>
<th>Trial</th>
<th>Pleasure $\alpha$</th>
<th>Arousal $\alpha$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.6450</td>
<td>0.3351</td>
</tr>
<tr>
<td>2</td>
<td>0.8408</td>
<td>0.4070</td>
</tr>
</tbody>
</table>

Comparing the values of between trials, the inter-coder reliability improves considerably from the first trial to the second trial. This is most likely due to the fact that the coders are now familiar with the images after Trial 1 and have developed a set of rules in which they will code the images. They have seen the types of images that they will rate, and have a good idea of what
they consider to be high or low pleasure or arousal. Looking at the values for the second trial, the reliability for pleasure is quite high at $\alpha = 0.8408$, but the reliability for arousal is quite low at $\alpha = 0.4070$. As per [64], Krippendorff's alpha considers a value of 0.8 to be reliable and any value lower than 0.667 to be poor, but the acceptable level of reliability is application specific and must be selected by the experimenter. This low level of reliability for the arousal dimension is not unexpected as arousal is a more abstract concept compared to pleasure, and visual cues for low vs. high arousal are much more difficult to detect than the cues for low and high pleasure.

Upon further examination of the coded results, it was noticed that one of the coders had rated a disproportionate number of images with arousal = 3. To investigate the impact of this, this coder's data was removed, and the reliability values were recalculated as shown in Table 2 under the '\(\alpha\) for 4 Coders' column.

**Table 2:** Comparison of inter-coder reliability $\alpha$ for rated data for pleasure and arousal with 5 and 4 coders

<table>
<thead>
<tr>
<th>Metric</th>
<th>$\alpha$ for 5 Coders</th>
<th>$\alpha$ for 4 Coders</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pleasure Trial 1</td>
<td>0.6450</td>
<td>0.6268</td>
</tr>
<tr>
<td>Pleasure Trial 2</td>
<td>0.8408</td>
<td>0.8580</td>
</tr>
<tr>
<td>Arousal Trial 1</td>
<td>0.3351</td>
<td>0.4972</td>
</tr>
<tr>
<td>Arousal Trial 2</td>
<td>0.4070</td>
<td>0.6747</td>
</tr>
</tbody>
</table>

The reliability for the pleasure values remain relatively unchanged, but the exclusion of the 5th coder increases the reliability of the arousal values to $\alpha = 0.6747$.

Using method 2, Krippendorff's $\alpha$ was calculated and displayed in Table 3

**Table 3:** Inter-coder reliability $\alpha$ for ranked data for pleasure and arousal

<table>
<thead>
<tr>
<th>Dimension</th>
<th>$\alpha$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pleasure</td>
<td>0.9693</td>
</tr>
<tr>
<td>Arousal</td>
<td>0.9427</td>
</tr>
</tbody>
</table>

The reliability for pleasure and arousal using the second method are much higher than the first method. This is expected because the coders can make a direct comparison between images and decide whether an image has a higher or lower arousal/pleasure value than another. In the first
method, no such comparisons are allowed, and biases may occur depending on the order in which images were rated.

5.2.4 Facial Action Parameters (FAPs)

To classify a set of facial feature points in an image to a value of pleasure and arousal, a feature vector of FAPs was created to reduce the dimension of the facial feature points to a more computationally efficient size. The feature points were converted into facial action parameters as shown in Figure 5-12 and below. The Equations in Table 4 are defined as follows:

\[
\text{Slope}(P_1, P_2) : \frac{(P_{2y} - P_{1y})}{(P_{2x} - P_{1x})}
\]

\[
\text{Abs}(x) = \text{absolute value of } x
\]

\[
\text{avgX}(P_1, P_2, ..., P_N) = \text{average X value of } P_1 \text{ to } P_N
\]

\[
\text{avgY}(P_1, P_2, ..., P_N) = \text{average Y value of } P_1 \text{ to } P_N
\]

\[
\text{avg}(P_1, P_2, ..., P_N) = \text{average point of } P_1 \text{ to } P_N
\]

These 11 FAPs are used as input vectors to the multi class SVMs for the pleasure and arousal outputs for facial expression recognition.

