Person Independent Classification of Facial Expressions Using Multi-Class Support Vector Machines

Abu Sayeed Md. Sohail

A Thesis

in

The Department

of

Computer Science and Software Engineering

Presented in Partial Fulfillment of the Requirements
for the Degree of Master of Computer Science at
Concordia University
Montreal, Quebec, Canada

August 2007

Library and
Archives Canada

Published Heritage
Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et
Archives Canada

Direction du
Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 978-0-494-34603-7
Our file Notre référence
ISBN: 978-0-494-34603-7

ABSTRACT

Person Independent Classification of Facial Expressions Using Multi-Class Support Vector Machines

Abu Sayeed Md. Sohail

This thesis describes a fully automated computer vision system for detection and classification of the seven basic facial expressions using Multi-Class Support Vector Machines (SVM). Facial expressions are communicated by subtle changes in one or more discrete features such as tightening the lips, raising the eyebrows, opening and closing of eyes or certain combination of them, which can be identified through monitoring the changes in muscles movements (Action Units), located around the regions of mouth, eyes and eyebrows. An analytic representation of face with fifteen feature points describing the geometric and physical (muscle) model of facial expression structure has been used that represents and identifies the principal muscle actions and also provides visual observation (sensing) of the discrete features responsible for each of the seven basic human emotions. Feature points from the region of mouth have been detected by segmenting the lip contour applying a newly introduced variational formulation of the existing level set method. In addition, a multi-detector approach of facial feature point detection has been utilized for identifying the points of interest from the regions of eyes, eyebrows and nose. The feature vector composed of fifteen features is then obtained with respect to the average representation of neutral face by calculating the degree of displacement of five different pairs of points, and measuring the deviations of ten points from a non-changeable rigid point. Finally, the obtained feature sets are used to train a Multi-Class SVM classifier. The proposed automated facial expressions classification system has been tested extensively on two publicly available facial expression databases and 92.04% and 86.33% of average successful classification rates have been achieved. Besides, satisfactory results have been obtained by comparing the proposed method with other previous methods of facial expression classification.

Acknowledgements

This thesis and the research that it describes could not be accomplished without the heartfelt guidance and support of multitude of individuals. Many of you had significant influence on me during my study period at Concordia University in a variety of ways, both academic and personal. I express my sincere gratitude to all of you, and would like to single out the contribution of the following people.

First of all, I would like to thank my supervisor Dr. Prabir Bhattacharya for his invaluable guidance, advice, support and criticism since my first arrival at Concordia two years back. It was because of him that my graduate studies were so enjoyable and intellectually rewarding. He has provided a good balance of freedom as well as interest, while teaching me the ways of conducting research. I feel myself privileged for being able to work under his experienced supervision towards accomplishing this research. In addition, I owe my thanks to the chair of my thesis defense Dr. Nematollaah Shiri and the examiners Dr. Ching Y. Suen and Dr. Peter Grogono for their valuable corrections and suggestions, which have improved the quality of this thesis significantly.

My sincere gratitude goes to Dr. Maja Pantic (Department of Computing, Imperial College of London) for clarifying many technical details of her research in the area of facial expression recognition. In addition, I would like thank Dr. Chunming Li (Institute of Imaging Science, Vanderbilt University) for his enlightening discussions on the level set method without re-initialization and also for his help in its implementation as a part of this research. Special appreciation also goes to all of my lab members for their cordial help, ideas and suggestions. Moreover, a hearty thanks to the most important part of my life, my family. If it were not for the caring, love, affection and continuous support of my father, mother and my brother, none of this would be possible at all.

Finally, I would like to thank my wife Fauzia Afrin who has been there for me as a supporter, a friend and a constant source of encouragement when I needed her, and borne all my mood swings due to the frustrations that come with being a researcher.

Dedicated

To My Parents

Table of Contents

List of Figures

List of Tables

List of Acronyms

AAM	Active Appearance Model
AHP	Analytic Hierarchy Process
AU	Action Unit
DAG	Directed Acyclic Graph
DDAG	Decision Directed Acyclic Graph
FACS	Facial Action Coding System
FAP	Facial Action Parameter
FCP	Facial Characteristics Point
k-NN	*k*-Nearest Neighbor
PCA	Principal Component Analysis
RBF	Redial Basis Function
LBP	Local Binary Pattern
LDA	Linear Discriminant Analysis
LP	Linear Programming
PDE	Partial Differential Equations
SVM	Support Vector Machines
SV	Support Vector

Chapter 1

Introduction

The human face is involved in an impressive variety of different activities. It accommodates the physical components for speech production (mouth, tongue and teeth) as well as the majority of our sensory organs: eyes, ears, mouth and nose, allowing us to see, hear, taste and smell. In addition to these biological functions, the human face also provides a number of social signals which are essentials for interpersonal communication in our everyday life and thus mediates person identification, attitudinal or emotional state, lip-reading and interpersonal behavior through perceiving the focus of social attention and facial attractiveness.

During the process of interpersonal communication, we speak and at the same time, we use three other senses – we hear, see and touch or feel. Hence, human communication has two main aspects: verbal and non-verbal. While words can be considered as the automatic information units of verbal communication, phenomena like facial expressions, vocal utterances, body movements and physiological reactions could be considered as the atomic units of non-verbal communication. It is quite clear that non-verbal communicative signals are not necessary for human-human communication; a phone call is an example. Still, considerable research is social psychology has shown that non-verbal communicative cues can be used to synchronize the dialogue, to signal comprehension or disagreement and to let the dialogue run smoothly with less interruption [1], [2]. In addition to their crucial role in non-verbal aspect of human communication, facial expressions also provide information about the observed person's attitudinal state, age, attractiveness, gender, as well as about his/her personality, cognitive activity and psychopathology [3].

Although humans can recognize facial expressions virtually without any error or delay, reliable and fully automated expression recognition by computers is still a challenge. Various approaches have already been attempted towards addressing this problem, but the complexities added by circumstances like inter-personal variation (i.e. gender, race) and inconsistency of acquisition conditions (i.e. lighting, resolution) have made the task quite complicated and challenging. However, the recent advancements in the area of image analysis and pattern recognition have opened up the possibility of automated measurement of facial signals. It is believed that the automated analysis of facial expressions can facilitate machine perception of human facial behavior, and thus opens up the way of bringing facial expressions into man-machine interaction as a new modality towards making the interaction more natural and efficient. It can also enable the classification and quantification of facial expressions widely accessible for the research in behavioral science and medicine by automated psychological observation of humans. Keeping all of these in consideration, this thesis addresses the various complexities related to the modeling, recognition and classification of the encountered facial expressions present in static facial images and thus provides a solution of the problem of classifying the seven important facial expressions namely, neutral, anger, disgust, fear, happiness, sadness and surprise. Six of these expressions except for the "neutral" have been defined as the basic human emotions by Ekman [4]. Instead of using the features from the whole face that can be obtained by the holistic face representation, an analytic representation of the face using a total of fifteen feature points from the regions of eyebrows, eyes, nose and mouth has been used to avoid the "curse of dimensionality" [60] of the extracted facial expression features. This analytic face representation describes the geometric and the physical (muscle) model of facial expression structure and thus identifies the principle muscle actions towards providing the visual observation (sensing) of the discrete features responsible for each of the seven basic human emotions. A multi-detector approach based on the standard image processing techniques of facial feature point localization has been utilized for identifying the eleven points of interest from the regions of eyes, eyebrows, and nose. For detecting the rest of the four points from the region of mouth, isolation of lip contour has been performed implementing a variational formulation of the level set method proposed by Li et al. [5]. This variational formulation forces the level set function to be close to a signed distance function and therefore completely eliminates the costly re-initialization procedure, which is considered as a severe drawback of the existing level set method. The feature vector composed of fifteen features is then obtained with respect to the average neutral face representation by calculating the degree of displacement of five different pairs of points, and measuring the deviations of ten points from a non-changeable rigid point. Finally, all the feature vectors, calculated from the training samples, are used to train a Multi-Class Support Vector Machines (SVM) classifier so that it can classify the facial expression of an unknown input image when given to it in the form of a feature vector.

1.1. Motivation

Recent technological advances have enabled human users to interact with computers in ways that were previously unimaginable. Beyond the confinement of keyboard and mouse, new modalities for human-computer interaction such as voice, gesture, and force-feedback are emerging. Despite these important advancements, one necessary ingredient for natural interaction is still missing: the facial expressions in the form of emotions [6]. Facial expression is a visible manifestation of the affective state, cognitive activity, intention, personality and psychopathology of a person that plays a communicative role in interpersonal relations as well as in human-to-human communication and interaction, allowing people to express themselves beyond the verbal domain [7]. Effectiveness of facial expressions as a non-verbal medium of communication has already been established by different researches. According to Mehrabian [8], the verbal part of a message contributes only for 7% to the effect of the message as a whole; the vocal part contributes for 38%, while facial expressions of the speaker contribute for 55% to the effect of the spoken message. Research conducted by van Poecke [9] indicates that 35% to 40% of the overall meaning of a communicated message is transmitted verbally and 60% to 65% is transmitted non-verbally. This implies that facial expressions form the major modality in human communication and can play an important role wherever humans interact with machines. So, the automated classification of facial expressions may act as a component of both natural human-machine interface and its variation known as the perceptual interface [10], [11]. Since the aggregation of the emotional information with human-computer interfaces allows much more natural and efficient interaction paradigms to be established, development of a system for the automated classification of facial expressions can play an increasing role in building effective and intelligent multimodal interfaces for next generation. This can also be a possible application domain for a diverse of disciplines including behavioral science, medicine, monitoring, communications, education, face modeling as well as face animation.

1.2. Related Work

Due to its importance for application domains in human behavior interpretation and human-computer interface, automated analysis and classification of facial expressions has attracted the interest of many researchers. The starting history of facial expression analysis research goes well back into the nineteenth century when Darwin [12] demonstrated in 1872 the universality of facial expressions and their continuity in man and animals; and claimed among other things that there are specific inborn emotions, which are originated in serviceable associated habits. Then in 1971, Ekman and Friesen [4], [13] postulated six primary emotions that process each a distinctive content together with a unique facial expression. These prototypic emotional displays are also referred to as the so called basic emotions which seem to be universal across human ethnicities and cultures and comprise anger, disgust, fear, happiness, sadness and surprise. Although the analysis of facial expressions was primarily considered to be a research subject for the psychologists in the past, it began to be amalgamated with the area of computer vision when Suwa et al. [14] presented their preliminary investigations on the analysis of facial expressions from an image sequence in 1978 [15]. Then with the pioneering work of Mase and Pentland [16], research on the automated analysis of facial expressions gained significant momentum. Different approaches that have been proposed in this regard since the mid of 1970s can be grouped into the following three categories on the basis of the face representation method used [3], [17]:

i. Holistic face representation based approach
ii. Approach based on the analytic representation of face
iii. Hybrid approach

1.2.1. Holistic Face Representation Based Approach

In holistic face representation based approaches [18-24], [26-27], the whole face or major parts of the face are processed in order to obtain visual features of facial expressions. These are also known as the template-based approaches where the template can be a 2-D array of intensity values, a labeled graph or some other type of templates that can describe the properties of the face as a whole [3].

Cottrell and Metcalfe [18] have used features from the whole face which they termed as "holons". The "holons" are in fact manually selected facial regions normalized and reduced to 64×64 pixels. A three layer back-propagation neural network has been used in their method for classifying these "holons" into one of the eight different facial expressions (angry, miserable, bored, relaxed, sleepy, pleased, happy and astonished). Rahardja et al. [19] have also used a holistic data representation approach similar to the one used by Cottrell and Metcalfe [18] for classifying the hand-drawn faces into six different facial expressions (happy, sad, surprised, angry, afraid and neutral). A pyramid-like feed-forward neural network, trained with the features extracted through modeling the concept of hierarchical (multi-resolution) representation of the image data, has been used in their work for classifying the facial expressions of the input images. Their proposed system can recognize facial expressions successfully from the images of the training data set but the performance achieved using the system over the unknown data set has not been evaluated. Moreover, a relatively poor recognition rate has been achieved using their system in classifying facial expressions from the blurred or distorted images. Another neural-network based approach of classifying facial expressions into one of the six basic emotion categories [4] has been proposed by Vanger et al. [20]. By manual procedures, they have averaged all eye and mouth parts of the 60 utilized static images containing the six basic prototypic expressions and created a prototype index for each emotion category. To classify the facial expression from an unknown input image, a synergetic method of matching the eye and mouth parts of the input image to the prototype index has been used in their work. The claimed success rate for their proposed system is 70%. A different holistic face model known as "Potential Net" has been deployed by Matsuno et al. [21] for the recognition of four different facial expressions (happiness, anger, surprise and sadness). A Potential Net consists of nodes, each of which is connected to four neighbor nodes through springs, and is positioned on the rectangular facial area extracted manually from a normalized static face image. Deformation of the Potential Net is achieved by the forces based on the smoothed gray-level value of the edge image so that each node is moved to the position of facial features such as eyebrows, mouth, and wrinkles. The displacement vector, obtained by measuring the movement of each node, has been used for the classification of facial expressions by analyzing the similarity between the vector of the input image and the vectors of the four different facial expressions. The proposed system has been tested with 44 unknown facial expression images taken from eleven subjects and successful recognition rate of 100% in the case of surprise, 93% for happiness 90% for anger and 70% in the case of sadness have been achieved respectively. Padgett and Cottrell [22] have used random block eigenvectors defined over 97 images taken from Ekman's database [4]. Classification of facial expressions has been performed in their work using a 15×10×7 back-propagation neural network with a success rate of 86%. Black and Yacoob [23] have used a local parameterized model of the image motion to separate and to recognize the non-rigid facial expression from the rigid head motion. Their high-level recognition approach was based on the mid-level index of the motion direction of each facial feature region (brows, eyes and mouth) that were selected manually. The mid-level representation was predicted, however, by taking the difference of the motion parameter estimation and a threshold value. Furthermore, different threshold were used for different motion parameters in the experiment: some were between $0.5 \sim -0.5$, and others were between $0.00005 \sim -0.00005$ [23]. The thresholding of motion parameters would filter out some subtle motion. As a result, this thresholding method for motion parameters, in effect, has reduced the reliability and accuracy of the recognition. In their studies of recognition of six basic facial expressions, an average recognition rate of 92% was achieved with 70 image sequences of 40 subjects. Multivariate multiple regression has been applied by Edwards et al. [24] for representing the facial expressions by modeling the relationship between the Active Appearance Model (AAM) [25] displacement and the image difference using manually localized 122 facial points. The classification of facial expressions has been performed in their work using the Mahalonobis distance based PCA and Linear Discriminant Analysis (LDA) with a reported successful classification rate of 74%. Feng et al. [26] have proposed a technique where Local Binary Pattern (LBP) has been used to represent a facial image for facial expressions through the manual selection of the eye positions. In the classification step of their method, seven expressions (anger, disgust, fear, happiness, sadness, surprise and neutral) have been decomposed into 21 expression pairs like anger-fear, happiness-sadness, etc. Then, 21 classifiers each corresponding to one of the 21 expression pairs has been created applying the Linear Programming (LP) technique. A simple binary tree tournament scheme with pair-wise comparisons, generated with these classifiers, has been used for classifying unknown expressions. The reported highest, lowest and average recognition rates of the proposed method are 95.6%, 93.4% and 94.6% respectively. Yu and Bhanu [27] have proposed a genetically inspired learning method for the recognition of facial expressions by introducing the Gabor wavelet representation of the primitive features and the linear/nonlinear operators to synthesize new features. The use of automated feature selection process of and the application of Support Vector Machine towards the classification of facial expressions are considered as the significant advancements of their method. However, a comparatively lower successful recognition rate of 80.95%, as reported for this method, appears to be a considerable drawback.

1.2.2. Approach Based on the Analytic Representation of Face

In approaches based on analytic (feature-based) face representation [28-33], [35], [37-40], [42], some facial points or contours of the prominent facial features (eyes, eyebrows and mouth) are used to model the face. The relative size and shape of the model features and the relative distances in between are then used for classification and recognition of facial expressions [3].

Kobayashi and Hara [28] have used 30 facial characteristics points hand-measured in facial images as input to their 60×100×100×6 back-propagation neural network for training as well as classification of the facial expressions into one of the six basic emotion categories. Correct classification rate of 90% has been achieved by their trained network. A similar approach based on the displacement of the manually selected facial characteristics points has been performed by Ding et al. [29]. They have used three sets of artificial neural network to recognize brows, eyes and mouth expressions only from the left half of the face assuming symmetrical change in both side of the face due to different facial expressions. Zhao and Kearney [30] have performed the classification of facial expressions from Ekman's 94 photos [4] using a 10×10×3 back-propagation neural network. Frontal-view and a point-based model of ten manually localized facial distances have been used as features in their work. Hara et al. [31] have applied features obtained from 30 facial characteristics points and thirteen vertical lines to train their 234×50×6 back-propagation neural network for classifying facial expressions into multiple categories and achieved a successful recognition rate of 85%. To classify facial expressions under one of the emotion categories: angry, happy, and sad, Ushida et al. [32] have deployed a multi-layered structure of bi-directional associative neural network along with fuzzy logic using the features extracted from the hand measured FCPs introduced by Kobayashi and Hara [28]. To reduce the quantity of input data, they took the advantage of the face symmetry and used the FCPs belonging to the eyebrows, the right eye and the mouth. A shortcoming of this is that their method is not sensitive to the unilateral appearance changes of the left eye. The reported correct recognition rate of classifying facial expressions using their proposed method is 79%. Kearney and McKenzie [33] have developed an expert system for the classification of facial expressions into one or more of the emotion categories defined by human observers. Their system converts manually measured facial landmarks into an intermediate facial-action-based representation, which is further interpreted in terms of the defined emotion categories by a dynamic memory. The memory is dynamic in the sense that the new emotion categories can be learned with experience. The production rules used for facial action coding are based on the rules defined for FAST (i.e. an early version of FACS [34]). Validation studies demonstrated that the facial action encoding achieved by the system in 90% of the cases is consistent with that of the human experts.

Those studies also reported a correct classification ratio of 91.78% for the six basic emotion categories and 91.21% for the learned categories. Cohn et al. [35] have used the hierarchical optical flow algorithm introduced by Lucas and Kanade [36] for estimating the optical flow in 13×13 pixel of facial regions. Their system requires manual initialization of 45 facial points and classification of the facial expression has been performed applying the discriminant function with a successful recognition rate of 88%. However, a weakness of their method is that the first frame must contain an expressionless face. Pantic and Rothkrantz [37] has performed rule-based classification of facial expressions by extracting features through geometric measurements among the landmarks located on the contours of eyebrows, eyes, nostrils, mouth and chin. Their method requires manual intervention in detecting the mouth boundary applying the active contour model proposed by Kass et al. [38] with a greedy algorithm based energy minimization of the snake. The achieved successful recognition rate of their system is 91%. Michel and Kaliouby [39] have applied Support Vector Machines for classifying the facial expressions using a feature displacement approach with 22 facial feature points and have achieved an average successful classification rate 86% on still images. However, details of the tracking method used for detecting the 22 facial feature points have not been reported. Gabor filter based feature calculation from the 34 manually selected landmark points has been introduced by Guo and Dyer [40] for extracting the features of six basic facial expressions. They have used a pair-wise framework for feature selection in training a range of classifiers including simplified Bayes classifier, Support Vector Machine and AdaBoost. The maximum recognition rate of 92.4% has been reported by their method using SVM with liner kernel for classifying facial expressions. The distance weighted *k*-Nearest Neighbor rule [41] has been applied by Sohail and Bhattacharya [42] for classification of facial expressions using a spatio-temporal representation of face with eleven feature points. Detection of these eleven feature points, used in their work for the analytic representation of face, has been performed in an automated manner applying the multi-detector approach of facial feature point localization [43]. The reported correct recognition rate for their proposed method is 91%.

1.2.3. Hybrid Approach

The hybrid approaches [16], [44], [46-50], [52], [54-55] perform face representation by combining the feature-based approach and the template-based approach. Usually, a set of facial feature points is used in these approaches for determining the initial position of a template that models the face. The template can be a 3-D wire frame, a labeled graph or a Potential Net [3].

Yoneyama et al. [44] have applied a Gradient-based optical flow algorithm for estimating an averaged optical flow in 80 blocks of 20×20 pixel regions within the grid placed over a normalized face image. Two 14×14 Hopfield NNs with Personnaz learning [45] have been used in their work for classifying the facial expressions and 92% of correct recognition rate has been achieved over 40 images taken from 10 subjects. Essa and Pentland [46], [47] have proposed two methods for recognizing facial expressions. The first method classifies five expressions: smile, surprise, raised brow, anger, and disgust by scoring the dot-product similarity based on the 36 peak muscle actuations in comparison to the standard training expression templates, but the temporal affect is ignored. The second method uses the temporal-template matching for two dimensional gray value images. Here, the time warping is an important consideration which improves the recognition accuracy since the temporal-template matching measures the correlation between the testing and the standard template image sequences. The overall recognition rate was 97.8% for their method on 6 subjects with 23 and 48 image sequences for training and testing respectively. Optical flow in two dimensions has been used by Mase and Pentland [16] to classify facial expressions by tracking and computing the motion of facial muscles. The Muscle regions were manually selected by referring to major feature points over the face. Optical flow was computed to extract 12 of the 44 facial muscle movements, which were interpreted as appropriate AUs in combination with the position of feature. Their approach relies heavily on accurate tracking of the manually selected muscle regions. Flow directions within each individual region were averaged to represent the flow direction of that region. However, when the selected area corresponds to a smooth, featureless surface in the face, the optical flow estimation will be unreliable, leading to tracking error. Manual selection of some muscle regions may be difficult since they are small and highly movable. In essence, Mase and Pentland built a model that was appropriate for synthesizing facial expressions but remains uncertain in analyzing facial expressions. They computed mean and covariance of the optical flow in each local region, and then, based on the highest ratio of between-class to within-class variability to classify various expressions; the *k*-

Nearest Neighbor rule was applied for recognition. Their experiments indicated an accuracy of approximately 86% in recognizing five expressions (happiness, anger, surprise, disgust, and unknown) over a single subject with 20 and 30 training and testing image sequences respectively. The work of Yacoob and Davis [48], and Rosenblum et al. [49] are related closely to that of the Mase and Pentland's [16] since both of them have used optical flow to track the motion of the surface regions of facial features: brows, eyes, nose and mouth, but not that of the underlying muscle groups. In each facial feature region, the flow magnitude was thresholded to reduce the effect of small computed motions which may be either produced from the texture-less parts or from the affect of illumination. The overall flow direction of each region is to conform to the plurality among the neighborhoods. The direction of any flow in this region is quantized to one of the eight main directions for providing a mid-level representation (to match with the dictionary or lookup table of the motion direction for each region of the basic facial action) so as to permit the high-level classification of facial expressions. Yacoob and Davis [48] have used this mid-level representation to classify the six basic facial expressions as well as eye blinking. The recognition rate for their method was 88% (except eye blinking for which it was 65%) among 32 subjects with 46 image sequences. Rosenblum et al. [49] have extended the work of Yacoob and Davis's [48] based on the similar mid-level representation to recognize the facial expressions of smile and surprise using an artificial neural network with radial basis function (RBF). The recognition rate achieved in their method is 88% for the images of 32 subjects. Wang et al. [50] have introduced averaged B-Splines of feature trajectories [51] for distance minimization in classifying facial expressions. Features of facial expressions, used in their work, have been obtained from a labeled graph of 19 nodes pointed out manually over the face. They have reported an accurate recognition rate of 95% using the system over 29 image sequences collected from eight subjects. A similar labeled graph based approach with 34 manually selected nodes over the face has been applied by Zhang et al. [52] for extracting the features of facial expressions. The classification of facial expressions has been performed in their work with a 646×7×7 back-propagation neural network trained using the RPROP learning algorithm [53] with a successful recognition rate of 90%. Lyons et al. [54] have adopted PCA and LDA of the labeled graph vectors for classifying the facial expressions where 34 nodes of the labeled graph were pointed out manually over the face. Correct recognition of 92% has been achieved using their system over 213 images of ten Japanese females. Hybridization of the high-level semantic concept and low-level features has been performed by Cheng et al. [55] using a semantic-based learning algorithm along with the analytical hierarchy process (AHP) [56]. Applying their method over 213 images of 10 Japanese females using the weight assigned semantic information supported *k*-NN classifier; they have been able to classify the five facial expressions (neutral, happiness, anger, sadness and surprise) successfully in 85.2% cases.

A summary of the selected important methods, proposed earlier for the classification of facial expressions, has been provided in Table 1.

Table 1: Summary of the methods (selected) proposed earlier by different researchers for the classification of facial expressions.

Method Proposed by	Technique of Classification Used	Test Cases	Accuracy	Comment
Vanger et al. [20]	Neural Network	60 images	70%	Created a prototypic index of each of the six basic emotion categories by manual procedures averaging all eye and mouth parts of the 60 utilized static images.
Matsuno et al. [21]	Similarity measurement between the vector of the input image and the vectors of the four facial expressions	44 images taken from 11 subjects	88.25%	Manual selection of the rectangular facial areas for feature extraction from a normalized static face image. Can classify only four different facial expressions (happy, angry, surprised and sad).
Padgett and Cottrell [22]	15×10×7 back-propagation neural network	84 Ekman's photos [7]	86%	Strictly constrained to image format. Test cases are not real-life shots and applicability in real-life situations is not provided.
Black and Yacoob [23]	Rule based method applied to local parameterized model of the image motion to separate and to recognize the non-rigid facial expressions	70 image sequences of 40 subjects	92%	The initial regions of the head and features (brows, eyes and mouth) have to be selected by hand. Use of the thresholding method for motion parameters, in effect, has reduced the reliability and accuracy of the recognition by filtering out subtle motion information.
Edwards et al. [24]	Mahalonobis distance based PCA and LDA	200 images of 25 subjects	74%	Manual localization of the 122 feature points used for feature extraction.
Feng et al. [26]	A simple binary tree tournament scheme consists of 21 classifiers generated applying the LP technique	213 images of 10 Japanese females	94.6%	Centers of the left and right eye are needed to be pointed out manually during the initial phase of feature extraction.
Yu and Bhanu [27]	Automated feature selection based Support Vector Machine	213 images of 10 Japanese females	80.95%	Genetically inspired automated process for feature selection. Relatively lower recognition rate in comparison to the other methods tested on the same database.
Kobayashi and Hara [28]	60×100×100×6 back-propagation neural network	19 Japanese students	90%	Use of 30 facial characteristics points hand-measured over the facial images.

Hara et al. [31]	234×50×6 back-propagation neural network	90 image sequences of 15 subjects	85%	Manual intervention is required for detecting the 30 FCPs and thirteen vertical lines. Changes in horizontal facial appearance are not modeled.
Cohn et al. [35]	Discriminant functions	504 image sequences of 100 subjects	88%	The first frame must contain an expressionless "neutral" face. Manual normalization has to be done by hand labeling of the first frame.
Pantic and Rothkrantz [37]	Expert System Rules	456 dual views of 8 subjects	91%	Classify facial expressions into six basic emotion categories. Can recognize facial expressions from dual view of the face images. Multiple quantified classifications.
Michel and Kaliouby [39]	Support Vector Machine	Total 72 test samples (12 for each of the 6 basic emotion)	86%	Applied a tracker that uses a face template to initially locate the position of the 22 facial feature points. Details of the tracking method are not provided.
Guo and Dyer [40]	Support Vector Machine with linear kernel	213 images of 10 Japanese females	92.4%	Manual localization of the 34 facial points used for extracting features in the form of Gabor filter coefficients.
Sohail and Bhattacharya [42]	Distance weighted *k*-Nearest Neighbor Rule [41]	213 images of 10 Japanese females	91%	Automated system for the classification of facial expressions by detecting eleven feature points. Cannot handle the "neutral" face.
Yoneyama et al. [44]	Two 14×14 Hopfield NNs with Personnaz learning [45]	40 images of 10 subjects	92%	Not tested on unknown subjects. Averaging of the flow is considered as a drawback.
Essa and Pentland [47]	Spatio-temporal matching of the motion-energy templates	30 sequences of 8 subjects	97.8%	2-D Spatio-temporal representation of the frontal facial view. Can recognize five expressions (smile, surprise, anger, and disgust)
Mase and Pentland [16]	*k*-NN rule based on the highest ratio of between-class to within-class variability	Single subject with 20 training and 30 and testing seq.	86%	The muscle regions were manually selected by referring to major feature points over the face. Can recognize only five expressions (happiness, anger, surprise, disgust, and unknown)
Yacoob and Davis [48]	Rule based systems for classifying basic action cues	46 image sequences of 32 subjects	88%	Used mid-level representation of facial dynamics for capturing the basic action cues. First frame must contain an expressionless face.

Wang et al. [50]	Averaged B-Splines of feature trajectories for distance minimization [51]	29 image sequences of 8 subjects	95%	Hand labeling of the first frame and manual localization of the 19 facial feature points used for feature extraction.
Zhang et al. [52]	646×7×7 Neural Network with resilient RPROP learning [53]	213 images of 10 Japanese females	90%	Manual localization of the 34 facial points. Image format strictly constrained.
Lyons et. al [54]	PCA and LDA of the labeled graph vectors	213 images of 10 Japanese females	92%	Manual localization of 34 facial points. Strictly constraint to the image format.
Cheng et al. [55]	*k*-NN classifier supported by the weight assigned semantic information	213 images of 10 Japanese females	85.2%	Can only classify facial expressions into one of five emotion categories (neutral, happiness, anger, sadness and surprise).

1.3. Objective of the Research

As can be observed from the discussion on the related works of section 1.2 and its summarization provided in Table 1, most of the previous works suffer from the drawback of manually initializing the landmark points or the facial feature regions used for representing the facial expressions as well as feature extraction. So, the principle objective of our research would be to make an attempt towards eliminating this problem by the automated detection of the landmark points used for the analytic representation of face for facial expressions classification. A small subset of the previously used landmark points would be utilized in this work that demonstrates comparatively less tendency of being rejected during the detection process and thus expected to contribute significantly in improving the performance of the proposed method. Since, handling of "neutral" face has not been carried out in most of the previous works; a new approach of feature extraction using the average neutral face representation would be incorporated in the proposed method so that facial expressions of the input images can be classified into seven emotion categories (neutral, anger, disgust, fear, happiness, sadness and surprise) instead of avoiding the "neutral" expression.