Figure 5-12: Feature point locations
Table 4: Facial Action Parameter Definitions

<table>
<thead>
<tr>
<th>FAP #</th>
<th>Description</th>
<th>Equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Slope of Outer Eyebrow</td>
<td>Slope(3, 4) + Slope(4, 5) - Slope(6, 7) - Slope(7, 8)</td>
</tr>
<tr>
<td>2</td>
<td>Slope of Inner Eyebrow</td>
<td>Slope(1, 2) + Slope(2, 3) - Slope(8, 9) - Slope(9, 10)</td>
</tr>
<tr>
<td>3</td>
<td>Distance from Right Eyebrow to Right Eye</td>
<td>Abs(avgY(7,8,9) - AvgY(17,18,19,20,21,22))</td>
</tr>
<tr>
<td>4</td>
<td>Distance from Left Eyebrow to Left Eye</td>
<td>Abs(avgY(2, 3, 4) - AvgY(11, 12, 13, 14, 15, 16))</td>
</tr>
<tr>
<td>5</td>
<td>Ratio from Nose to mouth Distance and Height of Mouth</td>
<td>Abs(avgY(27, 28, 29, 30, 31) - avgY(34,35,36)) / Abs(avgY(35)- avgY(41))</td>
</tr>
<tr>
<td>6</td>
<td>Ratio of Right Eye Height to Distance between Eyes</td>
<td>avgY(18, 19) - avgY(21, 22) / abs(avgX(20)- avgX(11))</td>
</tr>
<tr>
<td>7</td>
<td>Ratio of Left Eye Height to Distance between Eyes</td>
<td>avgY(12, 13) - avgY(15, 16) / abs(avgX(20)- avgX(11))</td>
</tr>
<tr>
<td>8</td>
<td>Ratio between Width of Mouth and Height of Mouth</td>
<td>Abs(avgX(32) - avgY(38)) / Abs(avgY(35)- avgY(41))</td>
</tr>
<tr>
<td>9</td>
<td>Ratio between Distances Between Upper Lip and Inner Lips and Lower Lip and Inner Lips</td>
<td>Abs(avgY(35) - avgY(32, 38)) / Abs(avgY(41) - avgY(32, 38))</td>
</tr>
<tr>
<td>10</td>
<td>Difference Between Slopes from Left to Right of Left corner, Middle and Right Corner of Mouth</td>
<td>Slope(avg(46, 51), avg(44, 53)) - Slope(avg(48, 49), avg(46, 51))</td>
</tr>
<tr>
<td>11</td>
<td>Distance from Upper Lip to Nose</td>
<td>Abs(avgY(27, 28, 29, 30, 31) - avgY(34,35,36))</td>
</tr>
</tbody>
</table>

5.2.5 Emotion Classification

Four sets of experiments were conducted to analyze the ability of support vector machines to classify the emotions of different facial expression images in terms of pleasure and arousal. Two experiments were conducted with the data obtained through method 1 of data acquisition where coders provided a numerical rating between 1 and 10 for each image, and two experiments were conducted on the ranked data. Each experiment investigates the use of linear, quadratic and cubic
kernel functions as parameters for the SVM. The slack variable that is used represents the allowable error to be classified as true. If slack = 0, the classification must be exact to be considered true. If slack = 1, the classification is still considered true if it is off by 1 class. (i.e., an image with a pleasure value of 2 classified as 3).

**Experiment 1: Seven-Class SVM for Quantified Coder Data**

To create the seven-class SVM, the coded data had to be split into seven groups. The original distribution of the rated data is shown in Figure 5-13a. In order to analyze the performance of a seven class SVM, a four-fold validation was selected. However, a seven class SVM could not be created with such low frequencies for the extreme values, as there would be no data to test or train. To deal with this, the extreme values of ratings were combined with the closest rating value, in this case, images ratings of 1 and 2 were grouped in the same category as the rating of 3, and the images with ratings of 9 and 10 were grouped together with images of rating 8. This gave a distribution for pleasure as shown in Figure 5-13b.

![Frequency distribution graphs](image)

**Figure 5-13:** a) histogram of rated data before re-categorization, and b) histogram of rated data after re-categorization

Performing four-fold validation on the 7 class SVM yielded results as tabulated in Table 5 and Table 6. Results were considered true when the average of the SVM classifications was within 1 of the true value.
Table 5: 7-class SVM Results for rated data in pleasure dimension with slack = 1

<table>
<thead>
<tr>
<th>Kernel Function</th>
<th>Training Set Accuracy (%)</th>
<th>Test Set Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear</td>
<td>62</td>
<td>50</td>
</tr>
<tr>
<td>Quadratic</td>
<td>100</td>
<td>57</td>
</tr>
<tr>
<td>Cubic</td>
<td>100</td>
<td>49</td>
</tr>
</tbody>
</table>

Table 6: 7-class SVM Results for rated data in arousal dimension with slack = 1

<table>
<thead>
<tr>
<th>Kernel Function</th>
<th>Training Set Accuracy (%)</th>
<th>Test Set Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear</td>
<td>49</td>
<td>44</td>
</tr>
<tr>
<td>Quadratic</td>
<td>100</td>
<td>67</td>
</tr>
<tr>
<td>Cubic</td>
<td>100</td>
<td>58</td>
</tr>
</tbody>
</table>

Experiment 2: Three-Class SVM for Rated Coder Data

Similar to experiment 1, the data was reorganized into 3 classes where images with values of 1, 2 and 3; 4, 5 and 6; and 7, 8, 9 and 10 were grouped into separate classes, yielding a distribution for pleasure as shown in Figure 5-14.

![Figure 5-14: Rated data re-categorized for 3-class SVM](image-url)
Performing the four-fold validation on the 3 class SVM yielded results as tabulated in Table 7 and Table 8. Results were considered true when the average of the SVM classifications was exactly the same as the true value.

**Table 7:** 3-class SVM results for rated data in pleasure dimension with slack = 0

<table>
<thead>
<tr>
<th>Kernel Function</th>
<th>Training Set Accuracy (%)</th>
<th>Test Set Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear</td>
<td>77</td>
<td>61</td>
</tr>
<tr>
<td>Quadratic</td>
<td>100</td>
<td>54</td>
</tr>
<tr>
<td>Cubic</td>
<td>100</td>
<td>49</td>
</tr>
</tbody>
</table>

**Table 8:** 3-class SVM results for rated data in arousal dimension with slack = 0

<table>
<thead>
<tr>
<th>Kernel Function</th>
<th>Training Set Accuracy (%)</th>
<th>Test Set Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear</td>
<td>80</td>
<td>61</td>
</tr>
<tr>
<td>Quadratic</td>
<td>100</td>
<td>62</td>
</tr>
<tr>
<td>Cubic</td>
<td>100</td>
<td>70</td>
</tr>
</tbody>
</table>

**Experiment 3: Binary SVM for Ranked Coder Data**

Performing four-fold validation on the binary SVM yielded results as tabulated in Table 9 and Table 10. Results were considered true when the average of the SVM classifications was exactly the same as the true value.

**Table 9:** Binary SVM for ranked data in pleasure dimension with slack = 0

<table>
<thead>
<tr>
<th>Kernel Function</th>
<th>Training Set Accuracy (%)</th>
<th>Test Set Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear</td>
<td>69</td>
<td>46</td>
</tr>
<tr>
<td>Quadratic</td>
<td>100</td>
<td>51</td>
</tr>
</tbody>
</table>
Experiment 4: Three-Class SVM for Ranked Coder Data with Exhaustive Search over FAPs

Based on the results in Experiments 1, 2 and 3, an exhaustive search on the ranked data was conducted with a three-class SVM to determine the effect that different features had on the results for both pleasure and arousal.

First, the ranked data was split into three groups, with each being classified as high, medium or low for pleasure/arousal. The three-class SVM was trained using fifteen images from each group: the 15 lowest arousal/pleasure images from the low class, 15 highest arousal/pleasure images from the high class, and middle 15 arousal/pleasure images from the medium class. This maximized the image ranks to assist the SVM in providing better results.

Of the 15 images in each group, a leave-one-out validation scheme, in which one facial expression image was used for testing with the remaining used for training. This was repeated 15 times, so that each image became the test image once, and the average results determined the result of that classifier.