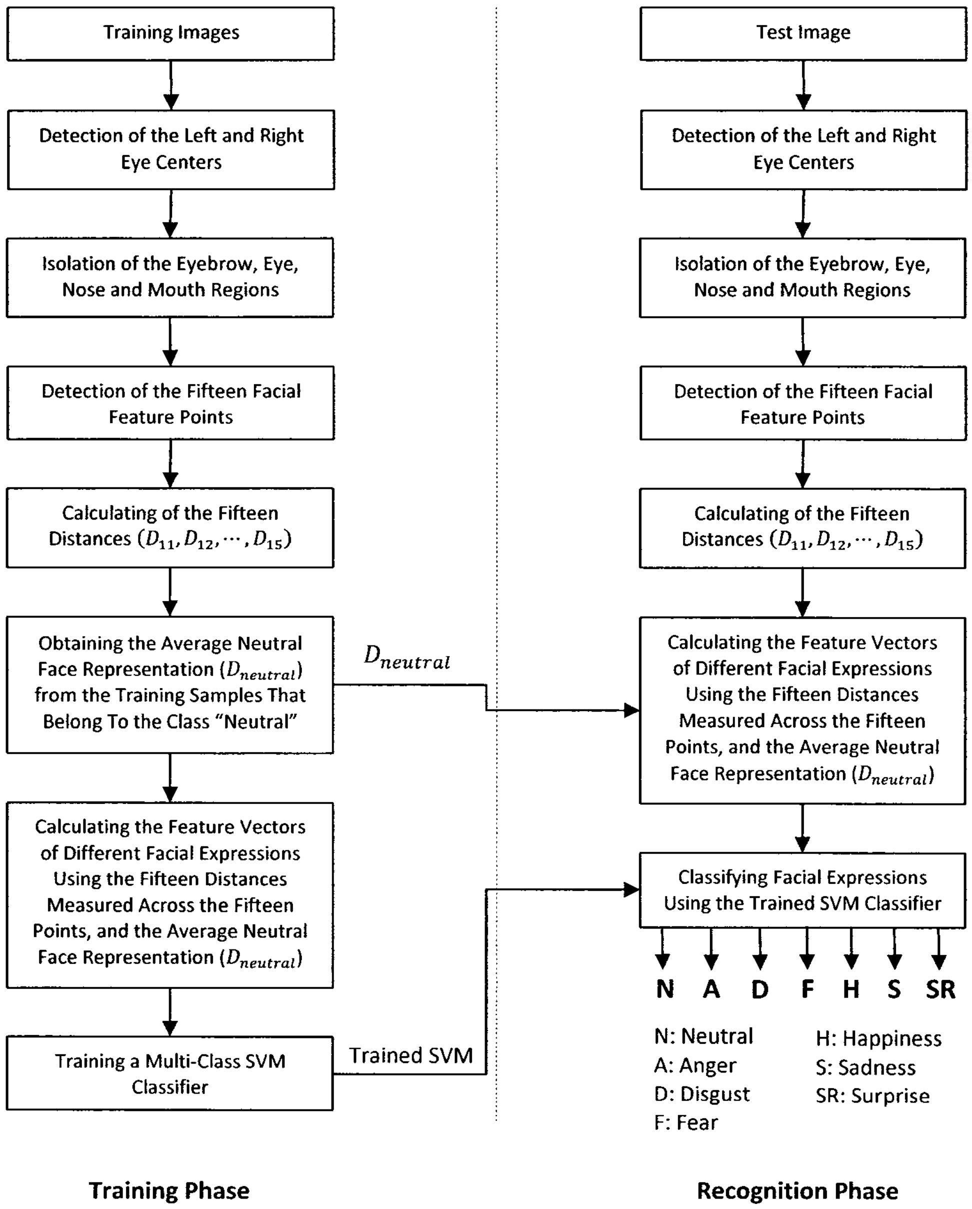

Training Images
Detection of the Left and Right Eye Centers
Isolation of the Eyebrow, Eye, Nose and Mouth Regions
Detection of the Fifteen Facial Feature Points
Calculating of the Fifteen Distances $(D_{11}, D_{12}, \cdots, D_{15})$
Obtaining the Average Neutral Face Representation $(D_{neutral})$ from the Training Samples That Belong To the Class "Neutral"
Calculating the Feature Vectors of Different Facial Expressions Using the Fifteen Distances Measured Across the Fifteen Points, and the Average Neutral Face Representation $(D_{neutral})$
Training a Multi-Class SVM Classifier
$D_{neutral}$
Trained SVM
Test Image
Detection of the Left and Right Eye Centers
Isolation of the Eyebrow, Eye, Nose and Mouth Regions
Detection of the Fifteen Facial Feature Points
Calculating of the Fifteen Distances $(D_{11}, D_{12}, \cdots, D_{15})$
Calculating the Feature Vectors of Different Facial Expressions Using the Fifteen Distances Measured Across the Fifteen Points, and the Average Neutral Face Representation $(D_{neutral})$
Classifying Facial Expressions Using the Trained SVM Classifier
N A D F H S SR
N: Neutral
A: Anger
D: Disgust
F: Fear
H: Happiness
S: Sadness
SR: Surprise
Training Phase
Recognition Phase

Figure 1: A complete block diagram of the proposed automated facial expression classification system including both the training and recognition phases.

As Support Vector Machines (SVM) have already gained much attention among the computer vision researchers for pattern classification, Multi-Class SVM would be deployed in this work for the purpose of classification, since it has been found to be rarely used in the application of facial expression recognition. Beside this, the robustness of the proposed method would be examined by performing experiments with two different facial expression databases that incorporates sufficient variation in the input patterns to reflect the real-life situations. A complete block diagram of the proposed automated system for facial expressions classification has been provided in Figure 1.

1.4. Principal Contributions

The following principal contributions have been made towards achieving the objectives of the research described in this thesis:

- A new model for the analytic representation of face towards capturing the features of different facial expressions has been proposed using carefully selected fifteen feature points that demonstrates comparatively less tendency of being rejected during their detection process.

- A way of handling the "neutral" face has been introduced for classifying the facial expressions into one of the seven emotion categories (neutral, anger, disgust, fear, happiness, sadness and surprise) instead of six (anger, disgust, fear, happiness, sadness and surprise) as done by most of the previous works.

- A new anthropometric face model based technique has been deployed for isolating the regions of facial features [57]. Detection of the eleven future points from the region of eyes, eyebrows and nose has been performed implementing a standard image processing based multi-detector approach of facial feature point localization.

- For detecting the rest of the four points from the mouth region, isolation of the lip contour has been carried out implementing the level set method of image segmentation [58]. Since, the traditional level set method suffers from the drawback of costly re-initialization procedure, a newly introduced variational formulation known as the level set method without re-initialization [5] has been implemented for isolating the lip contour towards detecting the four feature points from the mouth region.

- Multi-Class SVM classifier, constructed by combining 21 two-class SVM using the Decision Directed Acyclic Graph (DDAG) approach [59], have been implemented to achieve a comparatively better recognition rate in classifying the facial expressions. Further efficiency of classification has been achieved by combining several trained Multi-Class SVM classifier applying the Multiexpert method of classifier combining using the *Voting* Scheme [60].

- Performance of the proposed automated facial expressions classification method has been tested over two different facial expressions databases instead of confining the performance analysis process on a single database, as done in previous work.

- The developed system has been compared with three other classification methods namely, *k*-NN, Neural Network and Naive Bayes Classifier in terms of recognition accuracy with a view of establishing the supremacy of SVM in classifying facial expressions. Comparison with some other previously proposed method has also been documented.

1.5. Organization of the Thesis

The entire thesis deals with the development technique of a fully automated computer vision system for the detection and classification of facial expressions. In this relation, a brief introduction followed by the motivation behind the research, necessary discussion on the related pervious works, objective of the research and the summary of principal contributions have already been provided in this chapter. The subsequent discussion of this thesis has been organized into the following chapters:

A brief overview of the proposed automated facial expressions classification system along with necessary theoretical discussions has been provided in **Chapter 2**. This chapter also explains the deployed fifteen feature point based analytic face representation model, used for capturing the features of facial expressions. **Chapter 3** elaborates the method of isolating the facial feature regions (eyes, eyebrows, nose and mouth) applying an anthropometric face model based technique [57]. The theoretical description of the methods used for detecting the face and the eye centers from face images as a part of the complete framework have been provided in this chapter. **Chapter 4** explains the standard image processing based multi-detector approach of detecting the eleven feature points from the regions of eyes, eyebrows and nose [43]. It also includes the technique of detecting the four feature points from the mouth region by segmenting the lip contour applying a variational formulation of the level set method based image segmentation [58]. **Chapter 5** elaborates the theoretical background of deploying the Multi-Class Support Vector Machines as the classifier for classifying facial expressions. Besides, technique of tuning the SVM parameters using the k-fold cross validation procedure has also been discussed in this section. Experimental results, obtained by deploying the proposed automated facial expression system over two different publicly available facial expressions database have been accommodated in **Chapter 6**. Comparison of the proposed method with other classification techniques (*k*-Nearest Neighbor, Neural Network, and Naive Bayes classifier) and some other previously proposed methods of facial expression classification has also been included in this chapter. Finally, **Chapter 7** concludes the thesis with necessary discussions on the work that has been carried out during this research, and provides some suggestion related to its possible future extension.

Chapter 2

Features for Classifying Facial Expressions

An important fundamental issue in the recognition of prototypic emotional facial expressions is classification of the visual information that the examined faces might reveal [61]. According to Yamada [62], this process of classification has two stages. The first stage is to define the set of categories we have to deal with in classifying the expressions, and the second stage is to define the mechanism for classification. Most psychological studies on facial expression classification have been related to the first step. Probably the most known and the most commonly used study on the classification of facial expressions is the cross-cultural study on the existence of universal categories of emotional expressions [4], [63-65]. Ekman defined six such categories, known as the six basic emotions namely, happiness, sadness, surprise, fear, anger, disgust; and described each of these basic emotions in terms of a facial expression that universally and uniquely characterizes that emotion [4]. Although some psychologists, such as Russell [66], have doubted the universality of the six basic emotions, most of the researches of vision-based facial gesture analysis expressions [18, 19, 20, 26, 27, 28, 33, 37, 39, 40, 42, 48] rely on Ekman's emotional categorization of facial expressions. The method proposed in this thesis, which performs the emotional classification of facial expressions, is also based on the Ekman's description of the six basic emotions with a little exception that the absence of any of these six basic expressions, that is the neutral face, would also be considered in this work.

In contrast to the first stage of the facial expressions classification process, there have been rather few studies on the second stage. So, what kind of information from the face should be used in order to classify a certain facial expression into a particular emotional category is still an open question [40], [61]. Probably the best known study on this subject is the Facial Action Coding System (FACS) [67] that explains how to categorize facial behaviors based on the muscles that produce them (i.e. how muscular actions are related to facial appearances).

2.1. Anatomical Analysis of Facial Expressions

From the anatomical viewpoint of human face, there are total 26 muscles that are responsible for all the possible movements within the face [68]. However, according to Faigin [68], only eleven of these muscles are responsible for different facial expressions. A pictorial outline of these muscles is given in Figure 2, and their underlying functionalities are provided as follows:

1. Orbicularis oculi

This muscle is attached to the inner orbit of the eye socket and to the skin of the cheek. It is used for squinting by contracting the eyes.

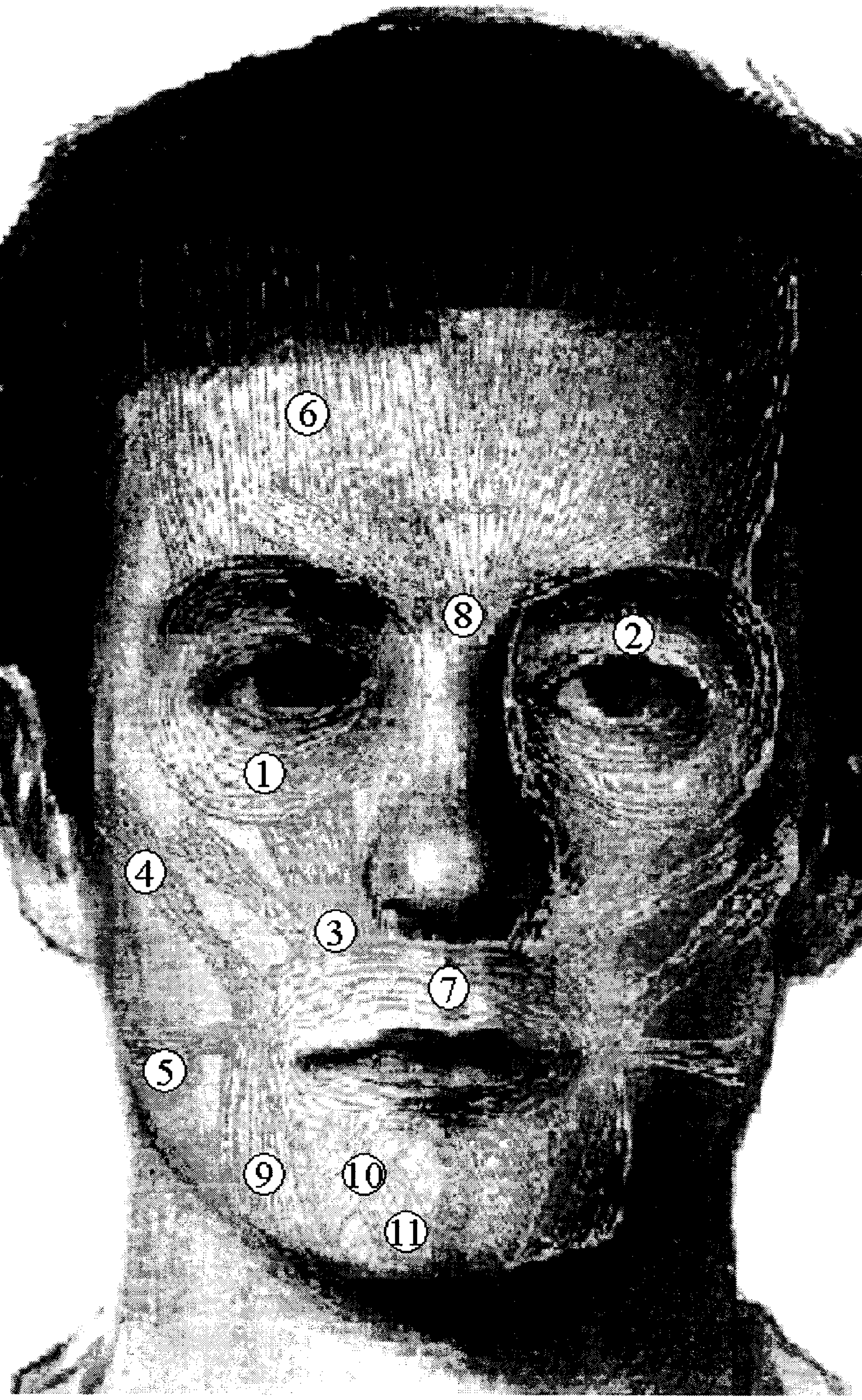


Figure 2: The eleven most influential muscles of human face that are responsible in the formation of different facial expressions (extracted from [68]).

2. Levator palpebrae

The levator palpebrae is attached to the upper eyelid. It is used to raise the eyelid, for example, to display the surprise expression over the face.

3. Levator labii superioris

This muscle is sub-divided into three branches: the inner branch originates at the base of the nose, the middle branch starts at the bottom edge of the orbicularis oris, and the outer branch is connected to the zygomatic arch (also known as the "cheek" bone, which is the arch of the bone on the side of the skull). The muscle Levator labii superioris is used for sneering.

4. Zygomatic major

The zygomatic major is positioned on the top of the zygomatic arch, and is used in the formation of smile.

5. Risorius/platysma

The platysma is used in conjunction with the risorius to stretch the mouth, as in crying. The risorius is positioned over the rear side of the jaw.

6. Frontalis

This muscle is found near the top of the skull and continues under the both eyebrows. It is used in the formation of the surprise expression. The frontalis is also known as the "brow lifter".

7. Orbicularis oris

The orbicularis oris lies at the corner of the mouth and is used primarily for the tightening of the lips. This muscle is also used to curl the lips.

8. Corrugator

The corrugator is found on the nasal bridge. It is attached to the skin between the eyebrows and is used to lower the inner ends of the eyebrows. For this reason, corrugator is also known as the frowning muscle.

9. Triangularis

Triangularis is located along the lower margin of the jaw and the corner of the mouth. It is one of the most significant muscles in generating the expression of sadness and thus known by some anatomists as the "have a bad day" muscle.

10. Depressor labii inferioris

This depressor labii inferioris is found at the bottom of the chin and at the lower lip. This muscle is used to pull the bottom of the lip down while speaking.

11. Mentalis

The mentalis is used for pouting. It stems from just the below of the teeth on the lower jaw and concludes at the ball of the chin.

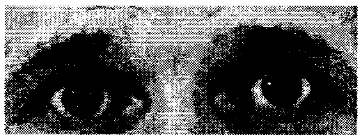
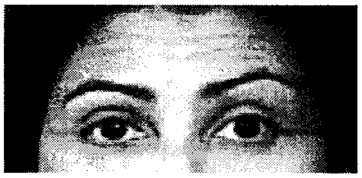
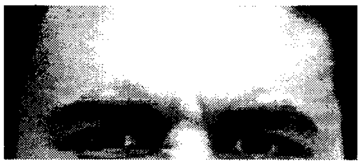
Although the description of Faigin [68] provides a good basis for understanding the anatomy of facial expressions, it does not provide a clear insight into which muscles work together to create a certain expression. For a more comprehensive analysis of facial expressions, it is necessary to look at the combined muscles rather than solitary elements to fully understand how each expression is generated. Such a linguistic abstraction of facial expressions has been provided by the Facial Action Coding System [67] that describes the facial expressions in terms of the codes of facial muscles involved in each expression.

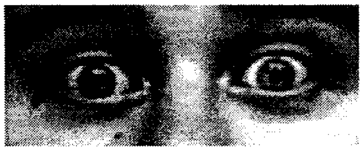
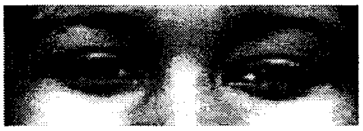
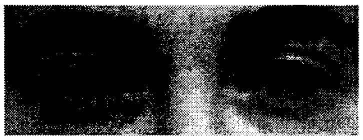
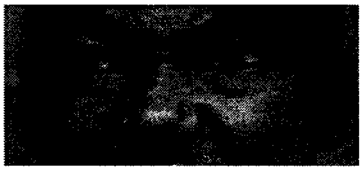
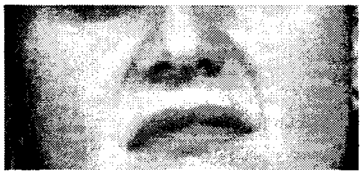
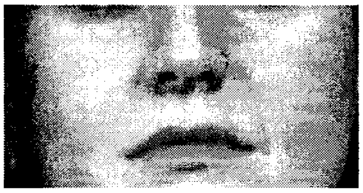
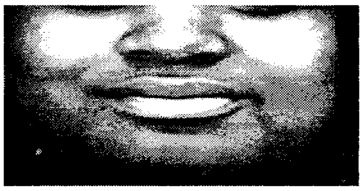
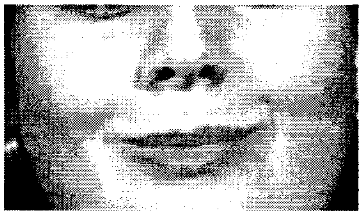
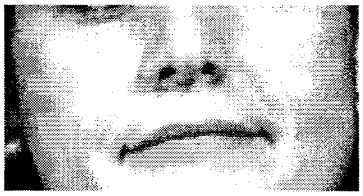
2.2. The Facial Action Coding System

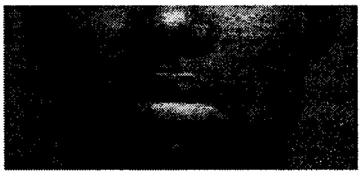
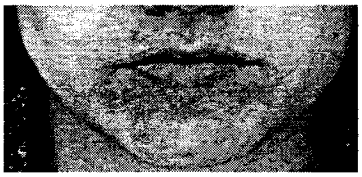
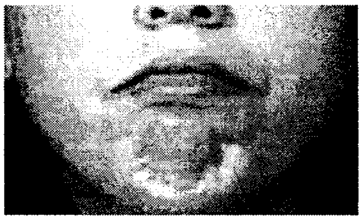
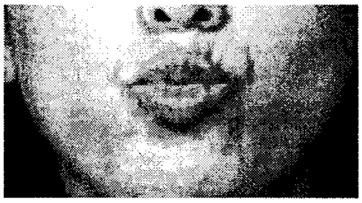
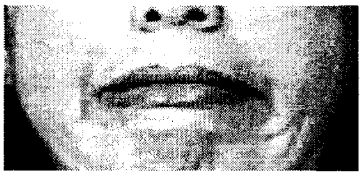
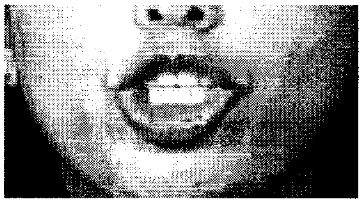
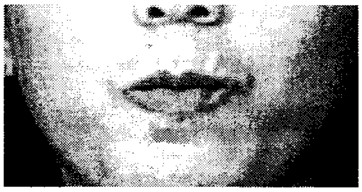
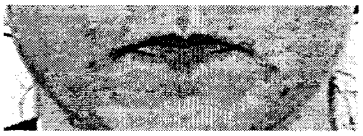
Facial Action Coding System [67] was developed by Paul Ekman and Wallace Friesen in 1978, to taxonomize every conceivable human facial expression and head movements. This is the most popular standard currently used in systematically categorizing the physical expression of emotions, and has been proven to be useful both to psychologists and to animators. FACS defines the expressions as one of 46 "Action Units" (AUs), each of which corresponds to the contraction or expansion produced by one or a group of related muscles. Activation of an AU is described in terms of the facial appearance change or head movement, i.e. changes of the facial feature components such as eyebrows, eyes, mouth and head caused by the activity of the underlying muscle(s). Using the FACS, all visually distinguishable facial movements can be described in terms of the AU codes.

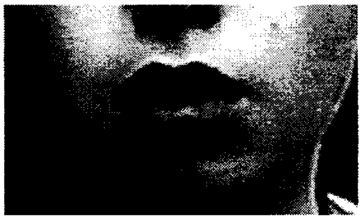
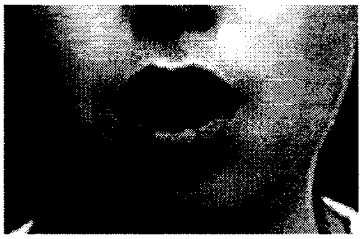
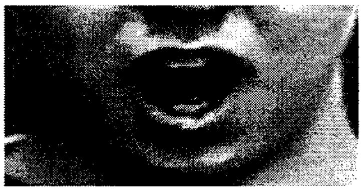
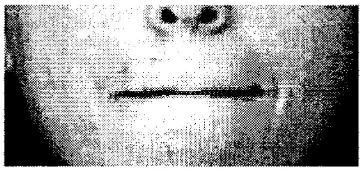
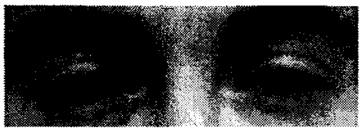
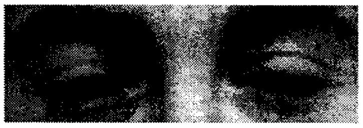
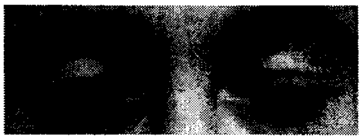
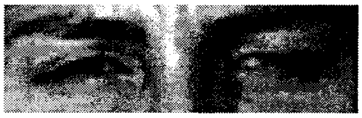
The primary goal of developing the FACS was to develop a comprehensive system which could distinguish every possible as well as visually distinguishable facial muscle movement. The motivation was that, since every facial movement is the result of different muscular action, a comprehensive system could be obtained by analyzing the anatomical basis of facial movement and discovering how each muscle of the face acts to change the visible appearance of the face. With that knowledge, it would be possible to measure any facial movement into its corresponding anatomically based minimal unit, termed as the Action Unit (AU). The FACS measurement units are called Action Units (AUs), not muscles movements for two reasons. First, for a few appearances, more than one muscles are combined into a single AU, since the changes in the appearance they produced could not be distinguished. Second, the appearance changes produced by one muscle were sometimes separated into two or more AUs to represent the relatively independent actions of different parts of the muscle. Part of the FACS that describes the Action Units involved in different facial expressions is provided in Table 2.

Table 2: Action Units from the Facial Action Coding System (FACS) that are involved in different facial expressions.

Action Unit	Linguistics Description	Example
Action Unit 1 (AU1)	Inner Brow Raiser	
Action Unit 2 (AU2)	Outer Brow Raiser	
Action Unit 4 (AU4)	Brow Lowerer	

Action Unit 5 (AU5)	Upper Lid Raiser	
Action Unit 6 (AU6)	Cheek Raiser	
Action Unit 7 (AU7)	Lid Tightener	
Action Unit 9 (AU9)	Nose Wrinkler	
Action Unit 10 (AU10)	Upper Lip Raiser	
Action Unit 11 (AU11)	Nasolabial Deepener	
Action Unit 12 (AU12)	Lip Corner Puller	
Action Unit 13 (AU13)	Cheek Puffer	
Action Unit 14 (AU14)	Dimpler	

Action Unit 15 (AU15)	Lip Corner Depressor	
Action Unit 16 (AU16)	Lower Lip Depressor	
Action Unit 17 (AU17)	Chin Raiser	
Action Unit 18 (AU18)	Lip Puckerer	
Action Unit 20 (AU20)	Lipstretcher	
Action Unit 22 (AU22)	Lip Funneler	
Action Unit 23 (AU23)	Lip Tightener	
Action Unit 24 (AU24)	Lip Pressor	

Action Unit 25 (AU25)	Lips part	
Action Unit 26 (AU26)	Jaw Drop	
Action Unit 27 (AU27)	Mouth Stretch	
Action Unit 28 (AU28)	Lip Suck	
Action Unit 41 (AU41)	Lid droop	
Action Unit 42 (AU42)	Slit	
Action Unit 43 (AU43)	Eyes Closed	
Action Unit 44 (AU44)	Squint	

The FACS has been developed by the trained human experts through strictly observing the muscle movements with respect to different facial expressions and head movement. However, until now, neither the facial muscle activity nor the AU codes can be extracted automatically from a digital face image applying the existing image processing techniques. Thus, the necessary action that can be taken with a view of resolving the problem of detecting the AUs automatically is to define some extractable visual properties of facial expressions towards constituting a face model that can be used in the computational analysis of facial expressions. A significant example of such a model is the point based analytic face model for representing the facial expressions.

2.3. The Analytic Face Model of Facial Expression Classification

In the analytic (feature-based) representation of face for facial feature extraction, some facial points or contours of the prominent facial feature regions (eyes, eyebrows and mouth) are used to model the face. The relative size and shape of the model features as well as the relative distances in between are then used for recognition and classification of facial expressions [3].

The idea of point based modeling and analysis of human face was introduced first by the point-light display experiments of G. Johansson in 1973 [69]. Through his experiment, Johansson suggested that the visual properties of the face, regarding information on facial expressions, could be made clear by describing the movement of some points belonging to the facial feature components (eyebrows, eyes, nose, mouth, and chin) and then by analyzing the relationship among those movements. The idea was extended further towards the analysis of facial expressions when Bassili [70] conducted an experiment in 1978 by covering faces of actors with black makeup and painting white spots randomly over it. Faces were divided into upper and lower regions (to correlate the FACS data of the upper and lower regions) and recognition studies were conducted. The study showed that in addition to the spatial arrangement of facial feature points, movement of the surface of the face does serve as a source of information for the recognition of facial expressions. This triggered the researchers of vision-based facial gesture analysis to initiate different attempts for determining the point-based visual properties of facial expressions through defining the point-based analytic face model, establishing a mechanism for automatic extraction of these points from digital facial image, and setting up a relation between the movement of the extracted facial feature points and the AUs. Then, some AU coded descriptions of the six basic emotional expressions are used to categorize them into the shown facial expressions. Examples of such 2-D point based analytic face models can be found in [28-33], [35], [37-40] and [42]. However, all of these models, except for the two described in [39] and [42], require various level of manual intervention for their formation by detecting the landmark points used in their structural constitutions.

In addition, the complicated 3-D wireframe face models [71-72] or the averaged optical flow within local regions (e.g. forehead, eyes, and mouth) [16], [44], [48-49] have also been used in some recently proposed facial expression classification method. However, it is difficult to design a 3-D face model that can accurately represent the facial geometric properties using currently available vision techniques. As a result, the initial adjustment between the 3-D wireframe and the surface images is usually manual, which affects the accuracy as well as efficiency of the recognition results. Similarly, precise and dense information of the facial expressions are not preserved when only the averaged optical flow within local facial regions is estimated.

For extracting the feature of facial expressions, an analytic face model composed of fifteen feature points from the frontal face view has been chosen for this work that can model the relationship between the facial muscles and the related AUs, along with preserving the deformation of the face due to different facial expressions. There are several motivating factors behind this choice. As shown by Johansson [69] and Bassili [70], a point based graphical face model resembles the model used by the human observers while judging a facial expression. As a result, the expression-classification rules, used by the human observers for recognizing facial expressions (e.g. the rules of FACS), can be converted easily into the rules of an automated classifier based on the point-based analytic face model. Another motivation is the simplicity of validating a point-based face model. The changes in the positions of the points in the face model are directly observable. So, by comparing the changes in the model and the changes in the modeled expression, the validity of the model can be visually inspected.

2.4. The Proposed Model for Extracting the Features of Facial Expressions

The target of this research is to develop a computer vision system that would be able to classify the seven facial expressions namely neutral, anger, disgust, fear, happiness, sadness and surprise from static face images in a fully automated manner. So, while proposing the analytic face model of facial feature extraction, the principal concern is to ensure that the facial feature points that would be used to form its structural constitution could be detected automatically towards minimizing the possible rate of rejection. The deformations of the feature points defined by the proposed analytic face model reveal changes in the appearance of eyes, eyebrows, nose and mouth and thus make it possible to establish a simple and unique relationship between the changes of the model features and the corresponding AUs of the Facial Action Coding System. To achieve this goal, a total of fifteen feature points (Figure 3.a) over the frontal view of human face has been selected that can detect the muscle movements of eyebrows, eyelids and mouth providing useful information about the involved action units (AU) or facial action parameters responsible for the seven facial expressions. A set of fifteen different measurements is performed using these feature points for estimating the level (strength) of activation of the triggered action units or the involved facial action parameters (Figure 3).

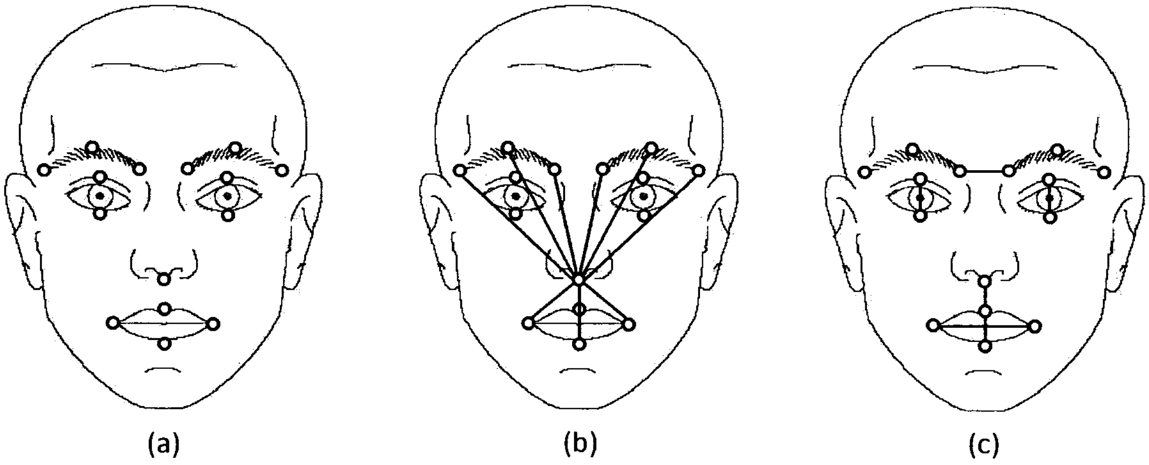


Figure 3: Analytic representation of face using fifteen feature points for classifying facial expressions (a) feature points for capturing the activated facial action units or facial

action parameters (b) and (c) fifteen distances that have been used in calculating the features for representing facial expressions.

Table 3: Description of the distances that have been used in calculating the features towards representing the facial expressions.

Distance	Description of the Distances used in Calculating Features
D_1 and D_2	Distance of the right and left eyebrow inner corners from the midpoint of nostrils
D_3 and D_4	Distance of the right and left eyebrow outer corners from the midpoint of nostrils
D_5 and D_6	Distance of the upper midpoints of right and left eyebrow from the midpoint of nostrils
D_7 and D_8	Distance of the right and left mouth corners from the midpoint of the nostrils
D_9	Distance of the midpoint of the lower lip from the midpoint of the nostrils
D_{10}	Distance of the mid upper lip from the midpoint of the nostrils
D_{11}	Distance between the inner corners of right and left eyebrows
D_{12} and D_{13}	Distance between the mid points of the upper and lower eyelids of left and right eyes
D_{14}	Distance between the mid points of the upper and lower lips
D_{15}	Distance between right and left mouth corners

Ten out of these fifteen measurements $(D_1, D_2, \cdots, D_{10})$ captures the deviations of the ten specific feature points from a non-changeable point. The midpoint of the nostrils has been used here as such a non-changeable rigid point as it remains in the same position even in the situation of different facial expressions and thus provides the level (strength) of activation of the triggered action units. The other five measurements $(D_{11}, D_{12}, \cdots, D_{15})$ reflect the deviations between five different pairs of points that can provide significant information about the presence of the seven facial expressions. Description of these distances that have been used in calculating the features for representing the facial expressions are provided in Table 3.