This process was repeated for every combination of input feature vectors from the 11 selected in Selection 5.2.4 with linear, quadratic and cubic kernels. This resulted in three-class

<table>
<thead>
<tr>
<th>Kernel Function</th>
<th>Training Set Accuracy (%)</th>
<th>Test Set Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear</td>
<td>51</td>
<td>30</td>
</tr>
<tr>
<td>Quadratic</td>
<td>100</td>
<td>52</td>
</tr>
<tr>
<td>Cubic</td>
<td>100</td>
<td>48</td>
</tr>
</tbody>
</table>

Table 10: Binary SVM for ranked data in arousal dimension with slack = 0
SVM accuracies for $2^{11}$ different combinations of input features where a classification is considered true when only one classifier recognizes the input feature as a member, and is matches with the true membership. The top two results for each kernel of this exhaustive analysis is shown in Table 11 and Table 12. The input flags column is a decimal value that represents an 11 digit binary number and represents which features are being used for training. When a binary digit is equal to 1, that corresponding feature is being used as part of the input feature vector, and when it is equal to 0, that corresponding feature is not used.

**Table 11:** Top Results for linear, quadratic and cubic kernel in the pleasure dimension

<table>
<thead>
<tr>
<th>Kernel</th>
<th>Input Flags</th>
<th>Low (%)</th>
<th>Medium (%)</th>
<th>High (%)</th>
<th>Avg (%)</th>
<th>Low (%)</th>
<th>Medium (%)</th>
<th>High (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lin</td>
<td>791</td>
<td>67</td>
<td>60</td>
<td>73</td>
<td>66</td>
<td>78</td>
<td>60</td>
<td>74</td>
</tr>
<tr>
<td>Lin</td>
<td>903</td>
<td>73</td>
<td>47</td>
<td>80</td>
<td>66</td>
<td>84</td>
<td>57</td>
<td>83</td>
</tr>
<tr>
<td>Quad</td>
<td>805</td>
<td>60</td>
<td>60</td>
<td>87</td>
<td>69</td>
<td>93</td>
<td>88</td>
<td>100</td>
</tr>
<tr>
<td>Quad</td>
<td>849</td>
<td>73</td>
<td>53</td>
<td>80</td>
<td>69</td>
<td>89</td>
<td>81</td>
<td>94</td>
</tr>
<tr>
<td>Cubic</td>
<td>1805</td>
<td>73</td>
<td>67</td>
<td>67</td>
<td>72</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Cubic</td>
<td>821</td>
<td>67</td>
<td>73</td>
<td>80</td>
<td>76</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
</tbody>
</table>

**Table 12:** Top Results for linear, quadratic and cubic kernel in the arousal dimension

<table>
<thead>
<tr>
<th>Kernel</th>
<th>Input Flags</th>
<th>Low (%)</th>
<th>Medium (%)</th>
<th>High (%)</th>
<th>Avg (%)</th>
<th>Low (%)</th>
<th>Medium (%)</th>
<th>High (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lin</td>
<td>1889</td>
<td>33</td>
<td>13</td>
<td>60</td>
<td>36</td>
<td>53</td>
<td>8</td>
<td>60</td>
</tr>
<tr>
<td>Lin</td>
<td>734</td>
<td>60</td>
<td>7</td>
<td>53</td>
<td>40</td>
<td>60</td>
<td>2</td>
<td>63</td>
</tr>
<tr>
<td>Quad</td>
<td>1017</td>
<td>73</td>
<td>33</td>
<td>60</td>
<td>55</td>
<td>100</td>
<td>99</td>
<td>100</td>
</tr>
<tr>
<td>Quad</td>
<td>124</td>
<td>73</td>
<td>27</td>
<td>73</td>
<td>58</td>
<td>94</td>
<td>56</td>
<td>84</td>
</tr>
<tr>
<td>Cubic</td>
<td>817</td>
<td>93</td>
<td>33</td>
<td>60</td>
<td>62</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Cubic</td>
<td>957</td>
<td>93</td>
<td>33</td>
<td>60</td>
<td>62</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
</tbody>
</table>
The correlation between the average classification accuracy (High+Medium+Low)/3 and the presence of a particular feature in the input for the SVM was calculated and displayed in Tables 13 and 14 from lowest correlation to highest correlation.