In order to perform person-independent classification of the facial expressions, it is necessary to normalize the values that correspond to the facial feature changes using the facial features extracted from the neutral faces. For this purpose, an average neutral face representation is generated from the distances measured among the fifteen feature points (Figure 3) of the training samples that belong to the class "neutral". The following formulation has been used for the purpose of this average neutral face generation:

$$D_{neutral} = \frac{1}{M}\sum_{i=1}^{M}(D_{i1}, D_{i2}, \cdots, D_{i15}) \tag{2.1}$$

Here, M is the number of samples of "neutral" faces among the whole training set, and D_{ij} is the j-th distance measured over the training sample i. Assuming that the number of images in the training set is N, the feature vector from each of the training samples is calculated as:

$$F_k = (D_k - D_{neutral}) \tag{2.2}$$

where $k = 1, 2, \cdots, N$; $D_k = (D_{k1}, D_{k2}, \cdots, D_{k15})$ is the k-th set of distances measured from the sample k, and F_k is the calculated feature vector for representing the facial expressions of the k-th training face image. All of these extracted feature sets are used to train a Multi-Class Support Vector Machine classifier so that it can classify facial expressions when given to it in the form of a feature vector. A similar process is carried out during the recognition phase for extracting the features of the facial expressions from the input image using the value of the average neutral face representation $(D_{neutral})$ obtained from the training session.

Chapter 3

Detection of the Facial Feature Regions

Among all the facial components, it is the eyebrows, eyes, nose and mouth that contribute most significantly in displaying the seven important facial expressions (neutral, anger, disgust, fear, happiness, sadness and surprise) over the human face. So, the fifteen feature points that have been used in defining the proposed analytic face model of facial feature extraction (please refer to Section 2.4) for facial expression classification are selected from these regions. But it is quite complicated and difficult to detect these fifteen feature points by directly searching the whole face region. As a result, attempts for the detection of such precise and specific feature points by searching the whole face directly yield a lower rate of successful identification, which in turn significantly deteriorates the performance of the underlying facial expression classification system [15]. So, the effective solution is to reduce the search space of these feature points by identifying each of these facial feature regions (eyebrows, eyes, nose and mouth) first, and then to perform detection of the corresponding feature points from their respective reduced search regions. One way of doing this is to adopt an existing method of facial feature region detection that implements a technique like geometrical shape analysis of facial features [73], [74], deformable and non-deformable template matching [75], [76], graph matching [77], snakes [78] or the Hough Transformation [79], combination of color information of each facial feature [80], [81], and machine learning approaches like Principle Component Analysis [82], Neural Network [83], Genetic Algorithm [84] and Haar wavelet based classifiers [85]. But the problem is that all of these methods specify only the center location of the facial feature regions which is not sufficient to confine the search space of the specific pre-defined feature points. As a solution to this problem, a method of facial feature regions localization has been proposed in this work that can isolate each of the facial feature regions with separate approximated bounding boxes utilizing the inherent anthropometric standard of human face and thus provides a strictly confined search space for finding the specific feature points [57].

After carefully observing the structural symmetry of human face and performing necessary anthropometric measurements over it, a generalized model of the human face has been constructed that can be used in detecting the approximated center points of the above mentioned facial feature regions. In this model, the distance between the centers of the left and right eye serves as the principal parameter of measurement using which the approximated centers of the other facial feature regions are identified. Hence, implementation of the method begins with the detection of both eye centers in every possible situation of eyes like open, closed, partially open, partially closed etc. Then the centre points of other facial feature regions are calculated using the anthropometric face model that have been developed statistically through performing craniofacial measurements over the subjects evolved in different geographical territories. Possible rectangular areas for bounding the facial feature regions around the calculated center points were also approximated using the distance between the eye centers as the key measuring parameter. Since the facial structures of peoples from different regions over the world have been considered while developing the anthropometric face model, the proposed method can localize facial feature regions from frontal face images of subjects evolved in different geographical territories. This is a fully automated process and can perform independently of the horizontal rotation (rotation over x-axis) and scale of human face in the face image.

3.1. Anthropometry and Its Application

Anthropometry is the biological science that deals with the measurement of the human body and its different parts. Data obtained from anthropometric measurement informs a range of enterprises that depend on knowledge of the distribution of measurements across human populations. For example, in human-factors analysis, a known range for human measurements can help guiding the design of products that fit most people [86]; in medicine, quantitative comparison of anthropometric data with patients' measurements before and after surgery can help in furthers planning and assessment of plastic and reconstructive surgery [87]; in forensic anthropology, conjectures about likely measurements derived from anthropometry figure in the determination of individuals' appearance from their remains [87], [88]; and in the recovery of missing children, anthropometry is used in aging their appearance taken from photographs [87]. The use of facial anthropometric data in this chapter describes another use of anthropometry in the localization of facial feature regions towards developing a computer vision system for automated classification of facial expressions.

Anthropometric evaluation begins with the identification of some particular locations on a subject known as the landmark points, which are defined in terms of the visible or palpable features over the skin of the subject. A series of measurements between these landmarks are then performed using some measuring instruments or methods following a set of carefully specified procedure. In order to develop useful statistics from anthropometric measurements, the measurements are made in a strictly defined way [89]. As a result, repeated measurements of the same individual or a group of people are very reliable, and measurements of different individuals can be compared successfully.

Systematic collection of anthropometric measurements has made possible a variety of statistical investigations in grouping of subjects on the basis of gender, race, age, "attractiveness" or the presence of a physical syndrome. Means and variances for the measurements within a group, tabulated in [87], [90], effectively provide a set of measurements which captures virtually all of the variation that can occur in the group. In addition to statistics on measurements, statistics on the proportions between measurements have also been derived. The description of the human form by proportions goes back to anthropometrists Dürer and da Vinci who found that proportions give useful information about the correlations between features and serve as more reliable indicators of group membership than simple measurements [91]. A widely used set of measurements for describing the human face have been elaborated by Farkas using a total of 47 landmark points [87]. Although many facial proportions show significant statistical differences across population groups, some of these proportions vary slightly across individuals as well as population groups and can be used to localize facial feature regions quite effectively.

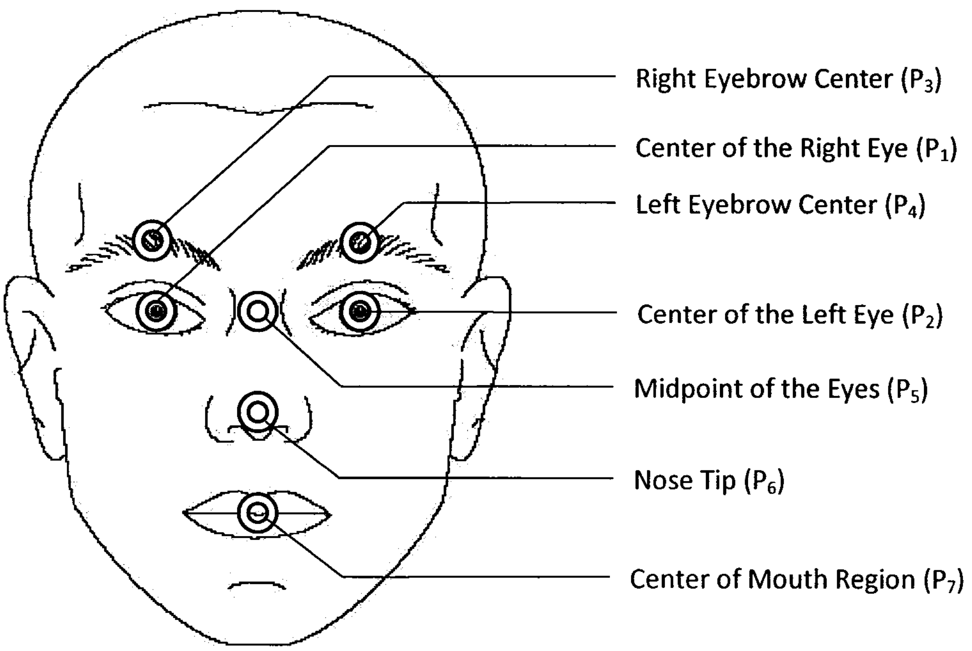
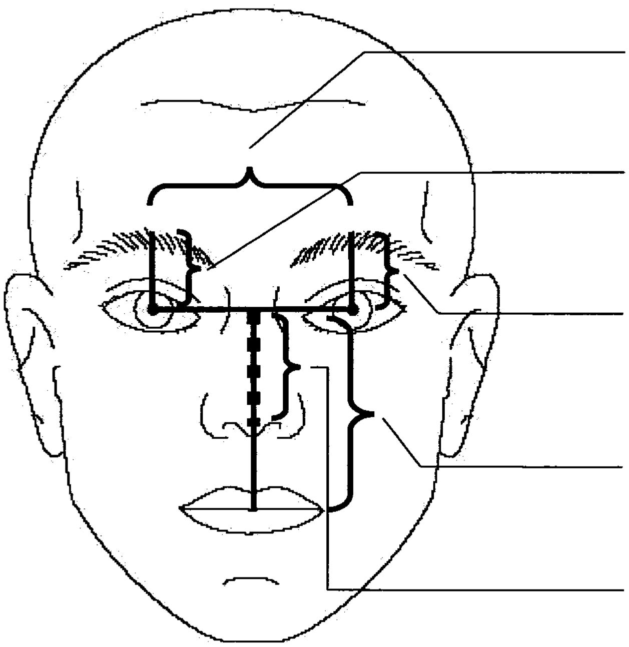

Right Eyebrow Center (P_3)
Center of the Right Eye (P_1)
Left Eyebrow Center (P_4)
Center of the Left Eye (P_2)
Midpoint of the Eyes (P_5)
Nose Tip (P_6)
Center of Mouth Region (P_7)

Figure 4: Landmark points used in defining the anthropometric face model for facial feature region localization.

3.2. Proposed Anthropometric Face Model of Facial Region Detection

After carefully analyzing the anthropometric measurements performed over 350 frontal face images taken from 180 subjects of different geographical territories, an anthropometric model of the human face has been constructed that can be used in localizing the facial feature regions from static face images. The face images that have been used for this purpose were collected from different publicly available face databases as well as from internet using the image searching facility of Google™. Rather than using all the landmarks introduced by Farkas [87], only a small subset of the points have been used and some new landmarks have been added in defining the proposed model that are either the centers of the facial feature regions (eyebrows, eyes, nose and mouth) or have significance in identifying the center points of these facial feature regions. The landmarks points that have been used in the anthropometric face model for facial feature localization are shown in Figure 4.



D_1: Distance between the Right Eye Center (P_1) and the Left Eye Center (P_2)

D_2: Distance between the Right Eye Center (P_1) and the Right Eyebrow Center (P_3)

D_3: Distance between the Left Eye Center (P_2) and the Left Eyebrow Center (P_4)

D_5: Distance between the Midpoint of Eyes (P_5) and the Centre of Mouth (P_7)

D_4: Distance between the Midpoint of Eyes (P_5) and the Nose Tip (P_6)

Figure 5: Distances (anthropometric measurements) used in defining the proposed model of facial region localization.

Analyzing the anthropometric data obtained by performing anthropometric measurements over the human faces, the following observations have been noticed regarding the structural symmetry of human face:

i. Distance between the centers of the right eye (P_1) and the left eye (P_2) remains almost the same for every subject even in the presence of each of the seven facial expressions (neutral, anger, disgust, fear, happiness, sadness and surprise).

ii. Proportion of the distance between the right eye center (P_1) and the right eyebrow center (P_3) to the distance between the center of left and right eyes remains almost constant for both the neutral faces and the faces with facial expressions. Similarly, this is also applicable in the case of the left eye center (P_2) and the center of the left eyebrow (P_4). However, a very little change to these proportional constants is observed in the presence of different facial expressions, which is ignorable.

iii. Proportion of the distance between the midpoint of eye centers and nose tip to the distance between the eye centers also maintains a constant value.

iv. Similarly, proportion of the distance between the midpoint of eye centers and mouth center to the distance between eye centers maintains a fixed value.

v. As the proportions specified in (ii), (iii) and (iv) remain almost same for the face images of different subjects, the centers of the eyebrows, nose and mouth can be obtained easily using the distance between the centers of the left and right eye as the principal parameter of measurement.

Since the measurements are performed on the digital image of human face, pixel based Euclidean distance has been used as the method of measurement. The performed anthropometric measurements are demonstrated in Figure 5 and the obtained proportional values are described in Table 4.

Table 4: Proportion of distances (D_2, D_3, D_4, and D_5) to D_1 measured from subjects of different geographical territories.

Proportion	Description	Values
D_2/D_1	Proportion of the distance between the right eye center and the right eyebrow center to the distance between the eye centers.	≈ 0.33
D_3/D_1	Proportion of the distance between the left eye center and the left eyebrow center to the distance between the eye centers.	≈ 0.33
D_4/D_1	Proportion of the distance between the midpoint of eye centers and the nose tip to the distance between the eye centers.	≈ 0.60
D_5/D_1	Proportion of the distance between the midpoint of eye centers and the mouth center to the distance between the eye centers.	≈ 1.10

It is clear from Figure 5 that if the center points of the eyes (P_1 and P_2) are known, the approximated centers (P_3 and P_4) of the right and the left eyebrow region can be obtained directly using the proportional values provided Table 4. Similarly, the center points of the nose and mouth region (P_6 and P_7) can also be determined using the midpoints between the left and right eye centers (P_5) as an intermediate point. Besides this, approximated bounding rectangular regions around the eyebrows, eyes, nose and mouth can also be set up using the distance between the eye centers as the principal measurement criteria.

3.3. Isolation of the Facial Feature Regions

For isolating the eyebrow, eye, nose and mouth region from a static face image using the proposed anthropometric face model of facial feature region detection [57], the centers of the eyes are detected first from the grayscale converted version of the face image. Then, the horizontal rotation (rotation over the x-axis) of the face in the face image is fixed. Centers of the other facial feature regions (eyebrows, nose and mouth) are then approximated using the proposed anthropometric face model described in Section 3.2. At the final stage, the rectangular facial feature regions are isolated as well as extracted using the distance between the centers of the left and right eyes as the principal parameter of measurement. Block diagram of the complete facial feature isolation process is given in Figure 6.

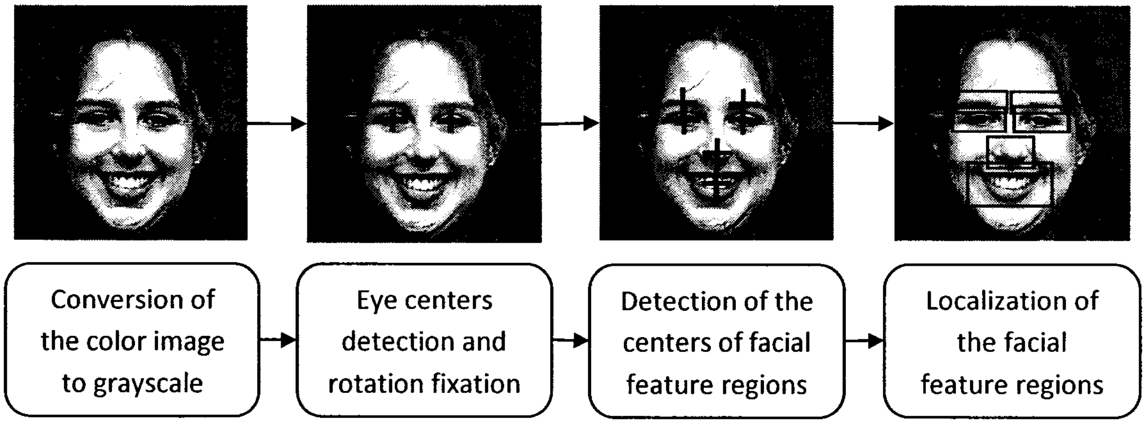

Conversion of the color image to grayscale
Eye centers detection and rotation fixation
Detection of the centers of facial feature regions
Localization of the facial feature regions

Figure 6: Detection of the facial feature regions using the anthropometric face model based technique of facial feature regions localization [57].

3.3.1. Conversion of the Color Face Image to Its Grayscale Version

The image databases that have been used in this research contain both the RGB color image and the grayscale image. But the image processing techniques that have been adopted for isolating the facial feature regions as well as for detecting the fifteen facial feature points from the face images are mostly effective when applied to the grayscale images only. As a result, conversion of the color face images to its corresponding grayscale version is necessary at the very beginning of the facial feature region localization process.

In RGB images, color value of each pixel is represented using three components, one for each primary color (Red, Green or Blue). The value of each of these color components can vary between the minimum (no color) and maximum (full intensity). If all the color values are at the minimum level, the result is black and if all the color values are at the maximum level, the result is white. For computer representation of the RGB images, each of the color values are stored as numbers within the range [0 - 255]. This requires a total of 24 bits for storing the color information of each pixel of a RGB image.

In grayscale images, color value of each of the pixels is represented using just a single integer. Displayed images of this sort are typically composed of the shades of gray, varying from black at the weakest intensity to white at the strongest, though in principle, the samples could be displayed as shades of any color, or even coded with various colors for different intensities. Often, the grayscale intensity is stored as an 8-bit integer giving 256 possible different shades of gray from black to white. If the levels are evenly spaced, the difference between successive grayscales is significantly better than the grayscale resolving power of the human eye.

To convert any RGB color image to its most approximate grayscale one, the values of its red, green and blue (RGB) components are obtained first. Then, the value of each pixel of the grayscale image is calculated by adding 30% of the red value, 59% of the green value and the 11% of the blue value from the corresponding pixel of the RGB image. The conversion mechanism can be formulated as:

$$Gray = 0.3 * R + 0.59 * G + 0.11 * B \tag{3.1}$$

These percentages are chosen due to the different relative sensibility of the normal human eye to every of the primary colors (higher to the green, lower to the blue). Once all the pixels of the RGB image are converted, the resulted image becomes the grayscale version of the original RGB color image.

3.3.2. Detection of the Eye Centers

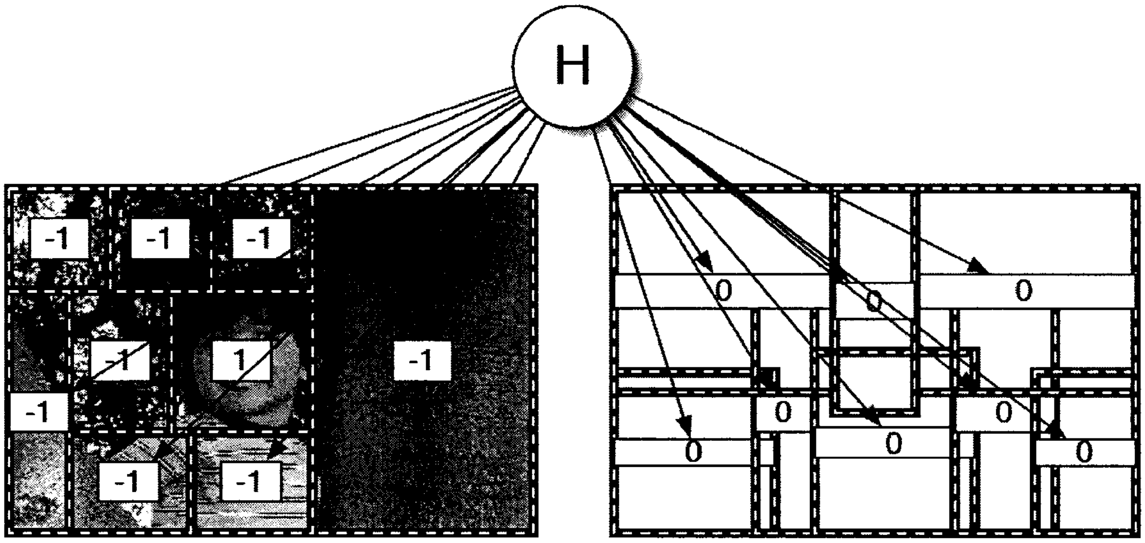
The generative framework for real time object detection and classification proposed by Fasel et al. [92] has been used in this work for detecting the eye centers from the face images. This framework uses a probabilistic model of image generation and an optimal inference algorithm for finding the objects and the object features within it. In this approach, images are modeled as a collage of patches of arbitrary size, some of which contain the object of interest and some of which are background. The likelihood-ratio models for the object versus background generated patches are developed using these arbitrary sized collage of patches, which are learned using a variation of the boosting method known as GentleBoost [93]. The objects of interest within the image are then identified applying a multi-scale image searching technique using these previously learnt likelihood-ration models.

3.3.2.1. Generative Modeling of the Images for Object Detection

The method of object detection proposed in [92] formulates the problem of finding objects as a Bayesian inference problem by deriving a model of how images are generated, and then develop an algorithm for making optimal inferences under this model. The images are modeled as a collage of rectangular patches of arbitrary size and location, some patches rendering the object of interest, the others rendering the background. Given an image, the goal is to discover the patches that rendered the object. Let Y be a random matrix representing an image with a fixed number of pixels and y be a specific sample from Y. Let $\mathcal{A} = \{a_1, a_2, \cdots, a_n\}$ be an enumeration of all possible rectangular image patches where, a_i determines the position and geometry of a rectangle on the image plane. Let y_{ai} be a matrix whose elements are the values of y for the pixels in the rectangle a_i. Let $H = \{H_1, H_2, \cdots, H_n\}$ be a random vector that assigns each of the n patches to one of three categories: H_i takes the value of 1 when the patch a_i renders the object of interest, the value of -1 when the patch is rendered as the background, and the value of 0 when it is not rendered (Figures 7).

The image generation process proceeds as follows (Figure 7). First, segmentation h is chosen with probability $p(h)$. Then for each patch a_i if $H_i = 1$, an image of size

a_i is chosen from the object distribution $q(\cdot\,|a_i, 1)$ independently of all the other patches. If $H_i = -1$, a background image y_{ai} is chosen from the background distribution $q(\cdot\,|a_i, -1)$. If $H_i = 0$, a_i is not rendered. So, the observed image y is the collection of the rendered patches.



(a) Observed Image

(b) Unrendered Patches

Figure 7: Modeling the images as a collage of arbitrary sized patches. The hidden variable H determines which image patches will render the background (-1), which patches will render the object of interest (1) and which patches will not be rendered (0). The set of rendered patches determines the observed image.

The model is specified by the prior probabilities $p(h)$ and by the object and background rendering distributions q. The prior is specified by the marginal probabilities $\{P(H_i = 1): i = 1, 2, \cdots, n\}$ with the constraint that values of h that do not partition the image plane have zero probability, and by one of the two following constraints: (I) for cases in which it is known that there is one and only one object of interest on the image plane, only the values of h with a single 1 are allowed. (II) for cases in which there may be an arbitrary number of objects of interest, it is assumed that the location of a rendered object does not inform us about the location of other objects, expect for the fact that each pixel can only be rendered by a single object or background element. More formally, for $i = 1, 2, \cdots, n$; the random variables $\{H_i : j \neq i\}$ are independent of H_i when conditioning on the event $H_i \neq 0$. For a given image y, the goal is to detect the patches that are rendered by the object. There are two cases of interest: (I) it is known that there is one and only one patch rendered by the object (II) there is an unknown and arbitrary number of patches rendered by the object model.

In the situation that there is one and only one patch in the image plane y that renders the object of interest, the goal is to find the most probable patch $\hat{k} \in \{1, 2, \cdots, n\}$ such that,

$$\hat{k} = \underset{i}{\operatorname{argmax}}\ P(H_i = 1 | y) \tag{3.2}$$

According to [92], equation (3.2) can be expressed in terms of the log-likelihood ratio as:

$$\hat{k} = \underset{i}{\operatorname{argmax}}\ \log P(H_i = 1) + \log \frac{q(y_{ai}; a_i, 1)}{q(y_{ai}; a_i, -1)} \tag{3.3}$$

This equation prescribes the way of scoring each of the possible patches in terms of a function that includes the prior probability of that patch containing an object and a likelihood ratio term. The patch that maximizes this score is then chosen.

In the situation when multiple objects of interest are present, it is not possible to know about their number in advance. So, to formalize the problem, a function Φ has been defined according to [92] for measuring the degree of similarity between any two arbitrary segmentations h and h' as:

$$\Phi(h, h') = \sum_{i}^{n} \rho(H_i, H_i') \tag{3.4}$$

$$\rho(H_i, H_i') = \left(\delta_{H_i,1} + \delta_{H_i,-1}\right)\delta_{H_i,H_i'} \tag{3.5}$$

Where, δ is the Kroenecker delta function [92], and ρ counts the number of patches for which both h and h' assign the same "object" or "background" label by ignoring all the patches that are not rendered by h. Here, the goal is to find a partition $\hat{h}$ that optimizes the expected match

$$\hat{h} = \underset{h'}{\operatorname{argmax}}\ E(\Phi(H, h')|y) = \sum_{h} p(h|y)\Phi(h, h') \tag{3.6}$$

This optimal assignment follows the following conditions:

$$\hat{h}_i = f(x) = \begin{cases} 1, & if\, p(H_i = 1|y) > p(H_i = -1|y) \\ -1, & otherwise \end{cases} \tag{3.7}$$

Thus, to find the optimal assignment, it is necessary to scan all the possible patches $a_1, a_2, \cdots, a_n$ for computing the following log-posterior probability ratio

$$\log \frac{P(H_i = 1|y)}{p(H_i = -1|y)} \tag{3.8}$$

and assign "object" label to the patches for which this ratio (3.8) is larger than 0. According to [92], since, $\{H_j : j \neq i\}$ are independent of H_i, given the fact that $H_i \neq 0$; the log-posterior probability ratio (Equation 3.8) can be expressed as:

$$\log\frac{P(H_i = 1|y)}{p(H_i = -1|y)} = \log\frac{P(H_i = 1)}{p(H_i = -1)} + \log\frac{q(y_{ai}; a_i, 1)}{q(y_{ai}; a_i, -1)} \tag{3.9}$$

In order to make optimal inferences under this framework, the things that are necessary are a model for the prior probability of object locations, and a model for the log-likelihood ratios of the image patches of arbitrary geometry.

3.3.2.2. Learning the Likelihood Ratio Using GentleBoost

The inference algorithm discussed in sub-section 3.3.2.1 requires a likelihood ratio model. Given an arbitrary image patch y, an estimation of the ratio between the probabilities of such a patch being generated by the object class vs. the background class has to be obtained. In [92], these likelihood ratios have been learnt using GentleBoost, a boosting algorithm developed by Friedman et al. [93]. Boosting [94] refers to a family of machine learning algorithms for learning classifiers by sequential accumulation of experts that focus on the mistakes made by previous experts. Friedman et al. [93] showed that boosting methods can be reinterpreted from the viewpoint of sequential statistical estimation, an interpretation that makes it possible to use it in the generative framework proposed in [92].

The goal is to learn a model of the log-likelihood ration of arbitrary image patches. During training, a set of examples $\{(y_i, z_i): i = 1, \cdots, m\}$ is given where, y_i is an image patch, and $z_i \in \{-1, +1\}$ is its category label, i.e. object or background. The model used in GentleBoost is of the following form:

- Let $\{(y_i, z_i): i = i, \cdots, m\}$, be a set of training examples, where y_i is the image patch and $z_i \in \{-1, +1\}$ is the label that defines its category.
- Let $P_i(t)$ represents the weight assigned to the i-th example at the beginning of the t-th iteration of the GentleBoost algorithm.
- Let the initial distribution be: $P_0(i) = \frac{1}{m}$, for $i = 1, \cdots, m$; that is, each training example is weighted equally.
- For time $t = 1, \cdots$
  - For wavelet $w = 1, \cdots, n$

    Use kernel-regression to find the tuning curve h that best fits the set of triples $\{(w(y_i), z_i, P_i(t)): i = 1, \cdots, m\}$.
  - Choose $(\hat{w}, \hat{f})$, the wavelet and the tuning curve that minimize the error function ρ. They define the selection expert at iteration t as:

    $$f_t(y) = \hat{h}(\hat{w}(y))$$
  - Update the distribution over the training elements as:

    $$P_{t+1}(i) = P_t(i)\frac{e^{f_t(y_i)z_i}}{Z_t}$$

    where Z_t is a normalization factor defined by:

    $$Z_t = \sum_i P_t(i) e^{f_t(y_i)z_i}$$
  - Update the posterior probability model

    $$P(y) = \frac{1}{1+e^{-2\sum_{n=1}^{t} f_n(y)}}$$

Figure 8: Algorithm for learning the likelihood-ratio models using GentleBoost [92].

$$p(y) = \frac{1}{1 + e^{-2\sum_j f_j(y)}} \tag{3.10}$$

where $p(y)$ is the probability that image patch y belongs to one of the two categories of interest, and $f_i(y)$ is the opinion of the i-th expert, as defined in Figure 8. GentleBoost can be seen as an application of the Newton-Raphson optimization algorithm to the problem of minimizing the following chi-square (χ^2) error [93]:

$$\rho = \sum_i \frac{t_i - p(y_i)}{\sqrt{p(y_i)(1 - p(y_i))}} \tag{3.11}$$

where $t_i = 0.5(z_i + 1) \in \{0, 1\}$ is the category label for the i-th training input y_i. Since $p(y_i)$ is the probability of a Bernoulli random variable with mean $p(y_i)$ and standard deviation $\sqrt{p(y_i)(1 - p(y_i))}$, then ρ can be seen as the number of standard deviations between the observed label and the average label value. As the number of examples in the training set increases, minimizing the chi-square (χ^2) error becomes identical to maximizing the likelihood. However when the number of samples is small, chi-square estimators can be more efficient than maximum likelihood estimators.

Selecting Wavelets and the Tuning Curve

GentleBoost chooses a set of experts $(f_1, f_2, \cdots)$ in a sequential manner. Given the already selected set of experts, each Newton-Raphson step results in selection of the expert that maximally reduces the current chi-square (χ^2) error. In practice this can be done in a variety of ways. However, the following approach introduced in [92] has been applied in this work:

A large pool (about 170,000) of wavelets $\{w_i, w_2, \cdots, w_n\}$ are created and experts are defined as the combination of a wavelet and a tuning curve h. In this way, $(t-1)$ experts are selected after t iteration of the Newton-Raphson method. Then, for each wavelet, the tuning function $h: \mathcal{R} \longmapsto [-1,1]$ that minimizes ρ given the outputs of the wavelet w and the information provided by the $(t-1)$ experts are already selected. This function can be shown to have the following form:

$$h\big(w(y)\big) = E^{P_t}[Z|w(y)] \tag{3.12}$$

where $Z \in \{-1,1\}$ is the category label, and the expectation is taken with respect to the distribution induced by the weight assigned by GentleBoost to different training data (Figure 8). The function h is estimated using the Nadaraya-Watson kernel regression method for density estimation [95]. The training examples used in this regression method are the set of triples $\big\{\big(w(y_i), z_i, P_t(y_i)\big): i = 1, \cdots, m\big\}$, where $w(y_i)$ is the regressor variable, z_i is the label to be predicted and $P_t(y_i)$ is the weight of the example $(w(y_i), z_i)$.

The function h is called the tuning curve of the wavelet w. After the optimal tuning curves for all the wavelets of the original pool are obtained, the wavelet $\widehat{w}$ and the corresponding tuning curve $\hat{h}$ are chosen that minimize ρ. This pair defines the expert selected for iteration t as:

$$f_t(y) = \hat{h}(\widehat{w}(y)) \tag{3.13}$$

The process is iterated, each time adding a new wavelet and tuning curve, until the value of ρ no longer decreases. The whole procedure has been illustrated in Figure 8.