**Table 13**: Correlation between FAPs and average classification accuracy for linear, quadratic and cubic kernels sorted in ascending order in the pleasure dimension

<table>
<thead>
<tr>
<th>Linear Kernel</th>
<th>Quadratic Kernel</th>
<th>Cubic Kernel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correlation Coefficient</td>
<td>FAP Number</td>
<td>Correlation Coefficient</td>
</tr>
<tr>
<td>-0.0764</td>
<td>6</td>
<td>-0.2064</td>
</tr>
<tr>
<td>-0.0642</td>
<td>9</td>
<td>-0.1469</td>
</tr>
<tr>
<td>-0.0196</td>
<td>7</td>
<td>-0.1355</td>
</tr>
<tr>
<td>0.0036</td>
<td>11</td>
<td>-0.1237</td>
</tr>
<tr>
<td>0.0416</td>
<td>2</td>
<td>-0.0917</td>
</tr>
<tr>
<td>0.0675</td>
<td>1</td>
<td>0.0739</td>
</tr>
<tr>
<td>0.1416</td>
<td>4</td>
<td>0.0786</td>
</tr>
<tr>
<td>0.1954</td>
<td>8</td>
<td>0.1386</td>
</tr>
<tr>
<td>0.2268</td>
<td>3</td>
<td>0.1770</td>
</tr>
<tr>
<td>0.3831</td>
<td>5</td>
<td>0.3454</td>
</tr>
<tr>
<td>0.5236</td>
<td>10</td>
<td>0.3603</td>
</tr>
</tbody>
</table>

**Table 14**: Correlation between FAPs and average classification accuracy for linear, quadratic and cubic kernels sorted in ascending order in the arousal dimension

<table>
<thead>
<tr>
<th>Linear Kernel</th>
<th>Quadratic Kernel</th>
<th>Cubic Kernel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correlation Coefficient</td>
<td>FAP Number</td>
<td>Correlation Coefficient</td>
</tr>
<tr>
<td>-0.0814</td>
<td>9</td>
<td>-0.1884</td>
</tr>
<tr>
<td>0.0684</td>
<td>2</td>
<td>-0.1105</td>
</tr>
<tr>
<td>0.0905</td>
<td>6</td>
<td>-0.0367</td>
</tr>
<tr>
<td>0.1026</td>
<td>4</td>
<td>0.0324</td>
</tr>
<tr>
<td>0.1041</td>
<td>3</td>
<td>0.0620</td>
</tr>
<tr>
<td>0.1389</td>
<td>7</td>
<td>0.0775</td>
</tr>
<tr>
<td>0.1671</td>
<td>1</td>
<td>0.1109</td>
</tr>
<tr>
<td>0.1761</td>
<td>8</td>
<td>0.1416</td>
</tr>
<tr>
<td>0.1952</td>
<td>11</td>
<td>0.2232</td>
</tr>
<tr>
<td>0.3725</td>
<td>10</td>
<td>0.3255</td>
</tr>
</tbody>
</table>
5.2.6 Emotion Classification Discussion

Experiment 1: Seven-Class SVM on Rated Data

The results for experiment 1 show that a linear kernel is insufficient in classifying the training images for both the pleasure and arousal domains. Quadratic and cubic kernels, however, are sufficient at classifying 100% of the training images. Also, the quadratic kernel outperforms both the cubic and linear kernels when classifying the test data. It should be noted that the performance for the seven-class SVM has a slack variable of one, that is, the detected value can be at most one class away from the true value to be considered as a correct classification.

Experiment 2: Three-Class SVM on Rated Data

Because the results for the seven-class SVM were not satisfactory (67% is not considered good, especially considering a slack variable that covers 3/7 of the output range), an experiment was run to analyze whether reducing the number of classification groups to three would produce better results. The analysis utilized a slack variable of zero (classifier must output exact match to true value) and showed that although the linear kernel performed better than in the seven-class SVM, it was still insufficient to classify all the training data. The performance of the kernels on the test data remained near the same interval, but the linear kernel performed the best at 61% for the pleasure dimension, and the cubic kernel performed the best at 70% accuracy for the arousal dimension.