By the end of training process, the posterior probability of the object class is modeled using the given arbitrary image patches y as:

$$p(y) = \frac{1}{1 + e^{-2\sum f_t(w_t(y))}} \tag{3.14}$$

This posterior probability estimate reflects the particular proportion π of examples of each class used during training. The inference algorithm in (3.3) requires log-likelihood ratios, not log-posteriors. According to [92], these can be easily derived from (3.14) using the Bayes rule as:

$$\begin{aligned} \log\frac{q(y_{ai}; a_i, 1)}{q(y_{ai}; a_i, -1)} &= \log\left(\frac{1-\pi}{\pi}\right) + \log\left(\frac{p(H_k = 1|y_{ai})}{p(H_k = -1|y_{ai})}\right) \\ &= \log\left(\frac{1-\pi}{\pi}\right) + 2f(x) \end{aligned} \tag{3.15}$$

Thus, combining (3.3) and (3.15), the final equation for inference is written as:

$$\hat{k} = \underset{i}{\operatorname{argmax}}\ P(H_i = 1|y) = \underset{i}{\operatorname{argmax}}\ \log p(H_i = 1) + 2f(y_{ai}) \tag{3.16}$$

3.3.2.3. Situation Based Inference for Eye Center Detection

One common approach to eye detection is based on the operation of a set of independent feature detectors. The output of these detectors (e.g., a detector for the left eye, a detector for the right eye, a detector for the tip of the nose, etc.) is integrated by looking for configurations that match the distribution of the inter-feature distances typical to human face. Unfortunately this method scales exponentially with the number of false alarms of each feature detector. Suppose the goal is to find the eye centers with one pixel accuracy. This requires for background models to include examples of eyes shifted by one pixel from the center position. In practice, a detector efficient in distinguishing eyes slightly shifted from the eye centers is also likely to produce a large number of false positives while scanning general backgrounds that do not contain faces, creating an insurmountable problem for methods that rely on feature detection.

The approach proposed in [92] is based on the idea of a bank of situational or context dependent experts operating at different levels of specificity. For example, since the eyes occur in the context of faces, it may be easier to detect eyes using a very large context that include the entire face and then formulate feature detectors specifically designed to work well under such context. This form of eye detection works under very general context conditions, avoiding the proliferation of false alarms, but provides poor information about the precise location of the eyes. These eye detectors are complemented by context-specific eye detectors that provide very precise information about the position of the eyes. More formally, let y represents an observed image, S represents a contextual situation (e.g., the location and scale of a face on the image plane), and O represents the location of the left eye of that face on the image. Using the law of total probability, the location of the left eye can be expressed as:

$$p(o|y) = \int p(s|y)p(o|sy)\, dh \tag{3.17}$$

Here $p(s|y)$ works as a situation detector. Its role is to find the regions in the image plane that are likely to contain eyes due to the fact that they contain faces. The $p(o|sy)$ term is a situation specific eye detector. For example it may work when the location and scale of the face on the image plane is known. In this example $p(s|y)$ partitions the image pixels into those belonging to the face y_f, and those belonging to the background y_b. Once the position and scale of the face are known, the background provides no additional information about the position of the eye, i.e.,

$$p(o|y_f, y_b, s) = p(o|y_f s) \tag{3.18}$$

The situational approach proposed in [92] is iterated, where the first one detects a general context, followed by the detection of a context within a context, each time achieving higher levels of precision and specificity allowed by the fact that the context models become smaller and smaller in each iteration.

3.3.2.4. Architecture of the Eye Detection System

As specified earlier, the eye detection system developed on the basis of the generative framework for real time object detection and classification proposed by Fasel et al. [92] has been used in this work for detecting the centers of the left and right eye. The system utilizes two types of eye detectors: The first type, which can be thought of as a face detector, starts with complete uncertainty about the possible location of eyes on the image plane. Its role is to narrow down the uncertainty about the location of the eyes while operating under a very wide variety of illumination and background conditions. The second type of detector operates on the output of the first detector such that it can assume a restricted context and achieve higher location accuracy. The flowchart for this procedure is shown in Figure 9.

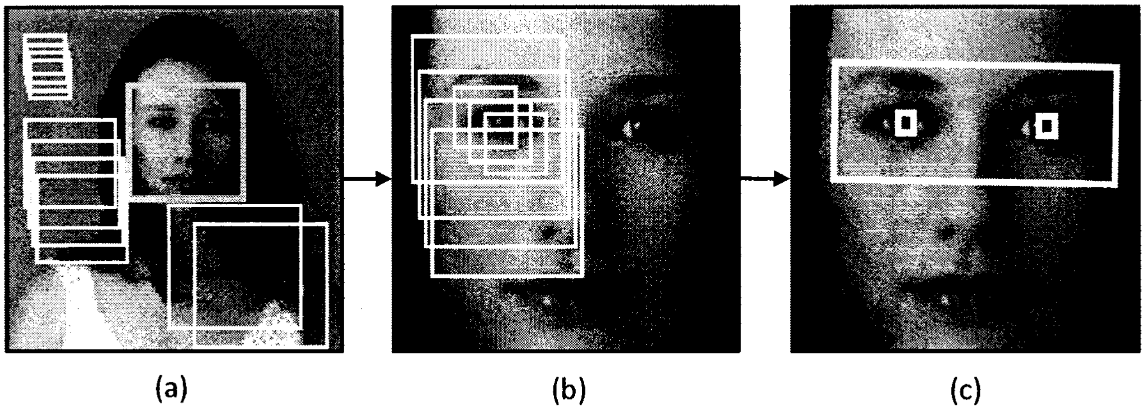


Figure 9: Flowchart of the eye detection procedure (a) the first detector scans for the face within the entire image at multiple scales (b) the second detector scan for the eyes at multiple scales within the region detected by the first detector (c) cropped and rotated best eye center region.

Stage I: Eye Detection in General Background Conditions

As described above, the first component of the inference process locates regions of the image plane that contain faces, and thus eyes. This module operates under very general background as well as illumination conditions and greatly narrows down the plausible locations of eyes on the image plane without making any prior assumptions about the location of the face.

The general procedure for the image searching is similar to the multi-scale search technique proposed by Rowley et al. [96], who trained a single binary classifier using fixed size (20 × 20 pixel) patches to classify face vs. non-face, and then used that classifier to classify all possible patches from the image. Faces larger than the original fixed size were found by repeating the search in copies of the image, scaled successively to some smaller sizes (thus, finding a 20 × 20 pixel face in a 1/4 size copy of the image means that the original image contains a 80 × 80 pixel face at that location).

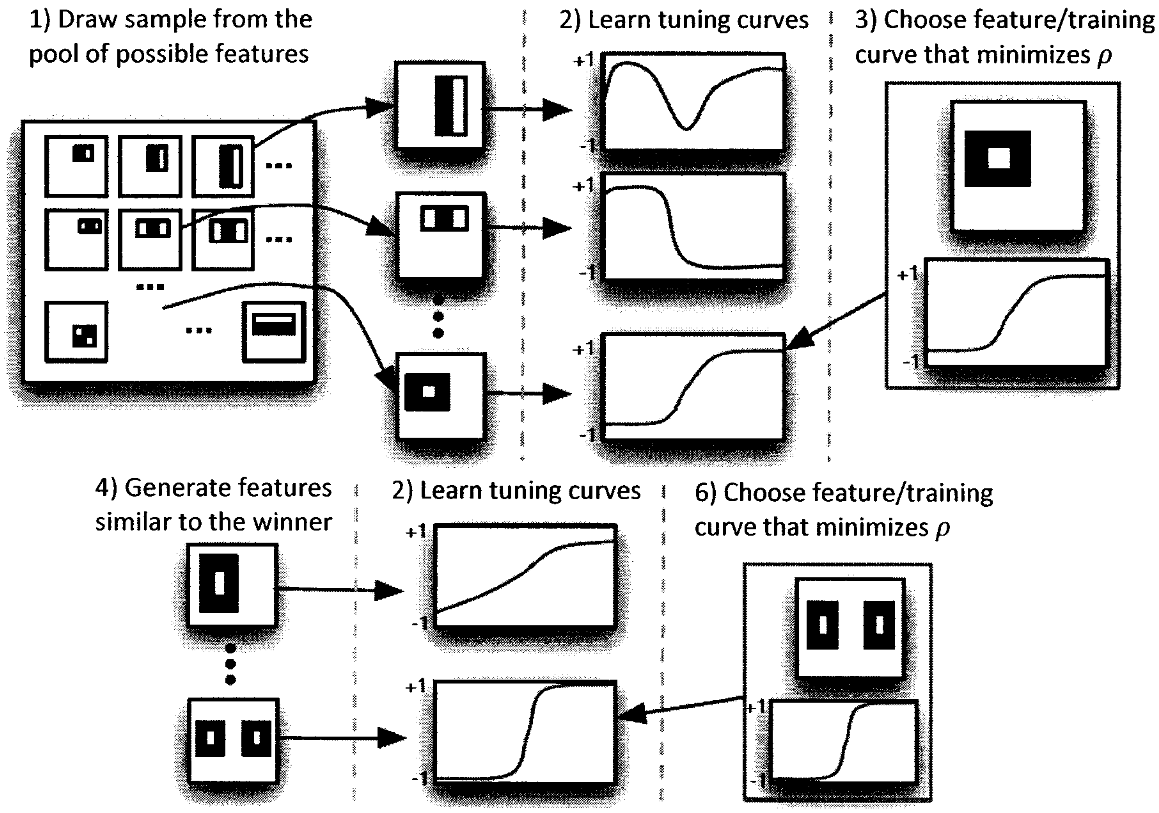

1) Draw sample from the pool of possible features
2) Learn tuning curves
3) Choose feature/training curve that minimizes ρ
+1
-1
4) Generate features similar to the winner
2) Learn tuning curves
6) Choose feature/training curve that minimizes ρ

Figure 10: Flowchart of one iteration of the Haar-like wavelet based feature selection procedure (Extracted from [92]). Each iteration of the procedure is divided into two stages. In the first step, a random sample of 5% of the possible wavelets is taken and the tuning curves are defined for each of them that minimize the loss function ρ. In step two, the selection of the first stage is refined by finding the best performing single-wavelet classifier from a new set of wavelet.

A very similar scheme of image searching has been used in the eye detection procedure discussed here. However, rather than using a binary classifier, a likelihood-ratio model has been developed using a dataset containing 5000 images of frontal upright faces taken under a variety of illumination conditions, facial expressions, facial hair, eyeglasses, hats, etc., of widely varying image quality. Faces were cropped and scaled to 24 × 24 pixels square. The negative examples were sampled from a dataset of 8000 images collected from the internet and are known not to contain faces. Similarly, these images contained a wide variety of natural indoor and outdoor scenes, text, illustrations, posed images of objects, etc., with varying image quality. The advantage of this web dataset is that it includes far more variability than most other closed databases.

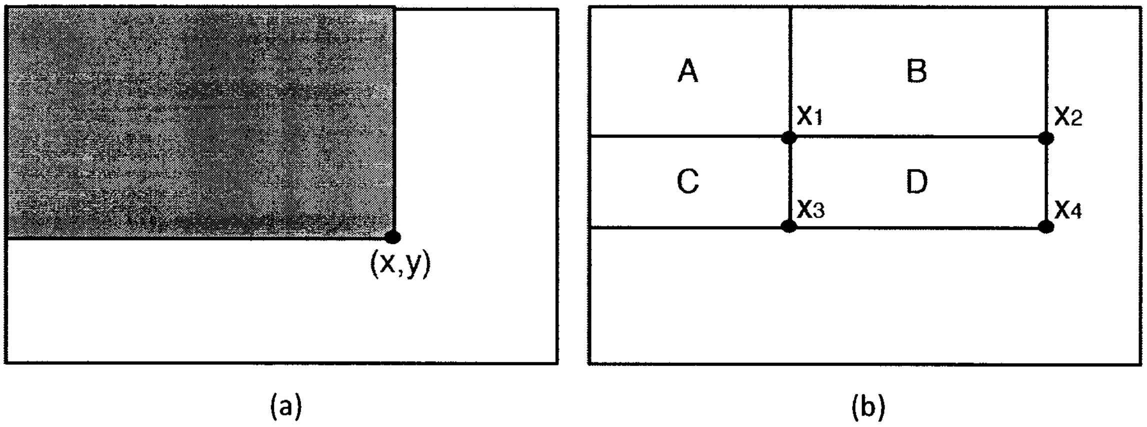
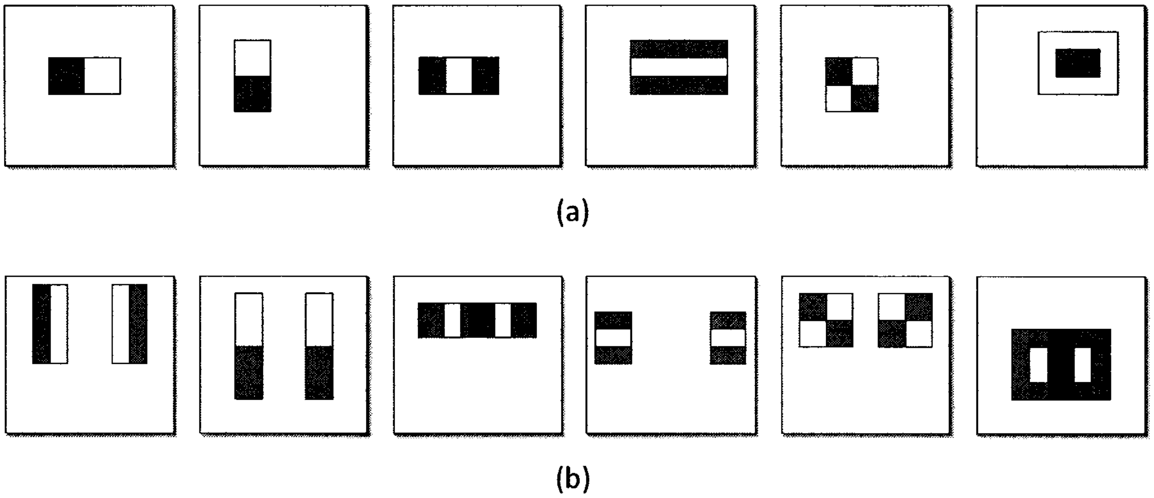


Figure 11: The integral image representation introduced by Viola and Jones [97] (a) the value of the pixel (x, y) is the sum of all the pixels above and to the left (b) the sum of the pixels within the rectangle D in the original image can be computed from the points of the integral image as: $x_4 - x_2 - x_3 + x_1$.

Due to the multi-scale search, about one billion total patches are possible in these 8000 images. As the initial negative examples of training, 10,000 square patches of arbitrary size and at arbitrary locations in the images, were sampled from this dataset. These patches were then scaled down to 24 × 24 pixels.


(a)

(b)

Figure 12: Haar-like wavelets used by the eye detector. Each wavelet is computed by taking the difference of the sums of the pixels in the white boxes and the gray boxes (a) wavelet types include those specified in [97], plus a center-surrounded type wavelet (b) in the refinement step, the same wavelet types superimposed on their reflection about the y axis have been used.

The likelihood-ratio model was trained using the GentleBoost method described in subsection 3.3.2.2 and Figure 8. GentleBoost sequentially chooses wavelets from a large pool and combines them to minimize a chi-square (χ^2) error function. The pool of wavelets chosen from was based on those introduced by Viola and Jones [97] and consists of Haar-like wavelets (Figure 12). The main reason for their use is that their output can be computed very fast by taking the sum of pixels in two, three, or four equalized, adjacent rectangles and taking differences of these sums (Figure 11). In addition, a center-surround type wavelets and mirror image wavelets that are sensitive to patches symmetric about vertical axis have also been used (Figure 12).

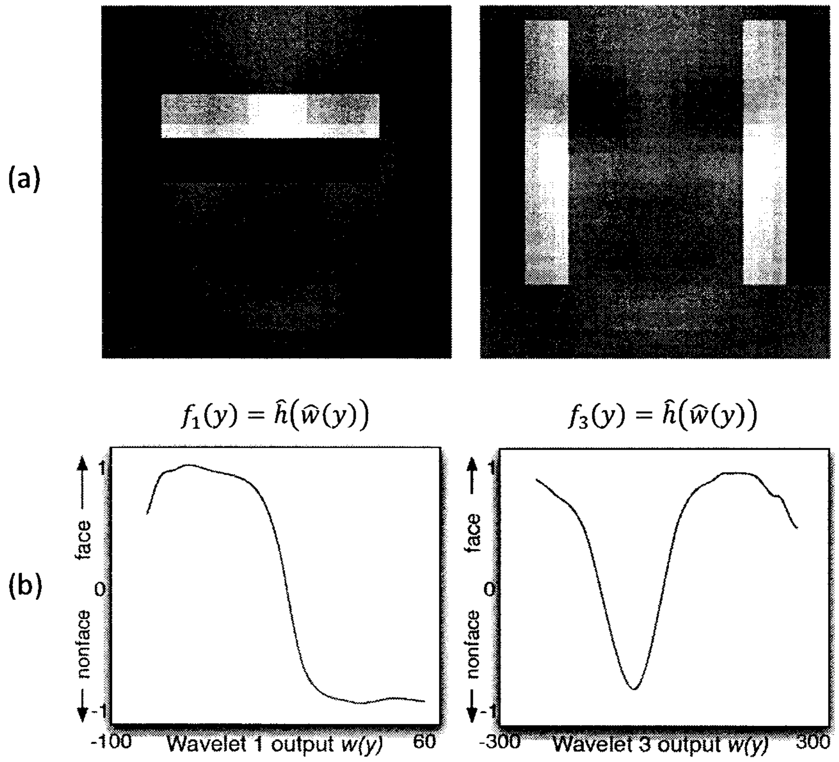


Figure 13: The first two wavelets shown over the face (a) and their respective tuning curves (b) for face detection. The tuning curves show the evidence for face (high) vs. non-face (low), as a function of the output of the wavelet (increasing from left to right). The first tuning curve shows that a dark horizontal region over a bright horizontal region in the center of the window is evidence for an eye and for non-eye otherwise. The second tuning curve is bimodal, with high contrast at the sides of the window evidence for a face, and low contrast evidence for non-face.

The GentleBoost approach described in sub-section 3.3.2.2 requires computing tuning curves on each of the wavelet candidates. But it is very computationally expensive to perform an exhaustive search over all these wavelets in a 24 × 24 pixel window since there are over 170,000 possible wavelets of this type. To speed up training, the wavelet selection step is divided into two stages (Figure 10). First, at each round of boosting, a random sample of 5% of the possible wavelets is taken. For each wavelet, the tuning curve is defined that minimizes the loss function ρ if that particular wavelet was added to the pool of the already chosen wavelets. In step two, the selection of the first stage is refined by finding the best performing single-wavelet classifier from a new set of wavelets generated by shifting and scaling the best wavelet by two pixels in each direction, as well as composite wavelets made by reflecting each shifted and scaled wavelet horizontally about the center and superimposing it on the original. Using the chosen classifier as the weak learner for this round of boosting, the weights over the examples are then adjusted applying to the GentleBoost rule. This wavelet selection process is then repeated with the new weights, and the boosting procedure continues until the performance of the system on a validation set no longer decreases.

The inference algorithm calls for likelihood ratio models at multiple scales. Likelihood ratios for larger image patches are obtained by linearly scaling the patches down to 24 × 24 pixels and then applying the likelihood ratio model trained on that particular scale. Because of the choice of Haar-like wavelets for the higher level image representation, this interpolation step is accomplished in constant time regardless of the image scale [97], [98]. Rather than training a "monolithic" classifier which evaluates all its wavelets before it makes a decision, the classifier of the eye center detection procedure is divided into a sequence of smaller classifiers as introduced in the work of Viola and Jones [97]. The classifiers are organized in a cascaded manner and can make an early decision to abort further processing on a patch if its likelihood-ratio falls below a minimum threshold. This can be considered as a situational cascade where each level of the cascade is trained only on patches that survived the previous levels. After each element of the cascaded is trained, a boot-strap round [99] is performed, in which the full system up to that point is scanned across a database of non-face images, and false alarms are collected and used as the non-faces for training the subsequent strong classifier in the sequence. Figure 13 shows the first two wavelet chosen by the system along with the tuning curves for those wavelets.

During the detection phase, the inference algorithm calls for scanning the entire image plane and looking for square patches of arbitrary scale and location with larger likelihood-ratios. In practice, it starts scanning for patches of size 24 × 24, the minimum scale of interest and shift one pixel at a time until all possible patches of this size are scanned. Each larger scale is chosen to be 1.2 times of the previous scale, and the corresponding offsets are scaled by the same proportion. For a 640 × 480 pixel image, this produces over 400,000 total patches [92]. However, since the early layers of the cascade need very few wavelets to achieve good performance (the first stage can reject 60% of the non-faces using only 2 wavelets that costs 20 simple operations, or about 60 microprocessor instructions), the average number of wavelets that need to be evaluated for each window is very small, making the overall system very fast while maintaining high level of accuracy.

Stage II: Detection of the Eye Centers in the Context of Face

The first stage of the eye center detection system is specialized in finding general regions of the image plane that are highly likely to contain eyes. The output of the system is very resistant to false alarms but does not specify well the precise location of the eyes. The second stage is specialized in achieving high accuracy provided it operates on the regions selected by the previous stage. This stage uses the same searching techniques as that of the previous stage: all patches at multiple scales, within a sub-region of the face restricted in both location and scale, are submitted to a boosted classifier which returns the eye versus non-eye log-likelihood ratio. This log-likelihood ratio is then combined with the prior probability of eye given the location and size with respect to the face detection window to produce a final log posterior ratio of eye versus non-eye.

For training the classifier of the second stage towards detecting the centers of the left and right eye, total 4826 positive eye examples and 10,000 non-eye examples have been used. Positive examples were selected by cropping patches from the images such that they contain eyes at a canonical scale and location with respect to their face (described next), and scaling the patches to 24×24 pixels. Non-eye examples were taken from the same images at multiple non-eye locations and scales within the faces, with constraints described below.

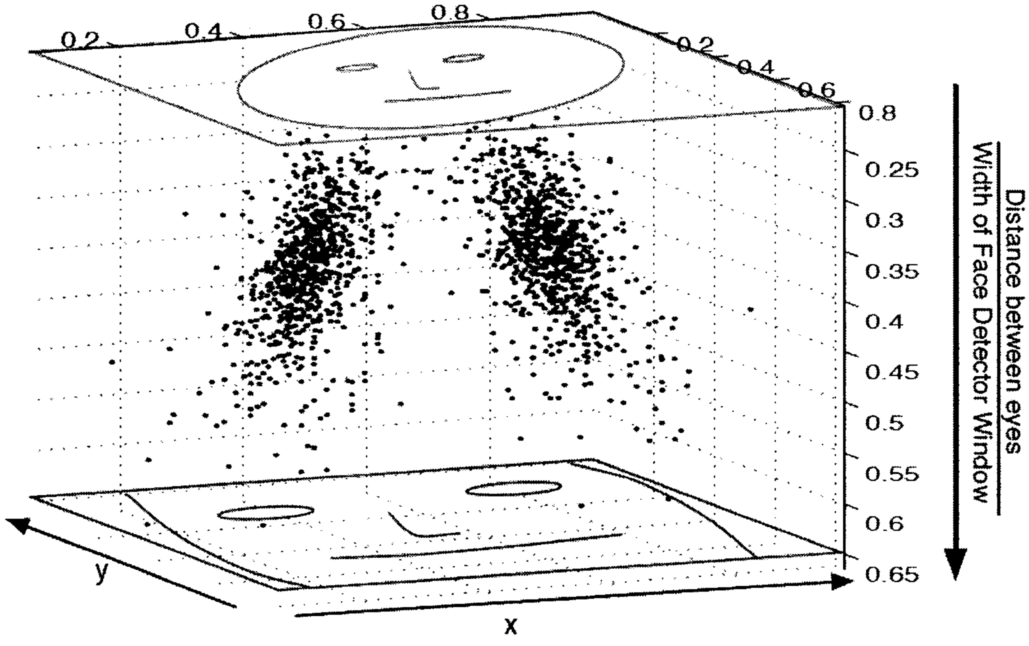


Figure 14: The face detection window can vary from closely cropping the face (negative z-axis) to loosely cropping the face (positive z-axis). The points show typical eye locations relative to the face detection window over a sample database of face images. This variability has been modeled with a three-dimensional Gaussian in [92], where the x-axis and the y-axis are space, and the z-axis is scale, i.e., ratio of distance between eyes to size of the face-detector window. This has been used to model the prior probability of a location containing an eye given the face detection window.

To crop and center the eye patches for training, four variables (d, r, t and q) have been used where, d: the distance between the eyes, r: the ratio of the distance between the center of the eyes and the left and upper edges of the face cropping window, t: an offset parameter and q: a scale parameter [92]. Positive training samples were then prepared by cropping example images such that $r = q(d + td)$ and scaling them to 24 × 24 pixel. In other words, the size of the window was chosen to be proportional to the distance between the eyes, and could be off center by some fixed amount. Thus, a small value of q results in a small receptive field with high resolution and a large value of q results in a large receptive field with relatively low resolution, while t shifts the location of the eye with respect to the center of the patch.

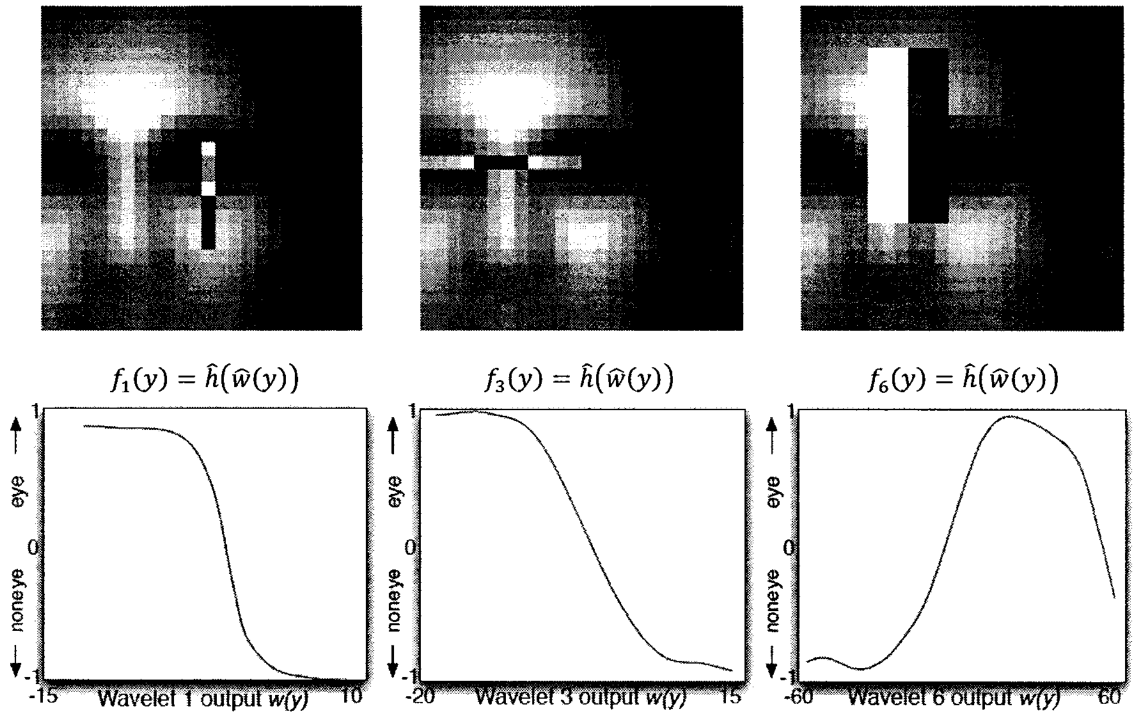


Figure 15: The first, third, and sixth wavelets (top) and their respective tuning curves (bottom) for the left eye detector centered on the eye with scale factor $q = 1$ [92]. Each wavelet is shown over the average positive (eye) example. The tuning curves show the evidence for eye (high) vs. non-eye (low) as the wavelet output increases (shown increasing from left to right). The first tuning curve shows that a dark vertical region over a bright vertical region in the center of the window is evidence for an eye and for non-eye otherwise. The middle tuning curve looks for a horizontal band that goes dark-light-dark towards the left of the window as evidence for an eye, which appears to be testing for the bridge of the nose. The rightmost wavelet also can be interpreted as a bridge of the nose detector, however it also indicates that too much difference between the left and right parts of the wavelet are evidence against eye.

From the situational inference approach, it might be expected that pixels which are generated by the background contain relatively little additional information once it is known that searching for eyes is being done within a face. Thus t and q should be chosen such that they maximizes the number of pixels in the positive example patches that are generated by face, i.e., about the size of the face and centered on the center of the face, so that very few background pixels enter into the window. However, given a fixed input size of 24 × 24, it is possible that smaller values of q, such as one that just covers the eye (resulting in higher resolution examples with less surrounding context) allows to be benefited maximally from the information in pixels generated by the eye only. Fasel et al. [92] tested the effect of the size and location of the receptive field used for eye detection, where receptive field size was expressed as the ratio q of the distance between the eyes and location was expressed as "face-centered" or "eye-centered". They found that varying patch size from small enough to cover just the iris ($q = 0.11$) to large enough towards covering an area four times the size of the head ($q = 2.5$) results in a U-shaped curve, with the best performance coming from the patches having size $q = 1$, which covers about 80% of the face. The best centering condition was "eye-centered". So, $q = 1$ and the centering condition "eye-centered" have been used in this work to obtain the best optimized performance of the eye center detector.

The situational inference approach described in sub-section 3.3.2.3 also allows to constrain how the non-eye examples are to be chosen: prior belief about the eye location π is modeled as a normal distribution with parameters for the mean and standard deviation of the true eye position and scale with respect to the window chosen by the face detector, as measured against the training set. Distribution of the locations of eyes with respect to the size of the face detection window for some example data is shown in Figure 14. The vertical axis (downwards) shows the increasing ratio of the size of the face detection window to the distance between the eyes. When the face detector selects a small window relative to the true face size resulting in a small detection width to eye distance ratio, the eyes tend to be far apart with respect to the detection window. When the face detector selects a large window compared to the distance between the eyes, the eyes tend to be located closer together near the center of the detection window. Using these statistics about the true eye positions with respect to the estimated face location, the set of patches for searching as well as training is restricted to a maximum Mahalanobis distance M from the mean location and scale of each eye. Choosing $M = 16.27$ gives a 99.9% confidence interval for one of the patches containing the eye [92].

Using these criteria, two positive training examples (one for each eye) and six negative training examples were created for each example face, where the negative examples were selected randomly from the set of patches satisfying the maximum distance from the mean eye patch size and location criterion. To make the best use of the data, the positive and negative examples from the right eye were flipped about the horizontal axis and were combined with the left eye examples to train a single left eye detector. Then this left eye detector was flipped about the horizontal axis to get a right eye detector. Once the set of positive and negative examples are collected, the stage of the situational inference cascade is trained with GentleBoost as described above. Figure 15 shows example wavelets and their corresponding tuning curves for the best eye-detector.

Before incorporating the eye center detection system as a part of the proposed automated system for facial expression classification, its reliability has been tested extensively. First of all, 170 face images were collected from the internet and eye center detection was performed over them applying the previously discussed eye detection method. In this experiment, successful detection was recorded for 168 images. Efficiency of the eye detection method was further tested at the presence of different facial expressions using images from the Japanese Female Facial Expression (JAFFE) Database [100] and the Cohn-Kanade AU-Coded Database of Facial Expression [101]. In case of the JAFFE database, successful eye center detection was performed on 212 images out of 213. From the Cohn-Kanade AU-Coded Database of Facial Expression, randomly selected 200 images were used for the purpose of verification and the eye detection system located the eye centers successfully from 198 images. The high detection accuracy as well as preciseness observed by performing experiments with different test cases, qualify the eye detection method to be integrated with the proposed system for automated facial expression classification.

3.3.3. Fixing the Horizontal Rotation of Face in the Face Image

After the right and left eye center (P_1 and P_2) are detected, the amount of horizontal rotation (rotation over the x-axis) of face in the input image is determined. For this purpose, a right angled triangle is imagined using the right eye center (x_1, y_1), the left eye center (x_2, y_2) and the third point that serves as the crossing point of the horizontal and vertical lines passing through the right and left eye centers respectively (Figure 16).

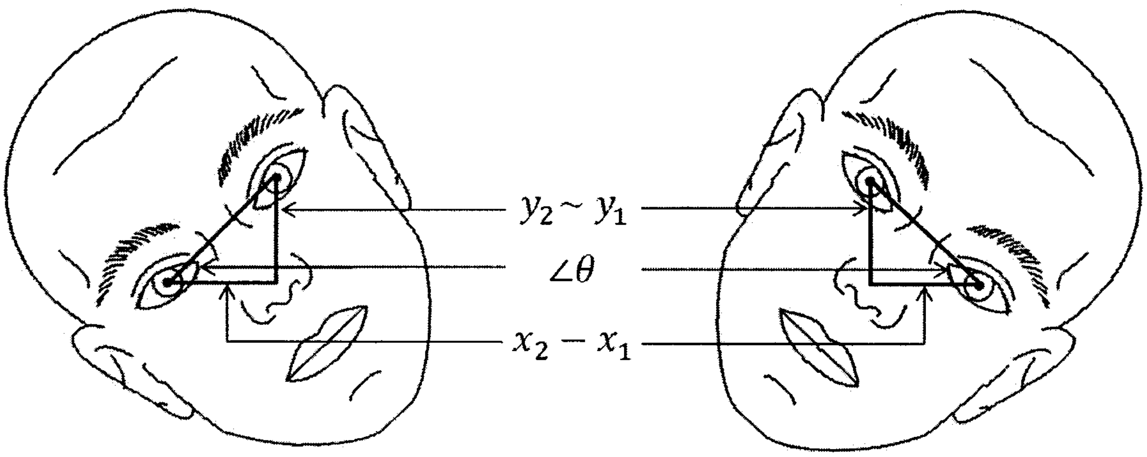


Figure 16: Calculation of the rotation angle θ for fixing the horizontal rotation (rotation over the x-axis) of face in the input face image.

In spatial coordinate system, each location in an image corresponds to its respective position on a plane, and is described in terms of the x and y coordinate values, where x increases from left to right and y increases from upwards to downwards. Using the coordinate values of the left and right eye centers, the amount of horizontal rotation θ of the face is then determined as:

$$\theta = \tan^{-1}\left(\frac{|y_1 - y_2|}{x_2 - x_1}\right) \tag{3.19}$$

For fitting the face with the proposed anthropometric face model described in Section 3.2, the whole image is rotated by the angle θ. The direction of the rotation (clock-wise or counter clock-wise) is determined by the polarity of the difference between the vertical distances of the right and left eye center $(y_1 - y_2)$. If the difference is positive, the image is rotated in clock-wise direction, otherwise the rotation is performed in counter clock-wise direction. The new locations of the right eye center (x_1, y_1) and the left eye center (x_2, y_2) are then updated by detecting them once again over the rotated face image.

3.3.4. Determining the Rectangular Facial Feature Regions

To reduce the search space of the fifteen feature points used in defining the analytic face model of feature extraction for facial expression classification (please refer to Section 2.4), approximation of the rectangular facial feature regions are necessary that will confine the search space of the feature points corresponding to each of the facial feature regions (eye, eyebrow, nose and mouth). For this purpose, approximated center points of these feature regions need to be identified. After the orientation of face in the input face image is fixed, the centers of the right and left eyes (P_1 and P_2) are determined using the eye center detection technique specified in sub-section 3.3.2. Midpoint of the eye centers (P_5) is then calculated as:

$$(x_{P_5}, y_{P_5}) = \left(\frac{x_{P_1} + x_{P_2}}{2}, \frac{y_{P_1} + y_{P_2}}{2}\right) \qquad (3.20)$$

As demonstrated in Figure 5, centers of the eyebrows (P_3 and P_4), nose (P_6) and mouth (P_7) are then calculated using the distance between the centers of the eyes (D_1) as the measuring criteria and applying the anthropometric proportional values of human face listed in Table 4. Confined regions for searching the specific feature points around the eyebrows, eyes, nose and mouth are then set by the rectangles defined according to Table 5. While defining the boundary of each of the facial feature regions, it has also been ensured through practical experiment that the rectangular regions surround the facial feature component entirely within its boundary.

Table 5: Dimensions of the rectangular facial feature regions defined using the distance between the center of the right and left eye (D_1).

Facial Feature Region	**Width of the Rectangle**	**Height of the Rectangle**
Eye	1.0×D_1	0.7×D_1
Eyebrow	1.2×D_1	0.6×D_1
Nose	0.8×D_1	0.6×D_1
Mouth	1.5×D_1	1.0×D_1

Chapter 4

Detection of the Facial Feature Points

Searching for the fifteen feature points that constitutes the analytic face model of feature extraction (discussed in Section 2.4) is done within the isolated facial feature regions obtained by applying the procedure discussed in Chapter 3. Instead of relying on a single image processing based method to obtain the feature points from the regions of eyebrow, eye, nose and mouth; a multi-detector approach [43] based on different standard image processing technique has been used. The motivation is that, the facial regions, where searching operation is performed for detecting the fifteen feature points differ significantly with respect to their inherent geometric pattern, intensity distribution as well as internal scene complexity. Moreover, shapes of these facial feature components demonstrate high level of variability in the presence of different facial expressions. Hence, it is not possible to obtain the best result in detecting these points of interest by applying just a single image processing based feature point detection technique over all the facial regions [37]. So, the multi-detector approach proposed in [43] has been applied in this work, which has demonstrated quite satisfactory performance in constituting the analytic face model through identifying the fifteen facial feature points. The detailed procedure for detecting these feature points from the regions of eyebrow, eye, nose and mouth has been elaborated in the subsequent discussion of this chapter.

4.1. Detection of Feature Points from the Eyebrows

Aside from the dark colored eyebrow, the eyebrow regions also contain relatively bright skin portion and are sometimes occulted partially with hair. Considering the darker pixels as the background, the original eyebrow image is complemented to convert the eyebrow region as the foreground object and the rest as background (Figure 17.b). The following formula is used for performing complementation of the grayscale image:

$$Q(i,j) = 255 - P(i,j) \tag{4.1}$$

Here $P(i,j)$ is the input grayscale image and $Q(i,j)$ is the complemented image.

A morphological image opening operation is then performed over the complemented image with a disk shaped structuring element of ten pixel radius for obtaining the background illumination (Figure 17.c). Let $f(x,y)$ is the input image and $b(x,y)$ is the structuring element. The opening of the grayscale image f by sub-image (structuring element) b is denoted as $(f \circ b)$ and is defined by [102]:

$$f \circ b = (f \ominus b) \oplus b \tag{4.2}$$

Here $(f \ominus b)$ denotes the erosion of image f by the structuring element b and $\{(f \ominus b) \oplus b\}$ denotes the subsequent dilation of the eroded image by the same structuring element. The erosion and dilation of grayscale image are defined as:

$$(f \ominus b)(s,t) = \min\{f(s+x, t+y) - b(x,y) \,|\, (s+x),(t+y) \in D_f; (x,y) \in D_b\} \tag{4.3}$$

$$(f \oplus b)(s,t) = \max\{f(s-x, t-y) + b(x,y) \,|\, (s-x),(t-y) \in D_f; (x,y) \in D_b\} \tag{4.4}$$

where D_f and D_b are the domains of f and b respectively.

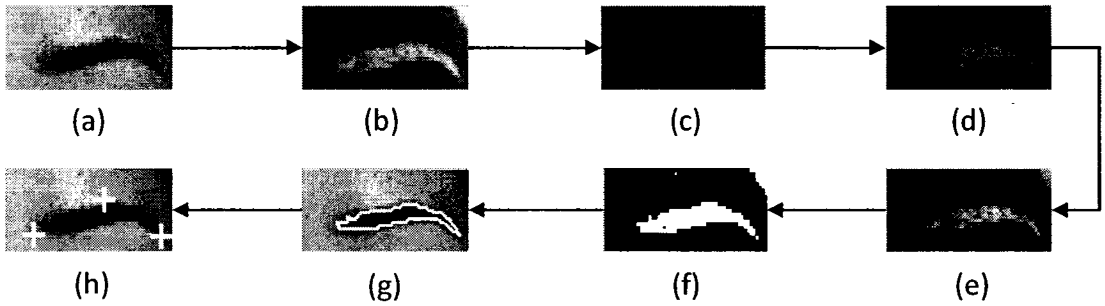


Figure 17: Detection of the inner and outer eyebrow corners, and the upper midpoint of the eyebrow (a) eyebrow region (b) complemented eyebrow image (c) estimated background (d) background subtraction (e) intensity adjustment (f) binary eyebrow region (g) isolated eyebrow contour (h) detected eyebrow corners and the upper midpoint of the eyebrow.

The image opening operation eliminates the eyebrow from the original image leaving just an estimation of the background. The estimated background is then subtracted from the complemented image to have a comparatively brighter eyebrow over a uniform dark background (Figure 17.d). Intensity of the resultant image is then adjusted on the basis of the pixels' cumulative distribution to increase the discrimination between the foreground and the background (Figure 17.e). For this purpose, cumulative distribution of the pixels' intensity values is calculated from the intensity histogram of the grayscale image. Using this distribution, the gray levels of the image histogram that contributes for the top 30% of the cumulative distribution are identified. The pixels of the original eyebrow image that falls within these gray level regions are then stretched up towards the highest intensity value of the image. Similarly, the gray levels that contribute for the lower 30% of the pixels' cumulative distribution are identified and intensity values of the pixels that fall within these gray levels are reduced towards the lowest intensity value of the image.

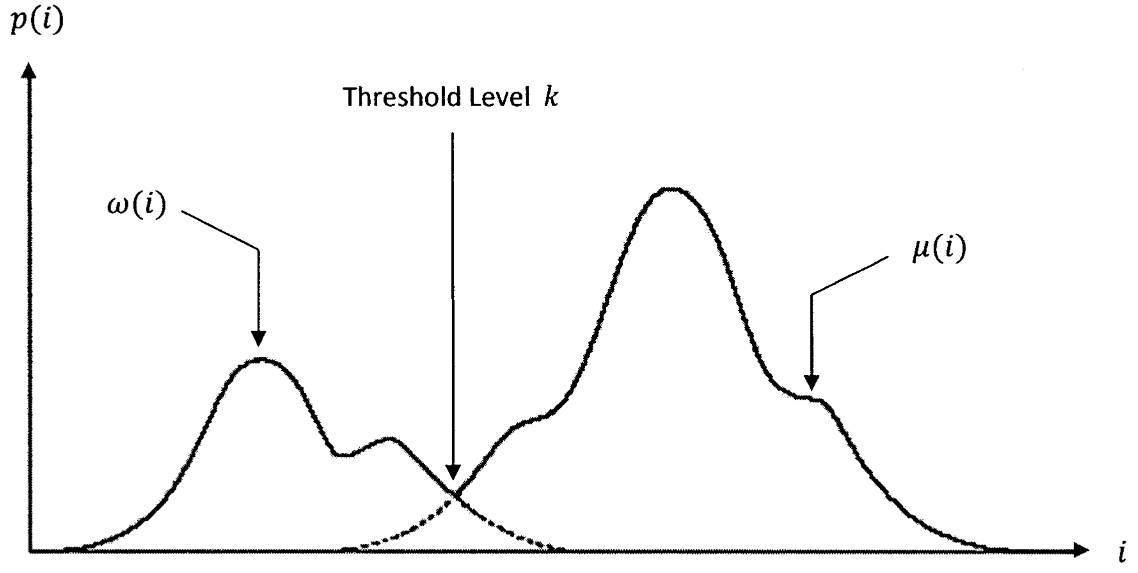


Figure 18: Splitting up the histograms of the object and background to obtain the appropriate threshold level k that maximizes the relation $\frac{(\mu_T\omega(k)-\mu(k))^2}{\omega(k)\mu(k)}$. Here μ_T is the image average.

The next step is to obtain the binary version of this intensity adjusted image. For this purpose, an appropriate threshold value has been calculated in an automated manner applying the technique of threshold selection proposed by N. Otsu [103]. In general, the individual histograms of the object and the background overlap (Figure 18) and without some prior knowledge of the individual histograms, it may be difficult to find a splitting point that can separate the object from the background. Otsu's method of threshold selection chose the threshold value by finding out this splitting point as the point where the two histograms crossover. For this purpose, the image histogram is described in terms of probability distribution as:

$$p_i = \frac{n_i}{N} \tag{4.5}$$

where n_i is the number of pixels with gray level i, N is the total number of pixels so that p_i is the probability of a pixel having gray level i. If the thresholding is done at level k, the probability that a pixel will have threshold value $\leq k$ can be defined as:

$$\omega(k) = \sum_{i=0}^{k} p_i \tag{4.6}$$

Similarly, the probability of a pixel's having threshold value $> k$ can be expressed as:

$$\mu(k) = \sum_{i=k+1}^{L-1} p_i \tag{4.7}$$

where L is the number of grayscales with the largest grayscale value $(L-1)$. By definition of probability, equation (4.6) and (4.7) can be combined as:

$$\omega(k) + \mu(k) = \sum_{i=0}^{L-1} p_i = 1 \tag{4.8}$$

The objective is to find the value of k that maximize the difference between $\omega(k)$ and $\mu(k)$ towards providing a good separation boundary between the object and the background. According the method describe by Otsu [103], this can be done by first defining the image average as:

$$\mu_T = \sum_{i=0}^{L-1} i p_i \tag{4.9}$$

and then finding a value k that maximizes the following expression:

$$\frac{\left(\mu_T \omega(k) - \mu(k)\right)^2}{\omega(k)\mu(k)} \tag{4.10}$$

Once the appropriate threshold value k is obtained, the binary version $g(x, y)$ of the original grayscale image $f(x, y)$ is as (Figure 17.f):

$$g(x,y) = \begin{cases} 1, & f(x,y) > k \\ 0, & f(x,y) \leq k \end{cases} \tag{4.11}$$

The binary image contains the eyebrow as well as some other noisy patterns created due the impact of the image pre-processing methods applied before. For detecting the three specific feature points (inner eyebrow corner, outer eyebrow corner and upper midpoint of the eyebrow) from the eyebrow region; it necessary to identify all the points that lie over the eyebrow contour. So, all the available contours of the binary image are detected using the 8-connected chain code based contour following algorithm specified in [104] and [105].

The algorithm finds out the contour of an object by walking around the boundary and noting down the directions of which way to turn at a specific point for staying over the boundary. At each point of the boundary, there are eight possible directions to choose from towards tracing the objects boundary (Figure 19).

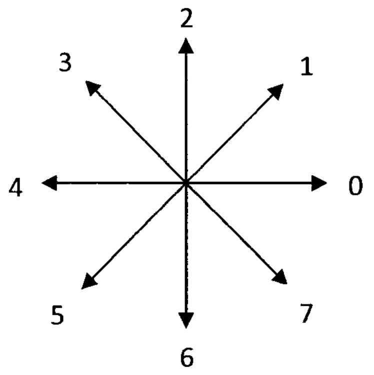


Figure 19: Possible directions for the 8-connected chain code based contour following algorithm.

Working procedure of the 8-connected chain code based contour following technique is described below:

Step 1: Start by finding the pixel in the object that has the left-most value in the topmost row; call this pixel P_0. Define a variable dir (for direction), and set it equal to 7 (Since P_0 is the top left pixel in the object, the direction to the next pixel must be 7).

Step 2: Traverse the 3×3 neighborhood of the current pixel in a counter-clockwise direction, beginning the search at the pixel in direction:

dir + 7 (mod 8), if dir is even

dir + 6 (mod 8), if dir is odd

This simply sets the current direction to the first direction counter clock-wise from dir:

dir	0	1	2	3	4	5	6	7
dir + 7 (mod 8)	7	0	1	2	3	4	5	6
dir + 6 (mod 8)	6	7	0	1	2	3	4	5

The first foreground pixel will be the new boundary element. Update dir.

Step 3: Stop when the current element of the contour P_n is equal to the second element P_1 and the previous pixel of the contour P_{n-1} is equal to the first element P_0

The eyebrow contour, which is usually the largest one, is then identified by calculating the area covered by each of the contours (Figure 17.g). For the left eyebrow, the point over the contour having the minimum values along the x and y coordinates simultaneously is considered as the inner eyebrow corner, and the point that has the maximum values along both the x and y coordinates is considered as the outer eyebrow corner (Figure 17.h). Similarly, for the right eyebrow, the point over the eyebrow contour that has the maximum values along the x-axis and minimum value along the y-axis simultaneously is considered as the inner eyebrow corner, and the point which has the minimum value along the x-axis and maximum value along the y-axis simultaneously is considered as the outer eyebrow corner. Once the inner and outer corners of both of the eyebrows are determined, the x coordinate values of their midpoints are calculated as:

$$x_{mid} = \frac{x_{inner} + x_{outer}}{2} \tag{4.12}$$

Since, the coordinates of the contour points cannot be represented using fractional values, the resultant values of x_{mid} are rounded to the nearest integer. The midpoint of each of the eyebrows is then identified as the point over the eyebrow contour that has the same x coordinate value as that of x_{mid} but the minimum value along the y coordinate (Figure 17.h).

4.2. Feature Point Detection from the Eyes

The eye region is composed of the dark upper eyelid with the eyelash, lower eyelid, pupil, bright sclera and the skin region that surrounds the eye. The most continuous and non-deformable part of the eye region is the upper eyelid, since both the pupil and the sclera change their shapes with various possible situations of the eyes, particularly when the eye is closed or partially closed due to various facial expressions. So, the inner and the outer eye corners are determined first by analyzing the shape of the upper eyelid and used later on for locating the mid upper and mid lower eyelids. To avoid the erroneous detection of the eye feature points, any discontinuity in the upper eyelid region must be avoided by changing the illumination of the upper eyelid so that it differs significantly from the surrounding region. This has been carried out by saturating the intensity values of all the pixels towards zero that constitutes the lower 50% of the image intensity cumulative distribution (Figure 20.b). The adjusted image is then converted to its binary version (Figure 20.c) using the threshold value obtained applying the iterative procedure of threshold selection [106], [107].

The iterative method of threshold selection starts with an initial guess of the threshold and refines this estimate by successive passes through the image. The initial guess can be the mean grey level of the image [107], an average of the mean grey level of the corner pixels, or the mean grey level of all other pixels in the image [106]. The later one assumes that the corner pixels represent the background, rather than object of interest. Four to ten iterations are usually sufficient for the convergence of the algorithm used for calculating the threshold value using the iterative procedure. Details of the iterative method for automated threshold selection are summarized as follows:

1) Pick an initial threshold value T.
2) Segment the image using T. This will produce two groups of pixels: G_1 consisting of all the pixels with gray level values $> T$ and G_2 consisting of all the pixels having gray level values $\leq T$.
3) Compute the average gray level values μ_1 and μ_2 for the pixels in regions G_1 and G_2.
4) Calculate the new threshold T_{new} as:

$$T_{new} = \frac{\mu_1 + \mu_2}{2}$$

5) If the threshold is stabilized $(T = T_{new})$, this is the appropriate threshold level. Otherwise, T becomes T_{new} and re-iterate from step 2.

The contour that covers the largest area is then isolated (Figure 20.d) applying the 8-connected chain code based contour following the algorithm specified in [104] and [105]. For the right eye, the inner eye corner is the rightmost point of the contour and outer eye corner is the leftmost point of the contour (Figure 20.e).

For the left eye, the rightmost point over the contour becomes the outer corner and the leftmost point becomes the inner corner. The whole eye contour region is then divided vertically into three equal parts and searching for the upper and lower mid eyelid is then done within the mid division. For each value of the x coordinate $\{x_1, x_2, \cdots, x_n\}$ that falls within this mid division, there will be two values of the y coordinate: one from the upper portion of the eye contour $\{y_{11}, y_{12}, \cdots, y_{1n}\}$ and another from the lower portion of the eye contour $\{y_{21}, y_{22}, \cdots, y_{2n}\}$. Distances between each pair of such points $\{(x_i, y_{1i}), (x_i, y_{2i})\}$ are then calculated. The maximum of the distances, which has been calculated from such pair of points and is closest to the midpoint of inner and outer eye corner, is considered as the amount of eye opening and provides the mid points of the upper lower eyelids respectively (Figure 20.f).

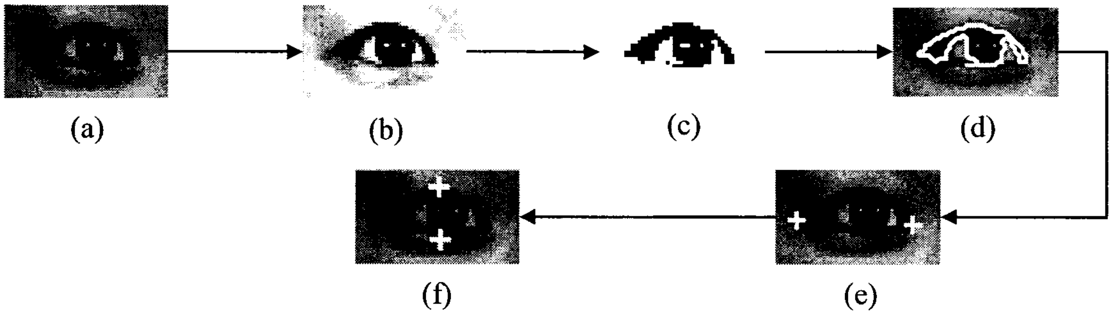


Figure 20: Mid upper and lower eyelid detection (a) eye region (b) intensity adjustment (c) binarization (d) isolated eye contour (e) inner and outer eye corner detection (f) detected mid points of the upper and lower eyelids.

4.3. Detection of the Midpoint of the Nostrils

Nostrils of the nose region are the two circular or parabolic objects having the darkest intensity level (Fig. 22.a). So, in order to identify the center points of the nostrils, it is necessary to separate them first from the nose region. For this purpose, the isolated nose region is filtered using the Laplacian of Gaussian (LoG) as the filter proposed by Marr and Hildreth [108].

Laplacian of Gaussian (LoG) filtering consists of convolving the image first using a Gaussian smoothing kernel, which is then followed by computing the Laplacian operator. The continuous domain LoG gradient can be expressed as [109]:

$$LoG(x,y) = -\nabla^2\{F(x,y) \circledast G_\sigma(x,y)\} \tag{4.13}$$

where $\circledast$ is the convolution operator, $F(x,y)$ is the input image, $H_\sigma(x,y)$ is the 2-D Gaussian smoothing kernel defined as:

$$G_\sigma(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{4.14}$$

and ∇^2 is the Laplacian operator defined by:

$$\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \tag{4.15}$$

As a result of the linearity of the second derivative operation as well as the convolution, the 2-D LoG function centered on zero and with Gaussian standard deviation σ can be expressed as [109]:

$$LoG(x,y) = -\frac{1}{\pi\sigma^4}\left[1 - \frac{x^2+y^2}{2\sigma^2}\right] e^{-\left(\frac{x^2+y^2}{2\sigma^2}\right)} \tag{4.16}$$

The benefits of using the LoG filter is that the size of the kernel used for the purpose of filtering is usually much smaller than the image, and thus requires far fewer arithmetic operations. The kernel can also be pre-calculated in advance and so only one convolution is needed to be performed at the run-time over the image. In addition, Laplacian of Gaussian is free from the problems that are usually observed while using the Laplacian alone. As a second order derivative, the Laplacian is unacceptably sensitive to noise. Magnitude of the Laplacian produces double edges, an undesirable effect which complicates segmentation. Finally, the Laplacian is unable to detect edge direction. As the Laplacian is combined with the Gaussian smoothing as a precursor towards forming the Laplacian of Gaussian (LoG), the limitations of Laplacian specified above are eliminated [102]. The three dimensional visualization of a Laplacian of Gaussian function centered on zero with Gaussian standard deviation σ is provided in Figure 21.

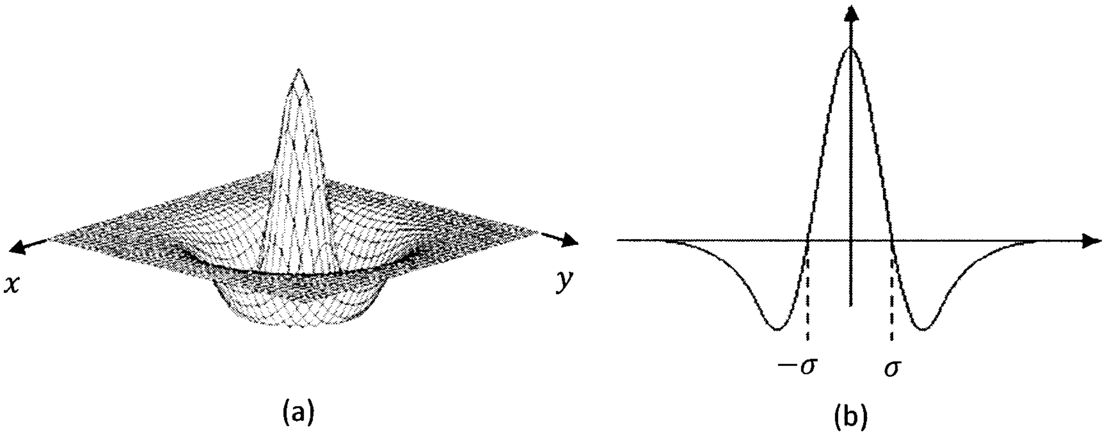


Figure 21: Visualization of the Inverted Laplacian of Gaussian (LoG) function (a) 3-D plot of the LoG operator (b) cross section showing the zero crossing.

The LoG operator calculates the second spatial derivative of an image. This means that in areas where the image has a constant intensity (i.e. where the intensity gradient is zero), the LoG response will be zero [102]. However, in the vicinity of a change in intensity, the LoG response will be positive on the darker side, and negative on the lighter side. This means that at a reasonably sharp edge between two regions of uniform but different intensities, the LoG response will be zero at a long distance from the edge as well as positive just to the one side of the edge and negative to the other side. As a result, the intensity of the filtered binary image gets complemented and changes the nostrils as the brightest part of the image (Figure 22.b). Searching for the local maximal peak is then performed within the filtered image to obtain the centre points of the nostrils. To make the nostril detection technique independent of the image size, the whole process is repeated varying the filter size starting from ten pixels until the number of peaks of the local maxima is reduced to two (Figure 22.c). Midpoints of the nostrils are then calculated by averaging the coordinate values of the identified nostrils (Figure 22.d).

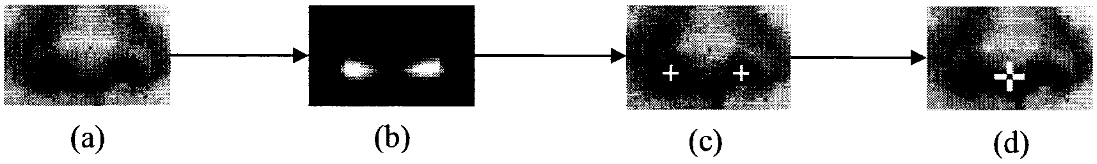


Figure 22: Detection of the midpoint of nostrils (a) isolated nose region (b) filtered by Laplacian of Gaussian (LoG) (c) detected nostrils by finding the centers of local maxima (d) detected midpoint of the nostrils.

4.4. Feature Point Extraction from the Mouth

As defined by the proposed analytic face model of feature extraction for facial expression classification (Section 2.4), four points namely, left mouth corner, right mouth corner, top midpoint of the upper lip and lower midpoints of the bottom lip have to be identified from the face region. So, the contour of the mouth has to be detected first from where, the specified four feature points can be identified simply by analyzing the coordinate values of the points that belong to the mouth contour. But variability emerged from situations like individual appearance (spatial variability), locutions as well as facial expressions (temporal variability) and lighting (spatiotemporal variability) makes the task of lip contour detection quite complicated and difficult. Since, the variation of Level Set method proposed by Li et al. [5] shows significant robustness against the temporal and spatiotemporal variability in image segmentation [58], it has been adopted in this work for the purpose of detecting the lip contour from the isolated mouth region.

Level set methods were introduced first by Osher and Sethian [110] for capturing moving fronts. In the level set formulation of moving fronts (or active contour), the fronts, denoted by C, are represented by the zero level set $C(t) = \{(x,y)|\phi(t,x,y) = 0\}$ of a level set function $\phi(t,x,y)$. The evolution equation of the level set function ϕ can be written as:

$$\frac{\partial\phi}{\partial t} + F|\nabla\phi| = 0 \tag{4.17}$$

which is known as the level set equation [110]. The function F is called the speed function which depends on the image data and the level set function ϕ for image segmentation. In traditional level set methods [111-114], the level set function ϕ can develop shocks, very sharp and/or flat shape during the evolution, which makes further computation highly inaccurate. To avoid these problems, a common numerical scheme is to initialize the function ϕ as a signed distance function before the evolution, and then "reshape" (or "re-initialize") the function ϕ to be a signed distance function periodically during the evolution. The standard re-initialization method involves solving the following re-initialization equation:

$$\frac{\partial \phi}{\partial t} = sign(\phi_0)(1 - |\nabla \phi|) \tag{4.18}$$

where ϕ_0 is the function to be re-initialized, and $sign(\phi)$ is the sign function. Various approaches have already been proposed for performing this re-initialization and most of them are the different variants of the above PDE-based method [115], [116]. Unfortunately, if ϕ_0 is not smooth or ϕ_0 is much steeper on one side of the interface than the other, the zero level set of the resulting function ϕ can be moved incorrectly from that of the original function [114], [115], [117]. Moreover, when the level set function is far away from a signed distance function, these methods may not be able to re-initialize the level set function to a signed distance function. In practice, the evolving level set function can deviate greatly from its value as a signed distance in a small number of iterations, especially when the time step is not chosen small enough. So, from a practical viewpoint, the re-initialization process can be quite complicated, expensive, and may have subtle side effects. Moreover, most of the level set methods are fraught with their own problems, such as when and how to re-initialize the level set function to a signed distance function. The variational level set formulation proposed in

[5] is considered as a solution to these problems and can be easily implemented using the simple finite difference scheme without the need of re-initialization.

4.4.1. Variational Level Set Formulation of Curve Evaluation without Re-Initialization

In this section, formulation of the variational level set method of curve evaluation without re-initialization will be discussed towards its implementation in lip contour detection. At first, formulation of the general variational level set method with penalizing energy will be elaborated, which will be followed by the discussion on the variational level set formulation of active contours without re-initialization.