Experiment 3: Binary Classification for Extreme Values in Ranked Data

The third experiment attempted to utilize ranked data rather than the numeric rated data in hopes to take advantage of the higher inter-coder reliability values. However, binary classification performed worse than the ranked data, even though less categories and more extreme values were used.
Experiment 4: Three-Class SVM for Ranked Data with Exhaustive Search over FAPs

The fourth experiments investigated the effect that certain features had on the results. The FAPs with the highest correlation values for the arousal domain are 5, 6 and 6, in linear, quadratic and cubic kernel cases respectively. FAP 6 represents a squint factor for the left eye. The more open the eyes are, the lower the value, and the more closed the eyes are, the higher the value. Intuitively, it makes sense that FAP 6 has a higher correlation with arousal than other features, because whether an individual has their eyes shut or wide open causes them to interpret arousal as low or high. Since FAP 6, is highly correlated, it is also expected that FAP 7 is similarly correlated. Though it does not have as high a correlation value, it is still ranked higher than the other features.

The data also shows that several feature points are not correlated with either pleasure or arousal. For example, FAP 2, the slope of the inner eyebrow has a very low or negative correlation with arousal. FAPs 5 and 6, though correlated with arousal, or negatively correlated with pleasure. It is important to note that the analysis for Experiment 4 is limited to only the correlation that an independent feature has with pleasure or arousal. The combination of interaction of features is not analyzed and must be done separately.

Since only polynomial kernels were investigated, it may be possible to obtain a better classifier given a different kernel function. The use of radial basis functions or other functions as kernel for the SVM may yield better results. The fact that the eyes influence the apparent arousal of a facial expression, and the degree of smile has a strong correlation with perceived pleasure can be leveraged to make a stronger classifier.

5.2.7 Summary of Experiments and Results

The overall system procedure is described in Figure 5-15.

The steps completed to develop a system for facial expression recognition were:

1. Images of facial expressions were obtained for labeling of feature points
2. Procrustes analysis and PCA was completed on labeled feature points to calculate the true mean and eigenvectors to obtain the shape model that represents the allowable variations for the feature points.

3. SVM feature detectors were trained for each feature point giving the highest score for better feature candidates.

4. Logistic regression was used to fit the SVM feature detector response to a probability that a particular pixel was a feature.

5. A constrained local model fitting algorithm was created to detect the feature points. It has been shown that the constrained local model converges within four or five iterations.

6. Facial expression images were coded with both numeric values and ranked against each other to provide pleasure and arousal values of emotion for the training images.

7. Multi-class support vector machines were trained in four different experiments to investigate the ability for SVMs to classify facial expression images into a two dimensional model of emotion with pleasure and arousal components. FAPs 5 and 6 are positively correlated with arousal but not correlated with pleasure. FAP 10 is positively correlated with pleasure. FAP 2 is negatively correlated with arousal and FAP is not correlated with either pleasure or arousal.
Figure 5-15: System Overview
Chapter 6

6 Conclusion

This work investigates the use of constrained local models to detect and track human facial feature points in facial expression images. The feature point locations are used to classify the facial expressions into corresponding emotions in terms of degrees of arousal and pleasure using multi-class SVMs.

The work presents a novel method of classifying facial expression images into a pleasure-arousal model of emotion using feature point locations on the face and multi-class SVMs. It has been shown that FAPs can be used with SVMs to classify facial expressions into low, medium and high levels of pleasure and arousal. The FAPs and corresponding classes for pleasure and arousal are not linearly separable and the use of different kernels is required for SVMs to classify facial expressions in this way. It has also been shown that specific FAPs are correlated with degree of pleasure and arousal, and the inclusion of such FAPs result in better classification rates.

Future work involves increasing the accuracy and ability of the constrained local model to detect facial feature points by modifying the training set for the SVMs, modifying the input resolution of the system, or reducing the number of features on the face to more easily defined ones.

Different kernel function can be investigated for use for the multi-class SVMs used in the facial expression recognition stage. Additionally, the investigation of the correlation between different FAP combinations and classification accuracies, as well as the introduction of new FAPs, can augment the knowledge obtained about the individual FAPs determined in this work.
Chapter 7

7 Bibliography


[53] J. Sung and D. Kim, "Real Time Facial Expression Recognition Using STAAM and


   http://www.asc.upenn.edu/usr/krippendorff/webrliability2.pdf