4.4.1.1. General Variational Level Set Formulation with Penalizing Energy

It is well known that a signed distance function must satisfy a desirable property of $|\nabla\phi| = 1$. Conversely, any function ϕ satisfying $|\nabla\phi| = 1$ is the signed distance function plus a constant [118]. Since it is crucial to keep the evolving level set function as an approximate of the signed distance function during the evolution, especially in a neighborhood around the zero level set; the following integral has been proposed in [5] as a metric to characterize how close a function ϕ is to a signed distance function in $\Omega \subset \Re^2$:

$$\mathcal{P}(\phi) = \int_{\Omega} \frac{1}{2}(|\nabla\phi| - 1)^2 \, dx \, dy \tag{4.19}$$

This metric plays a key role in the formulation of the variational level set method. With the above defined functional $\mathcal{P}(\phi)$, variational formulation has been proposed as [5]:

$$\mathcal{E}(\phi) = \mu\mathcal{P}(\phi) + \mathcal{E}_m(\phi) \tag{4.20}$$

where $\mu > 0$ is a parameter for controlling the effect of penalizing the deviation of ϕ from a signed distance function, and $\mathcal{E}_m(\phi)$ is a certain energy that would drive the motion of the zero level curve of ϕ. Here, the first variation (or *Gateaux* derivative) of the functional $\mathcal{E}$ is denoted by $\frac{\partial \mathcal{E}}{\partial \phi}$ and the gradient flow that minimizes the functional $\mathcal{E}$ is defined as [119]:

$$\frac{\partial \phi}{\partial t} = -\frac{\partial \mathcal{E}}{\partial \phi} \tag{4.21}$$

For a particular functional $\mathcal{E}(\phi)$ defined explicitly in terms of ϕ, the Gateaux derivative can be computed and expressed in terms of the function ϕ and its derivatives [119]. The variational formulation in (4.20) has been applied to the active contour for image segmentation so that the zero level curves of ϕ can evolve to the desired features in the image. For this purpose, the energy $\mathcal{E}_m$ has been defined as a functional that depends on the image data, and is therefore called the *external energy*. Accordingly, the energy $\mathcal{P}(\phi)$ is called the *internal energy* of the function ϕ, since it is a function of ϕ only. During the evolution of ϕ according to the gradient flow (4.21) that minimizes the functional (4.20), the zero level curve is moved by the external energy $\mathcal{E}_m$. Meanwhile, due to the penalizing effect of the internal energy, the evolving function ϕ is automatically maintained as an approximate of the signed distance function during the evolution according to the equation (4.21). Therefore, the re-initialization procedure is completely eliminated in the variational level set method proposed in [5]. This concept has been demonstrated further in the context of active contours in the next subsection.

4.4.1.2. Variational Level Set Formulation of Active Contours without Re-Initialization

Image segmentation is performed by following the active contour which is explicitly defined as the dynamic curves that moves toward the object boundary. To achieve this goal, an external energy term is defined that can move the zero level curve toward the object boundaries. Let I be an image, and g be the edge indicator function defined by:

$$g = \frac{1}{1 + |\nabla G_\sigma * I|^2} \tag{4.22}$$

where G_σ is the Gaussian kernel with standard deviation σ. For a function $\phi(x, y)$, the external energy is defined as:

$$\mathcal{E}_{g,\lambda,v}(\phi) = \lambda \mathcal{L}_g(\phi) + v\mathcal{A}_g(\phi) \tag{4.23}$$

where $\lambda > 0$ and v are constants, and the terms $\mathcal{L}_g(\phi)$ and $\mathcal{A}_g(\phi)$ are defined by:

$$\mathcal{L}_g(\phi) = \int_\Omega g\delta(\phi)|\nabla\phi|\, dx\, dy \tag{4.24}$$

$$\mathcal{A}_g(\phi) = \int_\Omega gH(-\phi)\, dx\, dy \tag{4.25}$$

where δ is the univariate Dirac function, and H is the Heaviside function. The total energy function $\mathcal{E}(\phi)$ is then defined in terms of the internal and external energy terms as:

$$\mathcal{E}(\phi) = \mu\mathcal{P}(\phi) + \mathcal{E}_{g,\lambda,v}(\phi) \tag{4.26}$$

The external energy $\mathcal{E}_{g,\lambda,v}$ drives the zero level set toward the object boundaries, while the internal energy $\mu\mathcal{P}(\phi)$ penalize the deviation of ϕ from a signed distance function during its evaluation.

To understand the geometric meaning of the energy $\mathcal{L}_g(\phi)$, it can be assumed that the zero level set of ϕ can be represented by a differentiable parameterized curve $C(p)$, where $p \in [0,1]$. It is well known that the energy functional $\mathcal{L}_g(\phi)$ in (4.24) computes the length of the zero level curve of ϕ in the conformal metric $ds = g\big(C(p)\big)|C'(p)|dp$ [120]. The energy functional $\mathcal{A}_g(\phi)$ in (4.25) is introduced to speed up the curve evolution. It should be noted that, when the function g is constant 1, the energy functional in (4.25) is the area of the region $\Omega_\phi^- = \{(x,y)|\phi(x,y) < 0\}$ [114]. The energy functional $\mathcal{A}_g(\phi)$ in (4.25) can be viewed as the weighted area of Ω_ϕ^-. The coefficient v of $\mathcal{A}_g$ can be positive or negative, depending on the relative position of the initial contour to the object of interest. For example, if the initial contours are placed outside the object, the coefficient v in the weighted area term should take positive value, so that the contours can shrink faster. If the initial contours are placed inside the object, the coefficient v should take negative value to speed up the expansion of the contours.

By calculus of variations [119], the Gateaux derivative (first derivation) of the functional $\mathcal{E}$ in (4.26) can be written as:

$$\frac{\partial \mathcal{E}}{\partial \phi} = -\mu\left[\Delta\phi - div\left(\frac{\nabla\phi}{|\nabla\phi|}\right)\right] - \lambda\delta(\phi)div\left(g\frac{\nabla\phi}{|\nabla\phi|}\right) - vg\delta(\phi) \tag{4.27}$$

where Δ is the Laplacian operator. Therefore, the function ϕ that minimizes this functional satisfies the Euler-Lagrange equation $\frac{\partial \mathcal{E}}{\partial \phi}$. The steepest descent process for minimization of the functional $\mathcal{E}$ is the following gradient flow:

$$\frac{\partial \phi}{\partial t} = \mu\left[\Delta\phi - div\left(\frac{\nabla\phi}{|\nabla\phi|}\right)\right] + \lambda\delta(\phi)div\left(g\frac{\nabla\phi}{|\nabla\phi|}\right) + vg\delta(\phi) \tag{4.28}$$

This gradient flow is the evaluation equation of the level set method without re-initialization proposed in [5] and has been applied in this work for lip contour detection.

The second and third term in the right hand side of (4.28) correspond to the gradient flows of the energy functional $\lambda\mathcal{L}_g(\phi)$ and $v\mathcal{A}_g(\phi)$, respectively and are responsible for driving the zero level curve towards the object boundaries. To explain the effect of the first term, which is associated to the internal energy $\mu\mathcal{P}(\phi)$, it can be noticed that the gradient flow:

$$\Delta\phi - div\left(\frac{\nabla\phi}{|\nabla\phi|}\right) = div\left[\left(1 - \frac{1}{|\nabla\phi|}\right)\nabla\phi\right] \tag{4.29}$$

has the factor $\left(1 - \frac{1}{|\nabla\phi|}\right)$ as the diffusion rate. If $\nabla\phi > 1$, the diffusion rate is positive and the effect of this term is the usual diffusion, i.e. making ϕ more even and therefore reduce the gradient $|\nabla\phi|$. If $|\nabla\phi| < 1$, the term has effect of reverse diffusion and therefore increase the gradient.

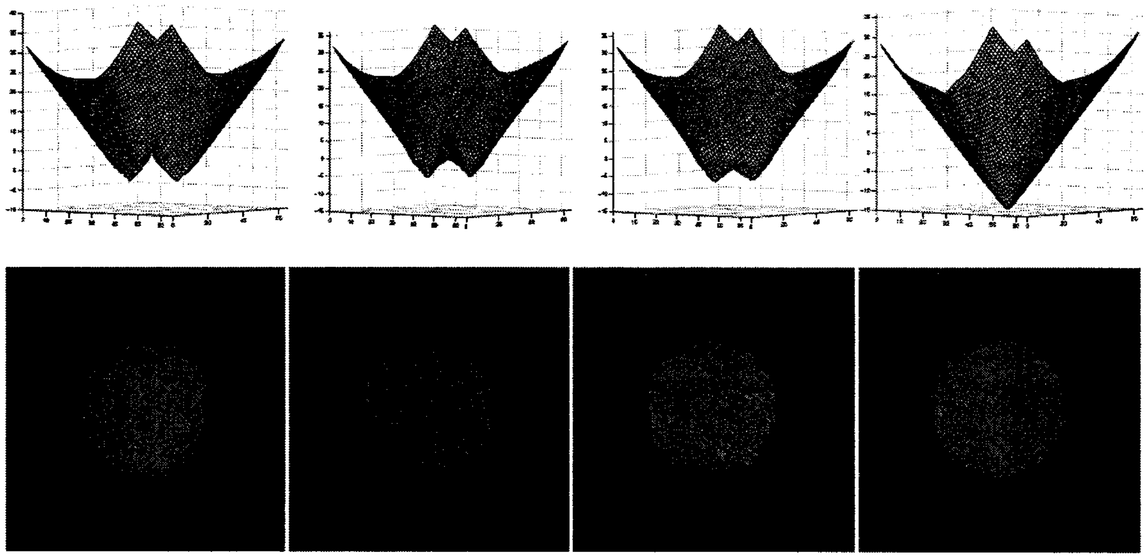


Figure 23: Evolution of level set function ϕ. Row 1: the evolution of the level set function ϕ. Row 2: the evolution of the zero level curve of the corresponding level set function ϕ in Row 1.

The evolution of ϕ, according to equation (4.29), has been demonstrated in Figure 23 using an image of a circular object. The upper row shows the evolution of the level set function ϕ, and the lower row shows the corresponding zero level curve of ϕ. The first figure of the upper row plots the initial level set function, and its zero level curve is plotted as the first figure of the lower row. The fourth column is the converged result of the evolution. As can be observed form Figure 23, the evolving level set function ϕ remains very close to that of a signed distance function during the evolution.

4.4.2. Implementation

As specified in [5], the Dirac function δ in (4.28) is slightly smoothed as the following function $\delta_\varepsilon(x)$ defined by:

$$\delta_{\varepsilon}(x) = \begin{cases} 0, & |x| > \varepsilon \\ \frac{1}{2\varepsilon}\left[1 + cos\left(\frac{\pi x}{\varepsilon}\right)\right], & |x| \leq \varepsilon \end{cases} \tag{4.30}$$

This regularized Dirac $\delta_{\varepsilon}(x)$ with $\mathcal{E} = 1.5$ has been used in this work. All the spatial derivatives $\frac{\partial \phi}{\partial x}$ and $\frac{\partial \phi}{\partial y}$ are approximated by the central difference, and the temporal partial derivative $\frac{\partial \phi}{\partial t}$ is approximated by the forward difference. The approximation of (4.28) by this difference scheme can be written as:

$$\frac{\phi_{i,j}^{k+1} - \phi_{i,j}^{k}}{\tau} = L(\phi_{i,j}^{k}) \tag{4.31}$$

where $L(\phi_{i,j})$ is the approximation of the right hand side in (4.28) by the above spatial difference scheme. The difference equation (4.31) can be expressed as follows in terms of iteration:

$$\phi_{i,j}^{k+1} = \phi_{i,j}^{k} + \tau L(\phi_{i,j}^{k}) \tag{4.32}$$

Equation (4.32) has been used to calculate the level set evaluation at any step rather than the initial one. The initial function ϕ_0 has been computed from the region enclosed by the quadrilateral enclosing as:

$$\phi_0(x,y) = \begin{cases} -\rho, & (x,y) \in \Omega_0 - \partial\Omega_0 \\ 0, & (x,y) \in \partial\Omega_0 \\ \rho, & \Omega - \Omega_0 \end{cases} \tag{4.33}$$

Here Ω_0 is a subset in the image domain Ω, $\partial\Omega_0$ be all points on the boundaries of Ω_0 which can be identified by some morphological operations [102], and ρ is a constant larger than 2ε, where ε is the width in the definition of the regularized Dirac function δ_ε in (4.30). This benefit of such region-based initialization of the level set function is that, it is not only computationally efficient, but also allows for flexible applications in some situations. For example, if the regions of interest can be roughly and automatically obtained in some way, such as thresholding, these roughly obtained regions can be used as the region Ω_0 to construct the initial level set function ϕ_0. Then, the initial level set function will evolve stably according to the evolution equation, with its zero level curve converged to the exact boundary of the region of interest.

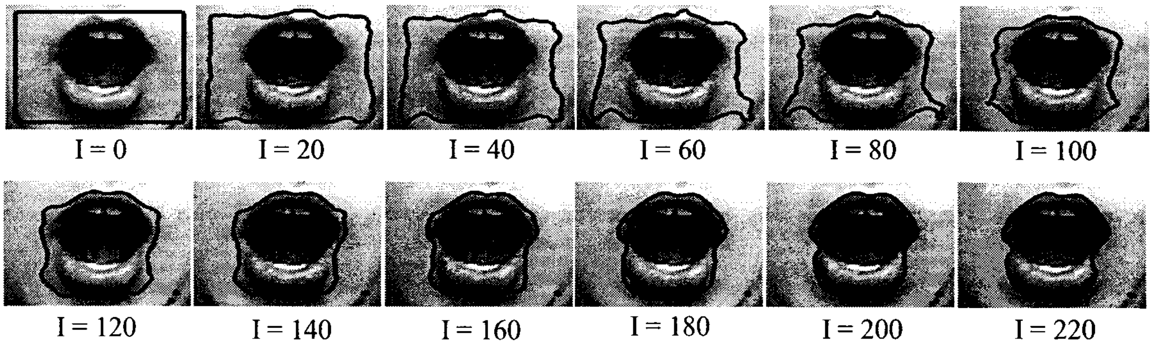


Figure 24: Evolution of the level set function for lip contour detection. Here, variable I indicates the number of iteration required for the curve evaluation.

In this work, we have used $\rho = 6$ in all the experiments. Besides, Ω_0 was set just close to the inside boundary of the isolated mouth region (Figure 24) without applying any morphological operation. According to (4.32), the time step τ has a significant impact on speeding up of the evolution as well as accuracy of the detected contour. A larger time step can speed up the evolution procedure by reducing the number of iteration required to set up the final contour, but it increases the possibility of error in the detected contour. The other important as well as adjustable parameters are λ, μ and ν. All these parameters have been adjusted using "trial and error" method to optimize the lip contour detection process in terms of the accuracy of the detected contour. An example of evolution of the level set curve during the lip contour detection process is given in Figure 24.

For detecting the lip contour from the isolated mouth region, the level set functions have been initialized as the function ϕ_0 defined by (4.33) with $\rho = 6$ and some regions Ω_0. For the subsequent phases, equation (4.32) with the parameters $\lambda = 5.0$, $\mu = 0.02$, $\nu = 1.25$ and time step $\tau = 10$ has been used. A small subset of the experimental results, obtained using this method, is provided in Figure 25.

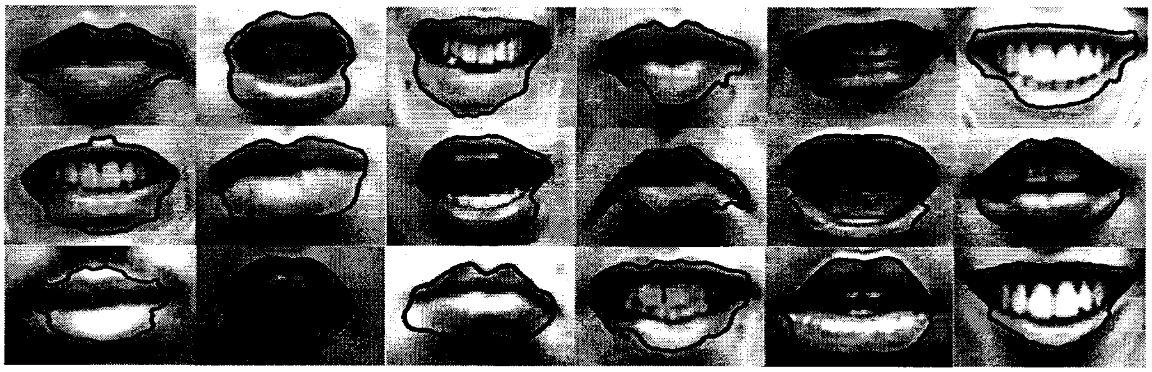


Figure 25: A small subset of the detected lip contours obtained using the level set method based lip contour detection technique.

Once the lip contour is detected from the isolated mouth region, feature points are identified by analyzing the coordinate values of the points lying over the contour. The right hand corner of the mouth is then identified as a point over the mouth contour having the minimum x coordinate value, and the point which has the maximum x coordinate value is considered as the left mouth corner. The middle point (x_{mid}, y_{mid}) of the left and right mouth corner are then calculated and upper and lower midpoints of mouth are obtained by finding the two specific points over the mouth contour which has the same x coordinate value as that of (x_{mid}, y_{mid}) but the minimum and maximum value of the y coordinate respectively.

Chapter 5

Classification of Facial Expressions Using SVM

The goal of any classification task is to label objects with respective classes based on their distinctive characteristics, which are also known as the features. In this work of facial expressions classification, the purpose of the dedicated classifier is to assign the facial expression present in each of the facial images to one of the categories namely, neutral, anger, disgust, fear, happiness, sadness and surprise. Therefore, features of the seven facial expressions are obtained first applying the analytic face model based technique of facial feature extraction discussed in Section 2.4. The obtained features are then used to train the Support Vector Machines classifier so that it can classify the facial expression of an unseen example when given to it in the form of a feature vector. Since there are total seven classes to deal with regarding the problem of facial expressions classification, multi-class Support Vector Machines constructed from the basic two-class SVM classifier has been adopted. However, instead of using the standard method for constructing the N-class SVM from N different two-class SVMs [121], the newly introduced Decision Directed Acyclic Graph (DDAG) based multi-class SVM approach, proposed by Platt et al. [59], has been used. For dealing with the N-class problem, the DDAG approach of multi-class SVM contains $\frac{N\times(N-1)}{2}$ binary SVM classifiers, one for each pair of classes and thus substantially reduce the required training as well as evaluation time.

5.1. Obtaining the Training Data

For obtaining the training data, fifteen facial feature points are detected following the procedure discussed in Chapter 4. The analytic face model (Section 2.4) for extracting the feature extractions of facial expressions is then formed using these fifteen points. Fifteen distances (D_1, $D_2, \cdots, D_{15}$), used in calculating the features of facial expressions (Table 2), are then measured from this analytic face model. Thereafter, the average representation of the neutral face ($D_{neutral}$) is calculated applying equation (2.1) over a subset of the training samples that belongs to the class "neutral". A feature vector from each of the training samples is then calculated following equation (2.2). The detailed procedure for generating the average neutral face representation as well as for obtaining the feature vectors from the training samples can be found in Section 2.4.

5.2. Classification of Facial Expressions Using SVM

Support Vector Machines have been extensively used as a classification tool achieving a great deal of success in many applications due to their attractive features and promising performance. They devise a computationally efficient way of learning a good separating hyperplane in a high dimensional feature space. In this research work, multi-class SVM has been applied towards the classification of facial expressions due to their outstanding ability to generalize. Unlike the conventional neural networks in which the network weights are determined by minimizing the mean squared error between the actual and the target outputs, denoted as the empirical risk minimization, SVM optimizes their parameters by minimizing the classification error, which is known as the structural risk minimization [122]. The empirical risk minimization strategy depends on the availability of a large amount of training data. In other words, it can be said that the empirical risks will be close to the true risk when the sample size is large and even a small empirical risk does not guarantee a low level of true risk for the problem of limited amount of training data. Under these circumstances, structural risk minimization strategy, which maintains a balance between empirical risk and complexity of the mapping function, is required. Therefore, due to its special property known as the structural risk minimization, SVM provides comparatively better solution in the cases where the training data is limited or difficult to obtain. This classification scheme also allows the use of nonlinear boundaries without extra computational cost.

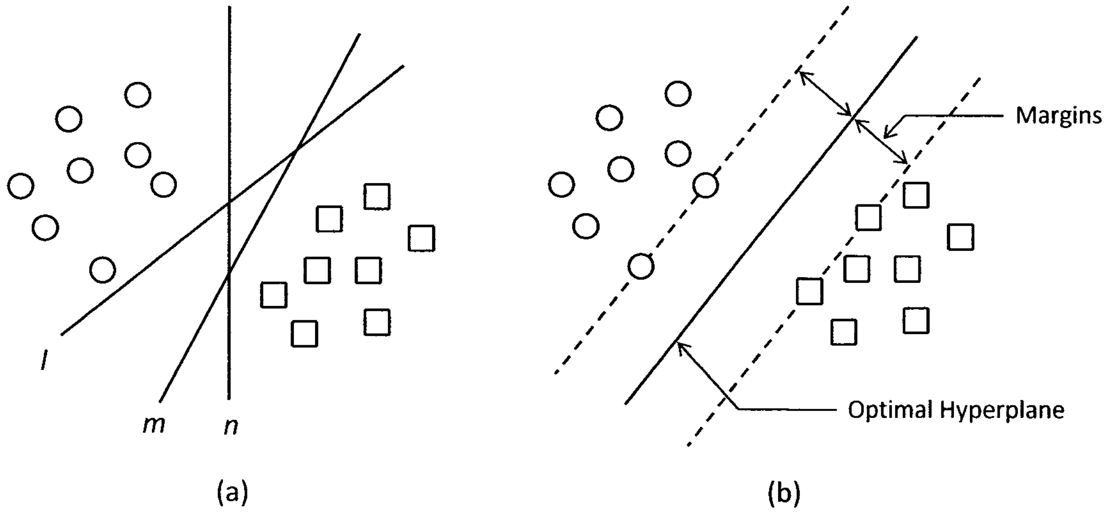


Figure 26: Classification between two classes using hyperplanes (a) arbitrary hyperplanes *l, m* and *n* (b) The optimal separating hyperplane with the largest margin identified by the dashed lines, which go through the support vectors.

To facilitate the discussion, a brief overview of SVM has been provided in this section. The basics of SVM for the two-class classification problem are elaborated first, followed by the discussion on how this technique can be extended to deal with the multi-class classification problem of facial expression recognition.

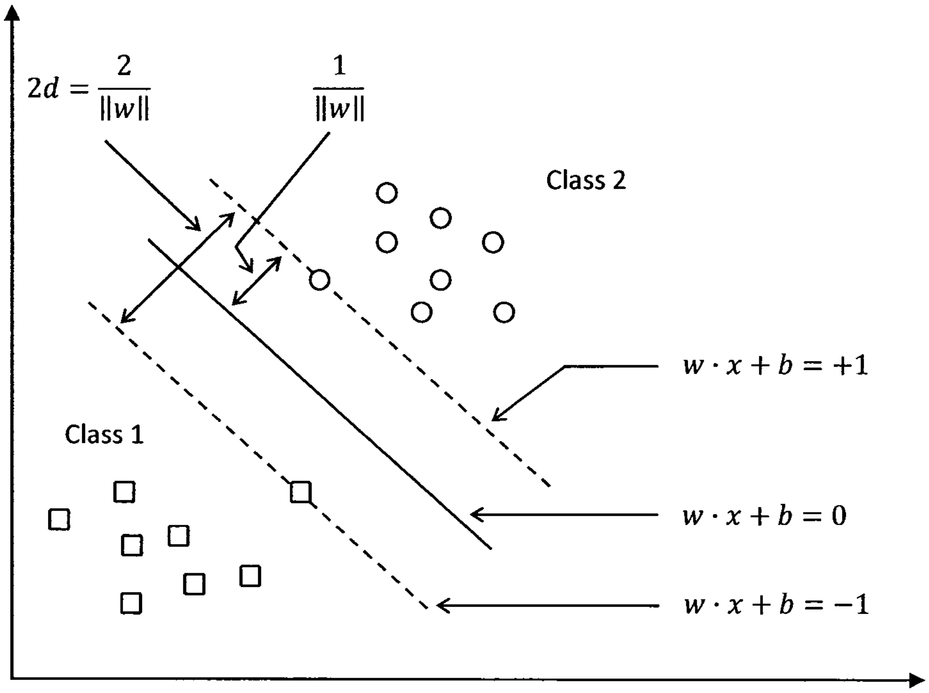


Figure 27: Demonstration of a two class classification process using Support Vector Machines (SVM).

5.2.1. Two-Class (Binary) Classification Using SVM

SVMs belong to the class of maximum margin classifiers, which perform classification between two classes by finding a decision surface that has a maximum distance to the closest points in the training set (Figure 26), termed as the support vectors [123]. Unlike other classifiers, SVMs control their generalization ability by minimizing their error rate on the training set and their capacity [124]. Let a training set of points be $x_i \in \mathbb{R}^n$, $i = 1, 2, \cdots, N$ where each point of x_i belongs to one of the two classes identified by the label $y_i \in \{+1, -1\}$. Assuming linearly separable data, the goal of maximum margin classification is to separate the two classes by a hyperplane such that the distance to the support vectors is maximized (Figure 26). Such a decision hyperplane, presented in Figure 27, can be expressed as:

$$x \cdot w + b = 0 \tag{5.1}$$

If all the training data satisfies the constraint of (5.1), then

$$x_i \cdot w + b \geq +1 \qquad for\ y_i = +1;\ i = 1, 2, \cdots, N \tag{5.2}$$

$$x_i \cdot w + b \leq -1 \qquad for\ y_i = -1;\ i = 1, 2, \cdots, N \tag{5.3}$$

If the equality holds for a data point x, then it is said to be right on the marginal hyperplane. Mathematically, the marginal hyperplanes are denoted as:

$$x \cdot w + b = \pm 1 \tag{5.4}$$

The two data points x_1 and x_2 that satisfy

$$x_1 \cdot w + b = +1 \tag{5.5}$$

$$x_2 \cdot w + b = -1 \tag{5.6}$$

will respectively fall on the two hyperplanes that are parallel to the decision plane and orthogonal to w shown in Figure 27. Subtracting (5.6) from (5.5), the following can be obtained:

$$w \cdot (x_1 - x_2) = 2$$

$$\Rightarrow \left(\frac{w}{\|w\|}\right) \cdot (x_1 - x_2) = \frac{2}{\|w\|} \tag{5.7}$$

Therefore, the distance between the two hyperplanes can be expressed as:

$$2d = \frac{2}{\|w\|} \tag{5.8}$$

where $2d$ is the width of separation that denotes how separable the two classes of training data are. The distance d is considered as the safety margin of the classifier. Now, by combining (5.2) and (5.3) into a single constraint, the following is obtained:

$$y_i(x_i \cdot w + b) \geq 1 \qquad \forall_i = 1, 2, \cdots, N \tag{5.9}$$

In the training phase, the main goal is to find the SV that maximize the margin of separation d. Alternatively, the similar objective can be achieved by minimizing $\|w\|^2$. Thus, the goal is to minimize $\|w\|^2$ subject to the constraint in (5.9). This can be solved by introducing the Lagrange multipliers $\alpha_i \geq 0$ and a Lagrangian

$$L(w, b, \alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{N} \alpha_i(y_i(x_i \cdot w + b) - 1) \tag{5.10}$$

where $L(w, b, \alpha)$ is minimized simultaneously with respect to w and b, and maximized with respect to α_i. Now setting of

$$\frac{\partial}{\partial b} L(w, b, \alpha) = 0 \tag{5.11}$$

$$\frac{\partial}{\partial w} L(w, b, \alpha) = 0 \tag{5.12}$$

subject to the constraint $\alpha_i \geq 0$, results in the followings:

$$\sum_{i=1}^{N} \alpha_i y_i = 0 \tag{5.13}$$

$$w = \sum_{i=1}^{N} \alpha_i y_i x_i \tag{5.14}$$

Substitution of (5.13) and (5.14) into (5.10) produces

$$L(w, b, \alpha) = \frac{1}{2}(w \cdot w) - \sum_{i=1}^{N} \alpha_i y_i (x_i \cdot w) - \sum_{i=1}^{N} \alpha_i y_i b + \sum_{i}^{N} \alpha_i$$

$$= \sum_{i}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) \tag{5.15}$$

This results in the following Wolfe dual optimization problem

$$\max_{\alpha} \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) \tag{5.16}$$

subject to the following:

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

2. $\alpha_i \geq 0, \;\; i = 1, 2, \cdots, N$

The solution of the dual optimization problem contains two types of multipliers α_k:

1. One type with $\alpha_k = 0$ and
2. The other with $\alpha_k \neq 0$

The data points corresponding to the zero multipliers ($\alpha_k = 0$) are considered to be irrelevant with the classification problem. On the other hand, the data points for which the multipliers $\alpha_k \neq 0$, are considered to be more critical for the classification problem and are denoted as the Support Vectors. According to the Karush-Kuhn-Tucker conditions of optimization theory [125] the following equality must be satisfied at all of the saddle points:

$$\alpha_i\{y_i(x_i \cdot w + b) - 1\} = 0 \qquad for \;\; i = 1, 2, \cdots, N \tag{5.17}$$

Therefore, only those data points for which $y_i(x_i \cdot w + b) - 1 = 0$ can be the SVs since these points have a non-zero value of multipliers α_i. Thus, all of the SVs satisfy the following:

$$y_k(x_k \cdot w + b) - 1 = 0 \qquad \forall_k \in S \tag{5.18}$$

where S is the set consisting of the indexes to the SVs. The values of α_i and w can be found from (5.16) and (5.14) respectively, and threshold b is calculated form (5.18) as:

$$f(x) = w \cdot x + b = \sum_{i=1}^{N} y_i \alpha_i (x \cdot x_i) + b = 0 \tag{5.19}$$

Sometimes the training data points are not clearly separable. Therefore, the concept of fuzzy or soft decision region is introduced to cope with the situations of such non-separable data points. A fuzzy SVM allows more relaxed separation which in turn provides more robust decision. In the case of fuzzy SVM, the separation width is denoted as the fuzzy separation region and thus, the distance $2d = \frac{2}{\|w\|}$ denotes the width of the fuzzy separation. If the data points are not separable by a linear separating hyperplane, a set of slack or relax variables $\{\xi = \xi_1, \xi_2, \cdots, \xi_N\}$ is introduced with $\xi_i \geq 0$ such that (5.9) becomes

$$y_i = (x_i \cdot w + b) \geq 1 - \xi_i \qquad \forall_i = 1, 2, \cdots, N \tag{5.20}$$

The slack variables allow some data points to violate the constraints noted in (5.9), which denotes the minimum safety margin required for the clearly separable training data. They also measure the deviation of the data points from the marginal hyperplane. In other words, the slack variables measure how severely the safety margin is violated. Therefore, the new objective function to be minimized becomes

$$\frac{1}{2}\|w\|^2 + C\sum_i \xi_i \tag{5.21}$$

subject to $y_i = (x_i \cdot w + b) \geq 1 - \xi_i$. Here C is the user defined penalty parameter to penalize any violation of the safety margin for all the training data. A smaller value of C leads to more SV while a larger value of C leads for fewer SV. It should also be noted that a smaller C leads to a larger width of the fuzzy separation while a larger C creates a narrower fuzzy separation region. Now the new Lagrangian becomes

$$L(w, b, \alpha) = \frac{1}{2}\|w\|^2 + C\sum_i \xi_i - \sum_{i=1}^{N} \alpha_i(y_i(x_i \cdot w + b) - 1 + \xi_i) - \sum_{i=1}^{N} \beta_i \xi_i \tag{5.22}$$

where $\alpha_i \geq 0$ and $\beta_i \geq 0$ respectively are the Lagrangian multipliers to satisfy that $y_i = (x_i \cdot w + b) \geq 1 - \xi_i$ and $\xi_i \geq 0$. Therefore, the Wholfe dual is calculated as:

$$\max_{\alpha} \sum_{i=1}^{N} \alpha_i - \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) \tag{5.23}$$

subject to $0 \leq \alpha_i \leq C$ for $i = 1, 2, \cdots, N$; and $\sum_{i=1}^{N} \alpha_i y_i = 0$. The output weight w is calculated as $w = \sum_{i=1}^{N} \alpha_i y_i x_i$ and the bias or the threshold term is measured by $(b = 1 - x_k \cdot w)$, where $y_k = 1$ and x_k is a support vector that lies on the plane $w^T x + b = 1$.

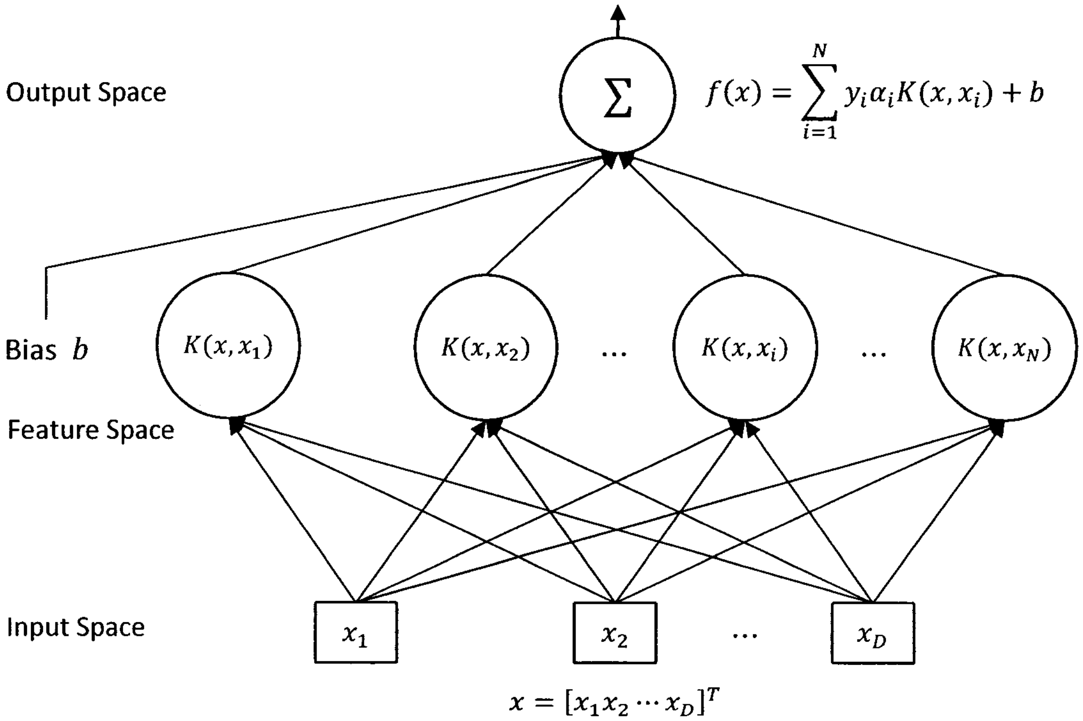


Figure 28: Two layer architecture of the Support Vector Machines (SVM).

A nonlinear hidden-layer is inserted between the input and output layers so that the two-layer network can provide an adequate amount of flexibility in the classification of the fuzzily separable data (Figure 28). The original linearly non-separable data points are mapped to a new feature space, denoted by the hidden nodes such that the mapped patterns become linearly separable. In order to obtain a nonlinear decision boundary, the inner product $(x \cdot x_i)$ of (5.19) is replaced with a nonlinear kernel $K(x, x_i)$, which results in the following:

$$f(x) = \sum_{i=1}^{N} y_i \alpha_i K(x, x_i) + b \tag{5.24}$$

The decision function in (5.24) can be implemented by a two-layer architecture shown in Figure 32 where it is depicted that the original input feature space is converted to a new feature space, manifested by the hidden-layer in the middle of the network.

Table 6: Kernels that are typically used with the SVM. Here, c, p, σ and b are the parameters used to define each particular kernel from the family given.

Kernel Function	**Formula**
Linear Kernel	$K(x, x_i) = x \cdot x_i$
Polynomial Kernel	$K(x, x_i) = \left(1 + \frac{x \cdot x_i}{\sigma^2}\right)^p, p > 0$
Radial Basis Function (RBF) Kernel	$K(x, x_i) = e^{-\left\{\frac{\lVert x - x_i \rVert^2}{2\sigma^2}\right\}}$
Sigmoid Kernel:	$K(x, x_i) = \frac{1}{1 + e^{-\frac{x \cdot x_i + b}{\sigma^2}}}$

The basic idea behind the nonlinear SVM is to use a kernel function $K(x, x_i)$ to map the data x from the input space to the new higher dimensional feature space on which the mapped data points become linearly separable. The hidden units on the middle layer are represented by the kernel function adopted by the SVM. The four typical kernels that have also been used in this work are provided in Table 6.

5.2.2. Multi-Class Classification Using SVM

As specified earlier, the problem of classifying facial expressions is a multi-class classification problem where the number of classes to deal with is seven. A number of methods that have already been proposed for multi-class classification using SVM, can be grouped into the following two categories:

i. 1-vs-rest Approach
ii. Pair-wise Approach

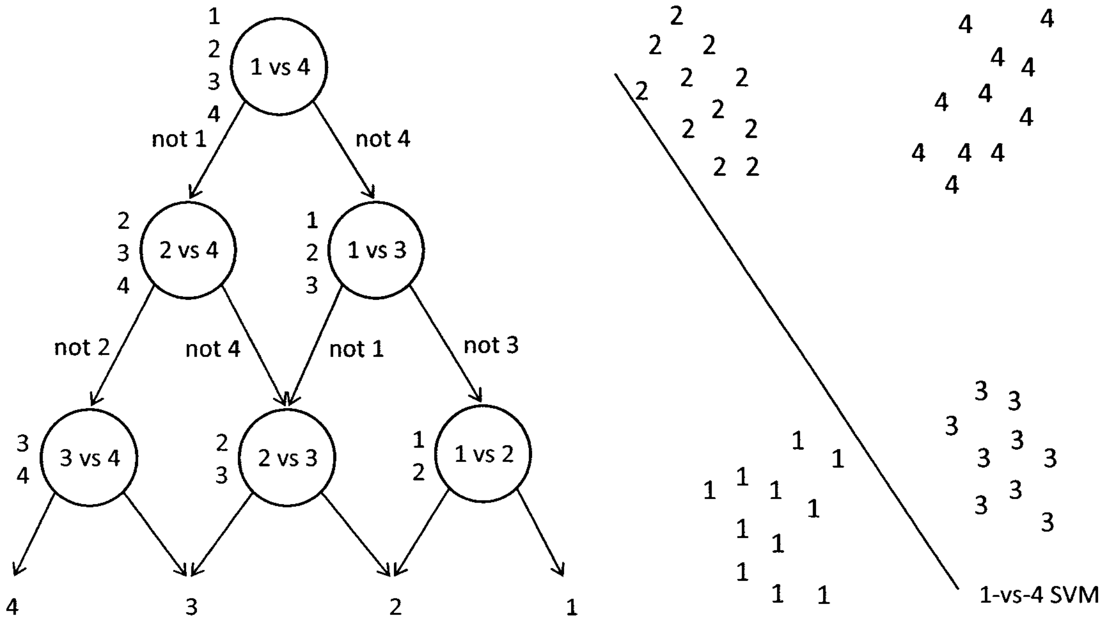
5.2.2.1. 1-vs-rest Approach

In the 1-vs-rest approach of multi-class classification, N different SVMs are trained to solve a N-class problem where each of the SVMs separates a single class from the remaining classes [121], [124], [126]. The i-th SVM is trained using all of the examples in the i-th class with positive labels, and all other examples with negative labels. The SVM trained in this way is also referred to as $1 - v - r$ SVM (short for one-versus-rest). The final output of the $1 - v - r$ SVM is the class that corresponds to the SVM with the highest output value. Unfortunately, there is no bound on the generalization error for the $1 - v - r$ SVM, and the training time of this method scales linearly with the value of N.

5.2.2.2. Pair-wise Approach

In the pair-wise approach of constructing N-class classifiers using SVM, $\frac{N\times(N-1)}{2}$ machines are trained where each SVM separates a pair of classes. Knerr et al. [127] proposed such a method for constructing N-class classifiers, which is actually derived from previous researches on combining binary classifiers. In this method, all possible two-class classifiers are constructed from a training set of N classes, each classifier being trained on only two out of these N classes. When applied to SVM, this method is referred to as the $1 - v - 1$ SVMs (short for one-versus-one). Knerr et al. [127] suggested combining these two-class classifiers with an "AND" gate. Besides, Friedman [128] proposed a Max Wins algorithm where each $1 - v - 1$ classifier casts one vote for its preferred class, and the final result is the class with the most votes.

A significant disadvantage of the $1 - v - 1$ approach is that, unless the individual classifiers are carefully regularized (as in SVM), the overall N-class classification system will tend to over fit [59]. Besides, none of the "AND" combination method and the Max Wins combination method has bounds on the generalization error. Finally, the size of the $1 - v - 1$ classifier may grow super linearly with N, and hence, may become slow in evaluating the large problems. The Decision Directed Acyclic Graph (DDAG) based multi-class SVM learning architecture, proposed by Platt et al. [59], shows significant improvement regarding these problems, and has been adopted in this research for the purpose of classifying facial expressions.



(a) (b)

Figure 29: Classification using multi-class SVM (a) the decision DAG for finding the best class out of four classes. The equivalent list state for each node is shown next to that node (b) a diagram of the input space of a four class problem. A $1 - \mathrm{v} - 1$ SVM can only exclude one class from consideration.

The DDAG approach consists of a set of binary SVM classifiers organized in a tree structure where each node is associated with a $1 - \mathrm{v} - 1$ classifier. Unseen data is evaluated at each node and depending on the result at each node; the data will traverse the tree until a solution is obtained. In other words this algorithm reduces a multi-class problem to a set of two-class classifiers at each node. Figure 29 illustrates the classification technique of DDAG based multi-class SVM.

Given a space X and a set of boolean functions $\mathcal{F} = \{f: X \longrightarrow \{0,1\}\}$, the class DDAG($\mathcal{F}$) of Decision DAGs on N classes over $\mathcal{F}$ are functions which can be implemented using a rooted binary DAG with N leaves labeled by the classes where each of the $K = \frac{N\times(N-1)}{2}$ internal nodes is labeled with an element of $\mathcal{F}$. The nodes are arranged in a triangle with the single root node at the top, two nodes in the second layer and so on until the final layer of N leaves. The i-th node in layer $j < N$ is connected to the i-th and $(i+1)$-th node in the $(j+1)$-th layer.

To evaluate a particular DDAG on input $x \in X$ starting at the root node, the binary function at a node is evaluated. The node is then exited via the left edge, if the binary function is zero; or the right edge, if the binary function is one. The next node's binary function is then evaluated. The value of the decision function $D(x)$ is the value associated with the final leaf node (Figure 29.a). The path taken through the DDAG is known as the *evaluation path*. The input x eventually reaches a node of the graph, if that node is on the evaluation path for x. The decision node distinguishing classes i and j is referred to as the ij-node. Assuming that the number of a leaf is its class, this node is the i-th node in the $(N-j+i)$-th layer provided $i < j$. Similarly the j-th nodes are those nodes involving class j; that is, the internal nodes on the two diagonals containing the leaf labeled by j.

The DDAG approach is equivalent to operating on a list, where each node eliminates one class from the list. The list is initialized with a list of all classes. A test point is evaluated against the decision node that corresponds to the first and last elements of the list. If the node prefers one of the two classes, the other class is eliminated from the list, and the DDAG proceeds to test the first and last elements of the new list. The process terminates when only one class remains in the list. Thus, for a problem with N classes, $(N-1)$ decision nodes will be evaluated in order to derive an answer. The current state of the list is the total state of the system. Therefore, since a list state is reachable in more than one possible path through the system, the decision graph that the algorithm traverses is a DAG, not simply a tree.

The learning algorithm of the Decision Directed Acyclic Graph based multi-class SVM is referred to as the DAGSVM. It creates a DDAG whose nodes are maximum margin classifiers over a kernel-induced feature space. Such a DDAG is obtained by training each ij-node only on the subset of training points labeled by i or j. The final class decision is derived using the DDAG architecture, described earlier.

For DAGSVM, the choice of the class order in the list (or DDAG) is arbitrary. This algorithm is also superior to other multiclass SVM algorithms regarding both training and evaluation time. Empirically, SVM training is observed to scale super-linearly with the training set size m [129], according to a power law: $T = cm^{\gamma}$, where $\gamma \approx 2$ for algorithms based on the decomposition method, with some proportionality constant c. For the standard $1 - \mathrm{v} - \mathrm{r}$ multiclass SVM training algorithm, the entire training set is used to create all N classifiers. Hence the training time of a single $1 - \mathrm{v} - \mathrm{r}$ is given as [59]:

$$T_{1-v-r} = cNm^{\gamma} \tag{5.25}$$

Assuming that the classes have the same number of examples, training each $1-\mathrm{v}-1$ SVM requires only $\frac{2m}{N}$ training examples. Thus, time required to train the K instances of $1-\mathrm{v}-1$ SVM towards forming an N-class DDAG based SVM can be expressed as:

$$T_{1-v-1} = c\frac{N(N-1)}{2}\left(\frac{2m}{N}\right)^{\gamma} \approx 2^{\gamma-1}cN^{2-\gamma}m^{\gamma} \tag{5.26}$$

For a typical case, where $\gamma = 2$, the amount of time required to train all of the $1-\mathrm{v}-1$ SVMs is independent of N, and is only twice that of training a single $1-\mathrm{v}-\mathrm{r}$ SVM. Thus the DDAG approach of multiclass-classification is comparatively faster both in terms of training as well as classification time [59].

5.2.3. Training the Classifier for Classification

During the training phase, the adjustment of the penalty parameter for error term C and the kernel parameters are important to improve the generalization performance of the SVM classifier. When the number of classes in the training set is very large, careful selection of a training subset and a validation set with small number of classes is required to avoid the training of SVM with all classes and evaluating its performance on the validation set due to high computational cost [60]. However, as the total number of classes to be dealt with in this problem of facial expression classification is only seven, the *k-fold cross validation* [130] procedure with training data from all the seven classes has been applied to select the optimum parameters for tuning the SVM classifier towards eliciting the best possible performance. The cross validation procedure provides an estimation of the general performance of the trained classifier. Based on this estimated general performance, the kernel parameters and the penalty parameter for error term C are tuned to achieve the level of confidence expected from the classifier in classifying facial expressions. Details of the k-fold cross validation procedure used in this thesis are provided below:

1. Let the set of data to be used for training the multi-class SVM classifier is V with cardinality $|V|$. This training set V is divided into mutually exclusive k-subsets $H_1, H_2, \cdots, H_k$ (here, $k \geq 1$) such that,

$$V = \bigcup_{i=1}^{k} H_i \tag{5.27}$$

Here, k and H_i are chosen in a way such that the class distribution in every H_i remains similar to the class distribution of V and the cardinalities of the sets $H_1, H_2, \cdots, H_k$ are equal or approximately equal to each other.

2. For $i = 1, 2, \cdots, k$; the SVM classifier is trained using the truncated training set $V - H_i$ and tested using H_i as if H_i were the recall set. Since the class of every pattern of H_i is known, the confusion matrix CM_i for the validation set H_i is calculated as:

$$CM_i = \begin{pmatrix} m_{11} & m_{12} & \cdots & m_{1n} \\ m_{21} & m_{22} & \cdots & m_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ m_{n1} & m_{n2} & \cdots & m_{nn} \end{pmatrix} \tag{5.28}$$

Here, $n = 7$ is the number of classes representing seven different facial expressions. The elements m_{pq} (for which $p \neq q$) of the confusion matrix CM_i represent the number of patterns that actually belongs to the class represented by p, but have been misclassified to the class represented by q. Then, the total number of facial expression patterns that have been misclassified from the validation set H_i is accumulated as:

$$err_i = \sum_{p=1}^{n} \sum_{q=1,q\neq p}^{n} m_{pq} \tag{5.29}$$

3. The total number of patterns that have been misclassified for the training set V is obtained by summing up the values of err_i ($i = 1, 2, \cdots, k$), i.e. the number of facial expression patterns that have been misclassified for each validation set H_i; $i = 1, 2, \cdots, k$. So, the error (misclassification) rate E over the whole training set is calculated as:

$$E = \frac{\sum_{i=1}^{k} err_i}{|V|} \tag{5.30}$$

For the work of facial expressions classification described in this thesis, the acceptable error rate during the training phase is set to 0.05. So, the cross validation procedure is repeated, each time modifying the value of the kernel parameters and the penalty parameter C until the value of the error term E reaches the maximum acceptable error rate of 0.05. Once such parameter values are obtained, the SVM classifier is trained finally using these parameter values applying the whole training set. This trained multi-class SVM classifier is used later on for classifying the facial expressions from the test images with a view of evaluating the performance of the proposed automated facial expression classification system.

Chapter 6

Experimental Results

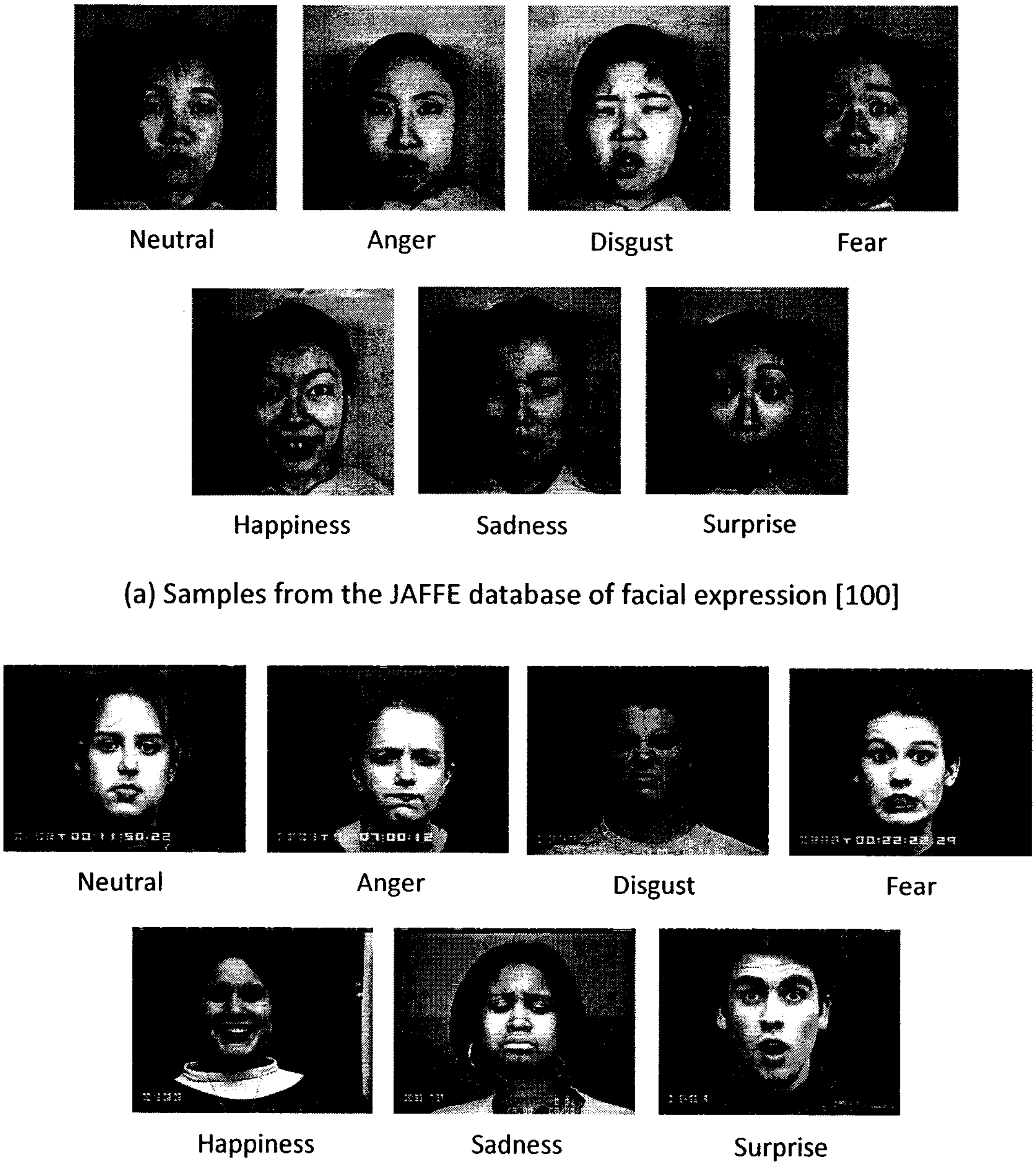
Capability of providing high classification accuracy over small training sets as well as generalization performance on data that is highly variable and difficult to separate make Support Vector Machines particularly suitable for the application of classifying facial expressions. Kernel choice is among the most important customizations that can be made while adjusting the SVM classifier to a particular application domain. By performing experiments with the trained DDAG based multi-class SVM using a range of Linear, Polynomial, Gaussian Radial Basis Function (RBF) and Sigmoid Kernels provided in Table 6, it has been observed that RBF kernels significantly outperform the others, boosting the overall recognition accuracy of the proposed automated facial expressions classification system. Besides, it has also been noted that use of different kernels has produced different decisions over a common test set in a partially overlapping manner while classifying facial expressions. This signifies that facial expressions of the test samples that are not correctly classified using a specific kernel, can be classified correctly using another kernel. As, use of multiple classifiers with a suitable combining strategy can improve the classification accuracy in such situation [131], the final classifier has been constructed from six different DDAG based multi-class SVM by combining them using the *Multiexpert Combination* method with the *Voting* scheme [60]. The prototype of the proposed automated facial expression system has been developed using Matlab® 7 with the support of "Machine Perception Toolbox" [132], "PRTools" [133] and the Matlab® code of level set method available from [134]. Functional overview of the prototyped system has been provided in Appendix A. Performance of the proposed multi-class SVM based automated facial expression classification method has been compared with three other well known classification techniques namely *k*-NN, Neural Network and Naive Bayes classifier. In addition, comparison of the proposed technique with most of the previously proposed significant methods has also been documented.

6.1. The Facial Expression Databases Used

The work of facial expression classification, discussed in this thesis, require frontal face images of seven expressions (neutral, anger, disgust, fear, happiness, sadness, and surprise) for the purpose of both training and testing. But it is really difficult for the subjects to generate these expressions artificially for the purpose of data collection, and 2 out of 20 subjects are expected to generate the true expressions in such situation [34], [67]. Besides, expert psychologists are required to mark each expression to its corresponding emotion category. So, instead of generating new datasets for this research, two of the existing databases have been used here that are mostly recognized for the research on facial expressions classification. These databases are:

i. The Japanese Female Facial Expression (JAFFE) Database

ii. Cohn-Kanade AU-Coded Facial Expression Database



(a) Samples from the JAFFE database of facial expression [100]

(b) Samples from the Cohn-Kanade AU-coded facial expression database [101]

Figure 30: Samples from the two facial expression databases used in training the testing the proposed automated system for facial expression classification.

The JAFFE database [100] contains 213 images, each representing seven different facial expressions (six basic facial expressions + one neutral) posed by ten Japanese female models. Each image of this database has been rated on seven emotion adjectives by 60

Japanese subjects. Since there are only 3-5 samples of each expression for each subject in this database, some new samples from the existing images have been generated by adding Gaussian random noise (with $\mu = 0$, $\sigma = 0.005$) and Salt-and-Pepper noise (with noise density $d = 0.04$) towards preparing fourteen samples per expression of each subject.

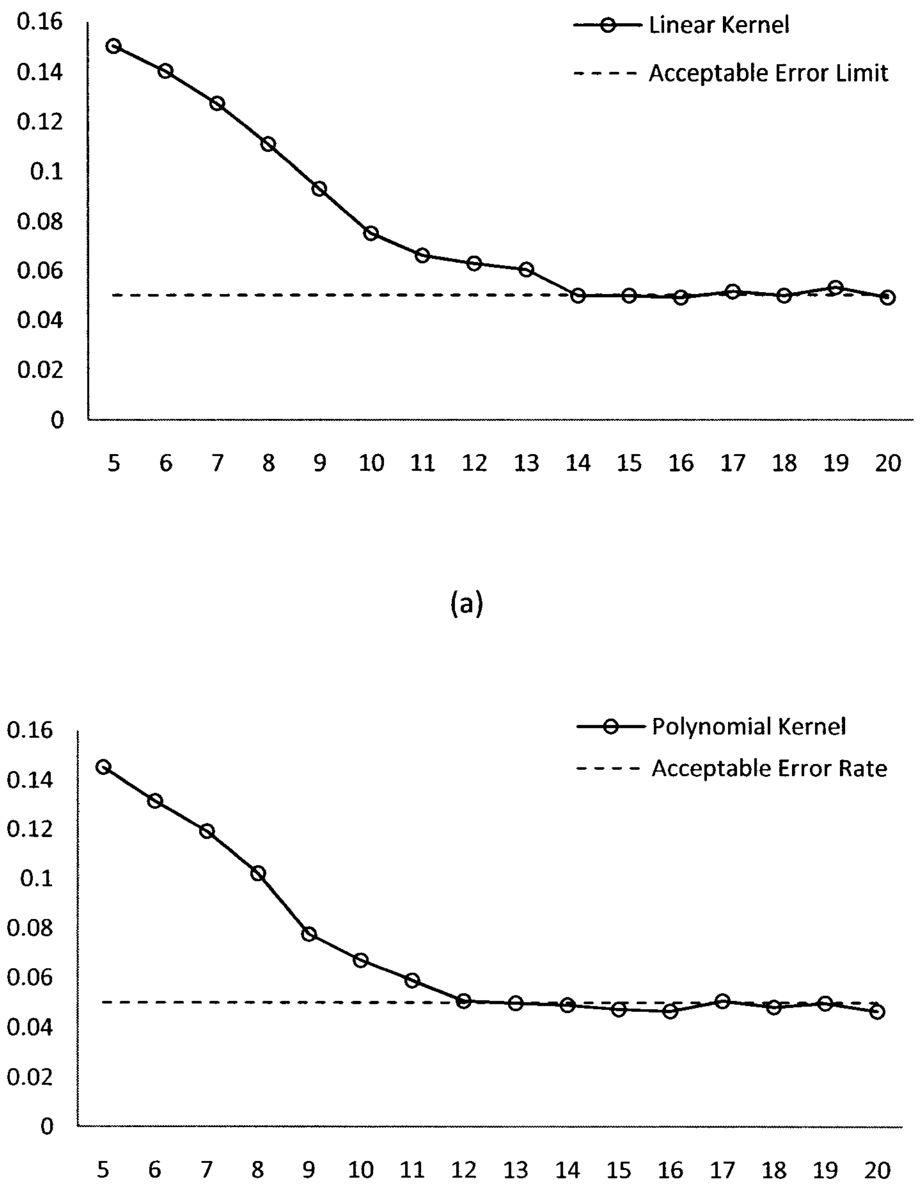
The Cohn-Kanade AU-Coded Facial Expression Database [101] includes 2105 digitized image sequences from 182 adult subjects of varying ethnicity performing multiple tokens of most primary FACS action units. All the subjects of this database are in between the ages of 18 and 50 years and there are 69% female, 31% male, 81% Euro-American, 13% Afro-American and 6% other groups. From them, total 2450 images (10 samples/expression of each subject) of selected 35 subjects have been used in the experiments for which the database contains samples of both the neutral face and the six basic expressions. Sample of facial expression images from both of the databases are provided in Figure 30.

6.2. Experiment with Different Kernels

At first, the performance of the proposed facial expressions classification system has been analyzed using each of the kernels provided in Table 6. For this purposed feature vectors extracted from the samples of each class are divided equally into two parts, one to be used for the purpose of training and another for testing. Thus, total $\frac{10\times7\times14}{2} = 490$ feature vectors extracted from the JAFFE database, and $\frac{35\times7\times10}{2} = 1225$ feature vectors extracted from the Cohn-Kanade database have been used for training the

DDAG based multi-class SVM classifier towards classifying facial expressions. Since the number of classes to be handled for the problem of facial expressions classification is seven, total $\frac{7\times(7-1)}{2} = 21$ two class $1 - v - 1$ SVMs has been trained first using the DAGSVM learning algorithm. Theses 21 trained binary SVM classifiers are then organized following the DDAG architecture, discussed in sub-section 5.2.2.2, towards forming the multi-class SVM classifier for facial expression classification.

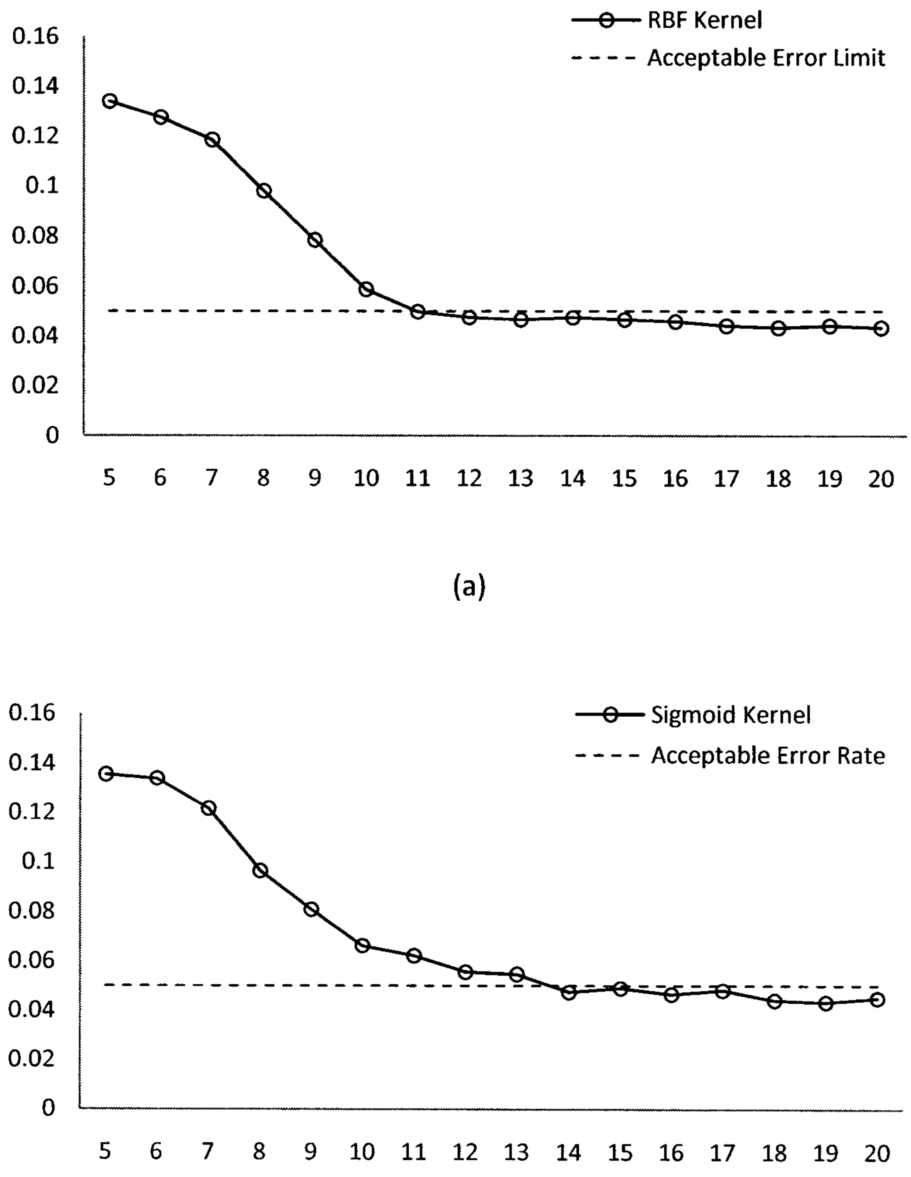
To improve the generalization performance of the multi-class SVM classifier, the kernel parameters and the penalty parameter C have been selected carefully following the k*-fold cross validation* procedure discussed in sub-section 5.2.3. The value of k that is the number of partition in the training data to be used for the purpose of cross validation is hugely dependent on the size of the training data as well as the recognition accuracy expected from the classifier. Increasing the number of partitions k can improve the recognition accuracy of the classifier over the training set up to a certain point, after which no significant improvement can be achieved increasing the value of k [130]. Besides, computational complexity of the training procedure increases linearly with the number of partitions used for the cross validation procedure. So, performing experiments using both the training datasets collected from the JAFFE and the Cohn-Kanade database with carefully following the guidelines of [130], $k = 14$ has been set in this work for the purpose of parameter selection using the cross validation procedure. The error rates achieved for different values of k using the linear, polynomial (with degree, $p = 4$), RBF and sigmoid kernels over the 1225 training sample of the Cohn-Kanade database have been demonstrated in Figure 31 and Figure 32.



(a)

(b)

Figure 31: Achieved error rate over the 1225 training sample of the Cohn-Kanade database for different values of k using (a) Linear kernel, and (b) Polynomial kernel. Here, x-axis represents the number of partitions (k) used to divide the training set for the purpose of cross validation, and y-axis represents the misclassification (error) rate calculated using equation (5.30).



(a)

(b)

Figure 32: Achieved error rate over the 1225 training sample of the Cohn-Kanade database for different values of k using (a) RBF kernel, and (b) Sigmoid kernel. Here, x-axis represents the number of partitions (k) used to divide the training set for the purpose of cross validation, and y-axis represents the misclassification (error) rate calculated using equation (5.30).

With this value of $k = 14$, the acceptable maximum misclassification rate of 0.05 has been achieved from the DDAG based multi-class SVM over the training datasets using all the four kernels listed in Table 6 (Figure 31 and Figure 32). The kernel parameter σ^2 is set at 0.8 and the penalty parameter C is set to 18.0 when the maximum acceptable misclassification rate of 0.05 has been achieved following the cross validation procedure discussed in sub-section 5.2.3. Once the suitable values of the parameters are obtained, the DDAG based multi-class SVM is trained separately using these parameter values and the four type of kernels (Table 6) applying the whole set of training data. Performance of the classifier in classifying facial expression is then evaluated using the complete set of feature vector that includes both the training and test data.

Table 7: Performance of the proposed automated facial expressions classification system in classifying facial expressions over the Japanese Female Facial Expression (JAFFE) database using different kernels listed in Table 6.

Expression	Recognition Accuracy (%) over the JAFFE Database			
	Linear Kernel	**Polynomial Kernel**	**RBF Kernel**	**Sigmoid Kernel**
Neutral	87.14	90.71	90.00	86.43
Anger	78.57	81.43	85.71	85.00
Disgust	81.43	87.86	90.71	88.57
Fear	77.86	84.29	81.43	79.29
Happiness	88.57	89.29	94.29	92.14
Sadness	77.14	82.86	82.14	83.57
Surprise	91.43	96.43	97.14	92.86
Average	83.16	87.55	88.77	86.84

The experiment for analyzing the performance of the proposed facial expression classification system using various kernels was conducted over the JAFFE database and the Cohn-Kanade database separately and the obtained results are provided in Table 7 and Table 8. As can be observed from the experimental results, classification accuracy obtained using the Polynomial, RBF and Sigmoid kernels are very closer as well as comparatively better than the recognition accuracy achieved using the Linear kernel. However, the best classification accuracy has been achieved using the RBF kernel over both of the databases, which is 88.77% for the JAFFE database (Table 7) and 83.63% for the Cohn-Kanade database of facial expression (Table 8).

Table 8: Performance of the proposed automated facial expressions classification system in classifying facial expressions over the Cohn-Kanade AU-Coded Facial Expression database using different kernels listed in Table 6.

Expression	Recognition Accuracy (%) over the Cohn-Kanade Database			
	Linear Kernel	**Polynomial Kernel**	**RBF Kernel**	**Sigmoid Kernel**
Neutral	75.43	79.71	78.86	76.29
Anger	74.57	78.57	81.43	78.00
Disgust	77.14	83.43	85.14	82.57
Fear	76.57	82.00	81.14	81.71
Happiness	81.43	86.29	87.14	84.86
Sadness	76.86	78.29	77.71	80.29
Surprise	87.43	93.14	94.00	91.14
Average	78.49	83.06	83.63	82.12

6.3. Combining Classifier to Achieve Better Classification Accuracy

While analyzing the performance of different kernels in classifying facial expressions (Section 6.2), it has been observed that use of different kernels have produced different decisions over the same test set in a partially overlapping manner. Besides, it has also been noted that facial expressions of the test samples that are not correctly classified using a specific kernel, has been classified correctly using another kernel. This leads to the decision of combining several SVM classifiers in this work of facial expression classification, all of which are constructed separately embedding different kernels.

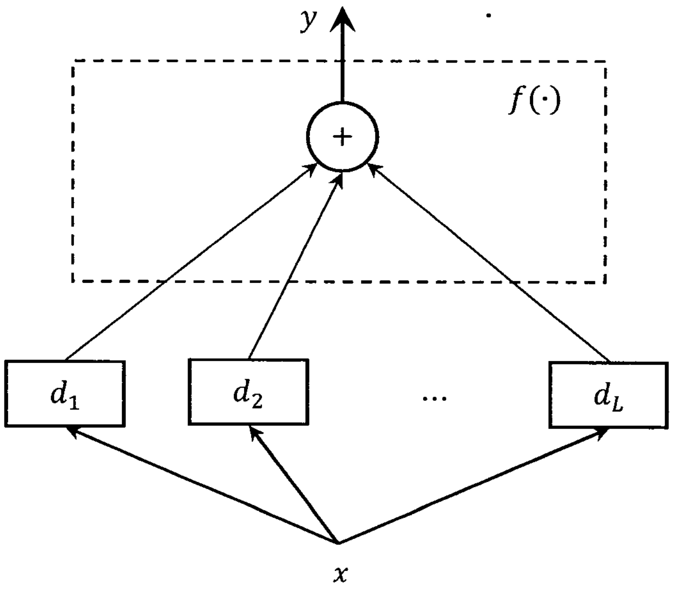


Figure 33: *Multiexpert Combination* method for combining classifies using the *voting* scheme. The combiner function $f(\cdot)$ is a weighted sum, d_j are the multiple learners, w_j are the weights of their votes, and y is the overall output.

Let us consider that there are L base-learners and $d_j(x)$ be the prediction of base-learner M_j given the arbitrary dimensional input x. In the case of multiple representations, each M_j uses a different input representation x_j and the final prediction is calculated from the predictions of the base-learners as:

$$y = f(d_1, d_{2,} \cdots, d_L | \Phi) \tag{6.1}$$

Here, $f(\cdot)$ is the combining function with Φ denoting its parameters. When there are K inputs, each learner has K outputs $d_{ji}(x)$ for $i = 1,2,\cdots,K; j = 1,2,\cdots,L$ and combining them, K values, y_i, $i = 1,2,\cdots,K$ can be generated. Final classification is performed by choosing the class with the maximum value of y_i.

The *Multiexpert Combination* method with the *Voting* scheme [60] has been used here for combing six different SVM classifiers. Three of these SVM classifiers have been constructed using the Linear, Sigmoid and RBF kernel. For generating the other three SVM classifiers, polynomial kernels with degree = 3, 4 and 5 have been used. The voting scheme of combining multiple classifiers corresponds to taking a linear combination of the learners. This is also known as *ensembles* and *linear opinion pools*. If w_j be the weight of learner j, the final output is computed as (Figure 33):

$$y = \sum_{j=1}^{L} w_j d_j \tag{6.2}$$

where, $w_j \geq 0, \forall_j$ and $\sum_{j=1}^{L} w_j = 1$. Comparing with equation (6.1), $f(\cdot)$ corresponds to the weighted sum in equation (6.2) where Φ is the set of weights, $w_j, w_2, \cdots, w_L$. For assigning the weights of the six SVM classifiers used for the purpose of classifier combining, the individual performance of each of the classifier, observed while using each of them separately with the proposed system of facial expression classification, has been considered. Due to its outperforming individual performance, SVM classifier with RBF kernel was assigned a weight of 0.3. Each of the other five classifiers was assigned a weight of 0.14 so that their outputs can have significantly less impact on the final decision comparative to that of the RBF kernel. Performance of the proposed method of facial expression classification using this combined multi-class SVM classifier has been documented in Table 9 and Table 11. As the experimental results indicates, recognition accuracy of the proposed system for facial expression classification has been increased to 92.04% for the JAFFE database and 86.33% for the Cohn-Kanade database due to the positive impact of the combined multi-class SVM classifier.

6.4. Comparison with Other Classifiers

Performance of the proposed combined multi-class SVM based automated facial expression classification system has been carried out with three other popular classification methods namely, *k*-Nearest Neighbor (*k*-NN), Neural Network (NN) and Naive Bayes classifier.

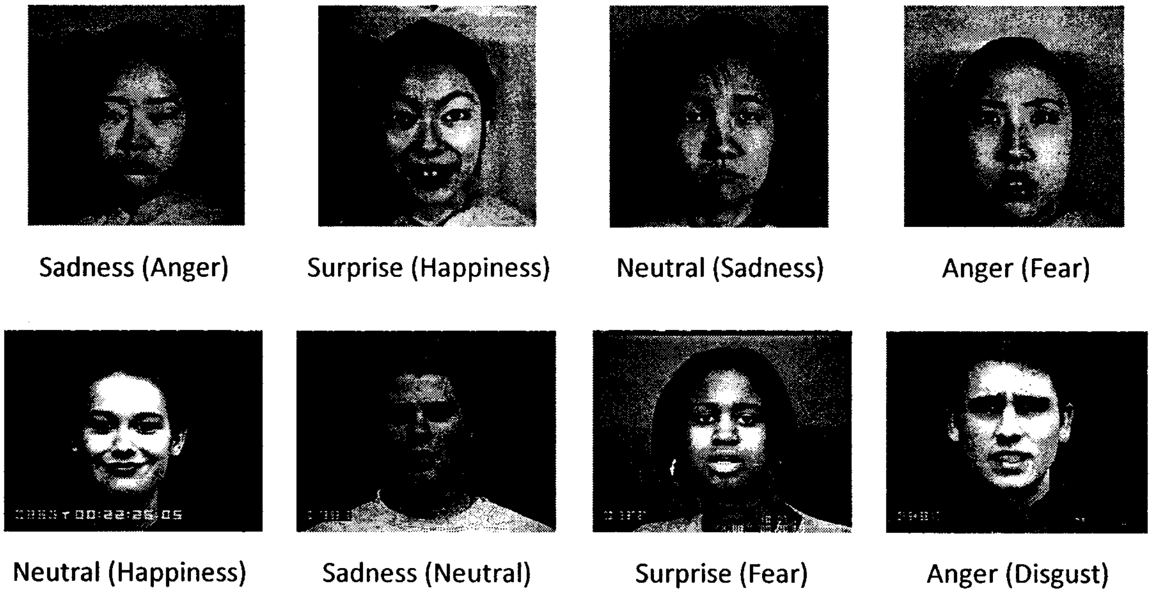
For the purpose of comparison, a suitable value of k has been chosen for the *k*-NN classifier using the cross validation procedure discussed in sub-section 5.2.3, and it has been found by experiments that $k = 3$ is the best choice for achieving the acceptable maximum misclassification rate of 0.05 over the training datasets collected from the JAFFE and Cohn-Kanade database. Besides, maximum recognition rate has been achieved with this value of *k* over the validation dataset using the *k*-NN classifier.

Table 9: Comparison of the proposed multi-class SVM based automated facial expressions classification system with other classifiers over the Japanese Female Facial Expression (JAFFE) database.

Expression	Recognition Accuracy (%) over the JAFFE Database			
	Multi-Class SVM (Combined)	***k*-NN (*k*=3)**	**Neural Network**	**Naive Bayes Classifier**
Neutral	91.43	82.86	82.14	76.43
Anger	88.57	80.71	77.86	73.57
Disgust	92.86	87.14	81.43	78.57
Fear	89.29	84.29	80.71	74.29
Happiness	96.43	90.00	84.29	80.00
Sadness	87.14	86.43	80.00	77.86
Surprise	98.57	92.86	88.57	83.57
Average	92.04	86.33	82.14	77.76

Table 10: Confusion matrix of the classification accuracy of the proposed automated facial expressions classification system over the JAFFE database (14 samples/expression of each subject).

Expression	**Neutral**	**Anger**	**Disgust**	**Fear**	**Happiness**	**Sadness**	**Surprise**	**Recognition Rate (%)**
Neutral	128	3	1	3	2	3	0	91.43
Anger	5	124	3	5	1	2	0	88.57
Disgust	3	5	130	1	0	0	1	92.86
Fear	6	4	2	125	1	2	0	89.29
Happiness	2	0	1	1	135	0	1	96.43
Sadness	8	3	1	6	0	122	0	87.14
Surprise	0	0	0	0	2	0	138	98.57


Sadness (Anger) Surprise (Happiness) Neutral (Sadness) Anger (Fear)

Neutral (Happiness) Sadness (Neutral) Surprise (Fear) Anger (Disgust)

Figure 34: Some of the facial expression images from the databases that have been classified incorrectly by the proposed automated facial expression classification system. The samples in the upper row are from the JAFFE database and the samples in the bottom row are from the Cohn-Kanade database of facial expressions. The label outside the parenthesis specifies the misclassified expression whereas the actual expression of an image is specified within the parenthesis.

The three layer neural network with Back-Propagation Learning algorithm [60] has been used in this work for the purpose of comparison. Using cross validation procedure, number of hidden nodes of the neural network was fixed to 220 and number of iteration was finalized to 500 towards achieving the acceptable maximum misclassification rate of 0.05, as defined by equation (5.30). Results of the comparative analysis of recognition accuracies using *k*-Nearest Neighbor (*k*-NN), Neural Network (NN) and Naive Bayes classifier with that of the combined DDAG based multi-class SVM classifier have been provided in Table 9 (for JAFFE database) and in Table 11 (for Cohn-Kanade database).

Besides, corresponding confusion matrices enlisting the number of misclassified and correctly classified facial expression patterns using the combined multi-class SVM classifier have been given in Table 10 and Table 12.

Table 11: Comparison of the proposed multi-class SVM based automated facial expressions classification system with other classifiers over the Cohn-Kanade AU-Coded Facial Expression database.

Expression	Recognition Accuracy (%) over the Cohn-Kanade Database			
	Multi-Class SVM (Combined)	***k*-NN (*k*=3)**	**Neural Network**	**Naive Bayes Classifier**
Neutral	80.29	80.00	74.57	68.86
Anger	83.43	77.14	74.29	70.00
Disgust	87.71	80.29	79.14	73.14
Fear	85.14	76.00	72.29	68.29
Happiness	90.29	83.71	80.57	75.71
Sadness	81.71	80.29	76.57	66.00
Surprise	95.71	89.14	87.14	81.43
Average	86.33	80.94	77.80	71.92

As can be observed from the experimental results of Table 9 and Table 11, the combined multi-class SVM classifier outperforms the other classifiers in terms of recognition accuracy while tested over the JAFFE database and the Cohn-Kanade database of facial expression. For the JAFFE database, the achieved average classification rate is 92.04% (Table 9) and for the Cohn-Kanade database, the obtained average correct recognition rate of the seven facial expressions is 86.33%.

Table 12: Confusion matrix of the classification accuracy of the proposed automated facial expressions classification system over the Cohn-Kanade AU coded database of facial expression (10 samples/expression of each subject).

Expression	Neutral	Anger	Disgust	Fear	Happiness	Sadness	Surprise	Recognition Rate (%)
Neutral	281	10	6	21	3	28	1	80.29
Anger	9	292	14	11	5	13	6	83.43
Disgust	9	11	307	8	4	6	5	87.71
Fear	17	8	11	298	2	14	0	85.14
Happiness	6	7	5	2	316	3	11	90.29
Sadness	21	10	9	17	4	286	3	81.71
Surprise	2	2	3	1	7	0	335	95.71

All the experiments were conducted using an Intel Pentium 4 processor based computer with 3.06 GHz of processing speed and 1GB physical memory. Average computational time required in recognizing the facial expression from a test image is provided in Table 13. Due to the variation of spatial resolution of the images, the proposed system can classify facial expression from an image of JAFFE database in 25.14 seconds whereas 33.92 seconds is required for recognizing facial expression from an image of Cohn-Kanade database.

Table 13: Average computational time (in second) required for recognizing the facial expression from a test image.

Database	Feature Region Detection	Feature Point Extraction	Recognition	Total (in second)
JAFFE (256×256)	5.44	16.09	3.61	25.14
Cohn-Kanade (640×490)	5.38	25.11	3.43	33.92

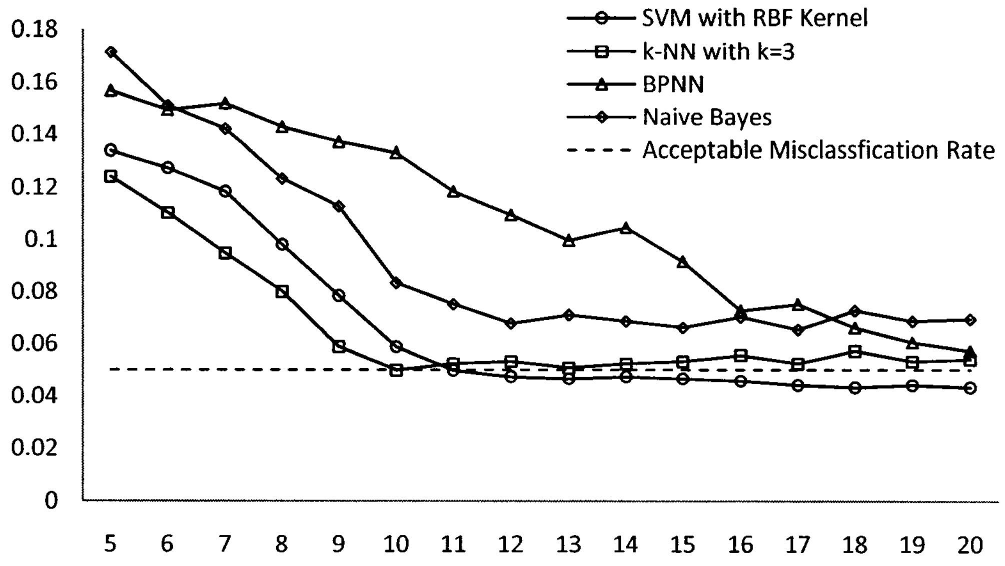


Figure 35: Obtained error rate using different classifiers with respect to the number of partitions of the training data used for cross validation. Here x-axis represents the number of partition and y-axis represents the error rate calculated using equation (5.30).

While tuning the k-NN, Neural Network and Naive Bayes classifier using the cross validation procedure towards obtaining the acceptable maximum misclassification rate of 0.05, it has been observed SVM achieves this acceptable rate comparatively faster than Neural Network and Naive Bayes classifier using a small number of partition of the training dataset. For neural network, this acceptable maximum error rate was achieved with more than 20 partitions of the training data and with Naive Bayes classifier, this rate was not achieved even using 40 partitions. Although k-NN achieves this acceptable error rate using only 10 partition of the training and validation dataset, it fails to maintain this rate with the increasing number of partitions. As, training time of the classifiers increases linearly with the number of partition in the training data used for the purpose of cross validation, it simply signifies that better generalization over the training and validation data set can be achieved using SVM with comparatively less amount of training than that of the Neural Network and Naive Bayes classifier.

6.5. Comparison with Existing Methods

Performance of the proposed method of facial expression classification has been compared with other previously proposed methods, listed in Table 1. Experimental results reported in the literature have been used for the purpose of this comparison, and findings of the comparative analysis have been summarized in Table 14. Since there is no standard bench marked database for testing as well as comparing the performance of facial expression classification methods, it is difficult to make concluding remarks on the classification performance of the systems that have been tested using different non-uniform datasets. However, several methods have been reported in the literature to be tested on a common database, namely the Japanese Female Facial Expression (JAFFE) Database [100]. So, another comparison has been made using the proposed method with the other techniques that have been tested on the JAFFE database only. Summary of findings from this comparative analysis has been provided in Table 15. As none of the previously proposed method has reported about the time complexity, comparison on the basis of computational time of the proposed method with other earlier methods of facial expression classification has been ignored in this thesis.

Table 14: Comparison of the proposed fully automated technique of facial expression classification with other previously proposed methods summarized in Table 1.

Method Proposed by	Test Cases	Accuracy	Method of Feature Point/ Feature Region Selection
Vanger et al. [20]	60 images	70%	Manual
Matsuno et al. [21]	44 images taken from 11 subjects	88.25%	Manual
Padgett and Cottrell [22]	84 Ekman's photos [7]	86%	Manual
Black and Yacoob [23]	70 image sequences of 40 subjects	92%	Manual
Edwards et al. [24]	200 images of 25 subjects	74%	Manual
Kobayashi and Hara [28]	19 Japanese students	90%	Manual
Hara et al. [31]	90 image sequences of 15 subjects	85%	Manual
Cohn et al. [35]	504 image sequences of 100 subjects	88%	Semi-Automated
Pantic and Rothkrantz [37]	456 dual views of 8 subjects	91%	Automated
Michel and Kaliouby [39]	Total 72 test samples (12 for each of the 6 basic emotion)	86%	Automated
Yoneyama et al. [44]	40 images of 10 subjects	92%	Manual
Essa and Pentland [47]	30 sequences of 8 subjects	97.8%	Automated
Mase and Pentland [16]	Single subject with 20 training and 30 and testing sequences	86%	Manual
Yacoob and Davis [48]	46 image sequences of 32 subjects	88%	Manual
Wang et al. [50]	29 sequences of 8 subjects	95%	Manual
Method proposed in this Thesis	213 images of 10 Japanese females [100], and 2450 images of 35 subjects [101]	92.04% and 86.33% respectively	Automated

As can be observed from Table 14, the proposed method of facial expressions classification performs better than most of the previously proposed techniques when tested over the JAFFE database. Only two methods proposed by Essa and Pentland [47], and Wang et al. [50] perform comparatively better than the proposed technique in terms of recognition accuracy. However, it should be noted that the method proposed by Wang et al. [50] requires hand labeling of the first frame and manual localization of the used 19 facial feature points (please refer to Table 1), which has been carried out in the proposed method in a fully automated manner. The method proposed by Essa and Pentland [47] can classify the facial expressions only into the five expression categories namely, smile, surprise, anger and disgust; but the method proposed in this thesis can classify facial expressions into seven emotion categories. In terms of test cases, the method of [47] has been tested on 30 image sequences of eight subjects, whereas the robustness of the proposed method has been examined by testing it over two different facial expression databases each of which contains comparatively larger test cases. It can also be observed from Table 14 that recognition accuracy of the methods proposed by Black and Yacoob [23], Kobayashi and Hara [28], Pantic and Rothkrantz [37], Yoneyama et al. [44] are very closer to that of the method proposed in this thesis. However, the proposed method is considered better than the methods proposed in [23], [28] and [44] since these methods require manual localization of feature points or feature regions for the purpose of feature extraction. The method of Pantic and Rothkrantz [37] ignores the neutral facial expression and can classify facial expressions into the six basic emotion categories. Amount of data used for testing their method is also comparatively less than that of the method proposed in this thesis.

Table 15: Comparison of the proposed method of facial expression classification with the previously proposed methods that have been tested on the Japanese Female Facial Expression (JAFFE) Database.

Method Proposed by	Test Cases	Accuracy	Method of Feature Point/ Feature Region Selection
Feng et al. [26]	The JAFFE Database [100], containing 213 images of seven facial expressions collected from 10 Japanese females	94.6%	Manual
Yu and Bhanu [27]		80.95%	Automated
Guo and Dyer [40]		92.4%	Manual
Sohail and Bhattacharya [42]		91%	Automated
Zhang et al. [52]		90%	Manual
Lyons et. al [54]		92%	Manual
Cheng et al. [55]		85.2%	Automated
Method proposed in this thesis		92.04%	Automated

On the other hand, the achieved classification accuracy using the proposed method over the Cohn-Kanade AU-coded facial expression database [101] is 86.33%, which is quite satisfactory considering the intensity variation of the images of this database. However, this recognition rate is less than that of the methods proposed previously in [21, 23, 28, 35, 37, 44, 47, 48 and 50]. Among them, the methods proposed in [21, 23, 28, 44, 48, and 50] are not automated since they require manual intervention in detecting the used feature points or isolating the facial feature regions. The method proposed by Cohn et al. [35] is a semi-automated method as manual initialization has to be done by hand-labeling the first frame of the image sequences. The methods of Pantic and Rothkrantz [37], as well as Essa and Pentland [47], are automated methods like that of the proposed one, but none of them can classify the facial expressions into seven emotion categories. Besides, the data sets used for testing their methods contain samples from fewer subjects comparative to the subset of Cohn-Kanade AU-coded facial expression database used for testing the performance of the proposed method.

As can be observed from Table 15 regarding the comparison of the proposed method with the previously proposed methods that has been tested on JAFFE database, the proposed method performs significantly better than all of the automated methods proposed in [27], [42], and [55]. Although our previously proposed *k*-NN based method [42] performs very closer to that of the method discussed in this thesis, it should be noted that, the facial expression classification technique of [42] cannot handle neutral face and thus classify the facial expressions into six basic emotion categories only. Besides, it has been observed from practical experiments that the eleven feature point based feature extraction technique used in [42] fails to perform well when the number of samples in the training set is large. Although the methods proposed by Feng et al. [26] and Guo and Dyer [40] performs apparently better than the proposed method over the same database, feature extraction procedure of these two methods are dependent on manual intervention and thus cannot be considered better that the method proposed in this thesis. So, from the comparative analysis with other previously proposed techniques, it can be concluded that the proposed method of facial expression classification is significantly in better many ways than most of the earlier methods found in literature.

Chapter 7

Conclusions and Future Directions

In this thesis, the development technique of a fully automated computer vision system for classifying seven important facial expressions has been discussed. For capturing the features of facial expression, a new analytic face model has been proposed (please refer to Section 2.4) using carefully selected fifteen feature points that demonstrate comparatively less tendency of being rejected during the detection process. Rather than classifying the facial expressions into the six basic emotion categories (anger, disgust, fear, happiness, sadness and surprise) like most of the previous works (please refer to Section 1.2 and Table 1), a way of handling the "neutral" face has been introduced for classifying the facial expressions into seven emotion categories (neutral, anger, disgust, fear, happiness, sadness and surprise). Towards detecting the fifteen feature points for feature extraction, their corresponding search areas over the face (i.e. eyebrows, eyes, and nose and mouth) have been separated at a prior stage by deploying an anthropometric face model based technique of facial feature region isolation [57]. Detection of the eleven future points from the isolated eye, eyebrow and nose regions have been performed implementing a standard image processing based multi-detector approach of facial feature point localization. For detecting the rest of the four points from the mouth region, isolation of the lip contour has been carried out implementing the level set method of image segmentation [58]. Since, the traditional level set method suffers from the drawback of costly re-initialization procedure, a newly introduced variational formulation known as the level set method without re-initialization [5] has been implemented that provides the facility of flexible initialization besides avoiding the costly procedure of re-initialization. In addition, this method has demonstrated significant robustness in lip contour detection against the shape variability of lip contour due to different facial expressions [58].

For the recognition part of the proposed facial expression classification system, a multi-Class SVM classifier, constructed by arranging 21 $\left\{\frac{7\times(7-1)}{2}\right\}$ independent $1-v-1$ SVM using the Decision Directed Acyclic Graph (DDAG) approach [59], has been implemented. In addition to better classification accuracy, training time of this method has been proven to be independent of the number of classes N, and thus remains significantly less than that of the other methods of constructing N-class SVM. Further efficiency of classification has been achieved by combining six trained multi-class SVM classifier applying the *Voting* Scheme [60] based Multiexpert classifier combining method.

Performance of the proposed method has been tested over two different facial expressions databases instead of confining the performance analysis process on a single database, as done in the previous works. In addition, accuracy of the proposed SVM based facial expression classification system has been compared using three other classification methods namely, *k*-NN, Neural Network and Naive Bayes Classifier towards establishing the supremacy of SVM in classifying facial expressions. Moreover, relative comparison of the proposed method of classifying facial expression has also been carried out with some other previously proposed methods and the findings of the comparison have been documented in Table 14 and Table 15.

As can be observed from the experimental results of Table 9 and Table 11, the proposed method performs better when combined multi-class SVM is used as the classifier rather than *k*-NN, Neural Network or Naive Bayes Classifier and provides average successful recognition rates of 92.04% and 86.33% respectively over the JAFFE database [100] and the Cohn-Kanade database of facial expression [101]. Comparative analysis with the performance of some other earlier methods (please refer to Table 14 and Table 15) reveals that the method proposed in this thesis has the capability of providing relatively better recognition accuracy than most of the previously proposed method listed in Table 1. Although some of the methods proposed earlier (Please refer to Section 1.2 and Table 1) show moderately better recognition rate than that of the proposed method, it should be noted that detection of feature points/feature regions for feature extraction has been carried out manually in those methods; which has been done in this work in a fully automated manner. Use of only fifteen feature points has enabled the proposed method to be computationally effective as compared to the other methods that work by identifying more feature points. Moreover, incorporation of the anthropometric model based facial feature regions localization technique [57] has further reduced the computational time of the proposed method by confining the search space of the fifteen feature points. Since the aggregation of the emotional information in human-computer interfaces allows much more natural and efficient interaction paradigms to be established, it is believed that the proposed system for the automated classification of facial expressions can play an increasing role in building effective and intelligent multimodal interfaces for next generation.

7.1. Direction for Future Works

Based on the work of facial expression classification described in this thesis, the following can be suggested as possible directions of improvements as well as further research:

1. The proposed method of classifying facial expression works well for frontal face images of facial expressions. It can also handle the horizontally tilted face images (rotated over the x-axis), but cannot recognize facial expressions from the images that are tilted vertically (rotated over the y-axis). So, improvement of the proposed method can be carried out by incorporating necessary enhancements so that it can recognize facial expressions from the side-view of the face images, and thus can classify facial expression even if the face in the image is rotated vertically over the y-axis.

2. The recognition functionality of the proposed method is limited to seven expression categories: neutral, anger, disgust, fear, happiness, sadness and surprise. The number of AUs that are handled by the proposed method in this respect is also restricted. So, the method described in this thesis can be extended further towards increasing its recognition functionality to more than seven categories. One possible way of doing this is to identify all the 46 AUs and to categorize them to different

facial expressions based on their inherent relationship with each individual facial expression.

3. Considering the present status of image processing and computer vision research, expression recognition by tracking the feature point is assumed to be the most possible approach regarding its real time implementation. So, improvement of the proposed method towards classifying facial expressions in real-time from video is considered as a possible significant extension of the work described in this thesis.

4. Classification of facial expression is considered as a part of the emotion recognition process, which modulates almost all modes of human communication – facial expression, gestures, posture, tone of voice, choosing of words, respiration, skin temperature, clamminess, etc. [6]. So, another possible extension of the work, discussed here, can be the development of a multimodal emotion recognition system by fusing multisensory data that are responsible in forming human emotion either in an individual or in a combined manner.

References

[The references have been organized in the order of their citations in the thesis.]

[1] E. Boyle, A.H. Anderson, and A. Newlands, "The effects of visibility on dialogue and performance in a co-operative problem solving task," *Language and Speech,* vol. 37, no. 1, pp. 1–20, 1994.

[2] G.M. Stephenson, K. Ayling, and D.R. Rutter, "The role of visual communication in social exchange," *Britain Journal of Social Clinical Psychology,* vol. 15, pp. 113–120, 1976.

[3] M. Pantic, "Facial expression analysis by computational intelligence technique," Ph.D. Dissertation, Delft University of Technology, Netherlands, 2001.

[4] P. Ekman, and W.V. Friesen, *Unmasking the Face*, Prentice Hall, New Jersey, 1975.

[5] C. Li, C. Xu, C. Gui, and M.D. Fox, "Level set evolution without re-initialization: a new variational formulation," in *Proc. IEEE International Conference on Computer Vision and Pattern Recognition,* San Diego, CA, 2005, vol. 1, pp. 430–436.

[6] N. Sebe, I. Cohen, and T.S. Huang, "Multimodal emotion recognition," C.H. Chen, P.S.P. Wang (Eds.), *Handbook of Pattern Recognition and Computer Vision,* pp. 387–409, World Scientific Publishing, Singapore, 2005.

[7] G. Donato, M.S. Bartlett, J.C. Hager, P. Ekman, and T.J. Sejnowski, "Classifying facial actions," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 21, no. 10, pp. 974–989, 1999.

[8] A. Mehrabian, "Communication without words," *Psychology Today*, vol. 2, no. 4, pp. 53–56, 1968.

[9] L. van Poecke, *Nonverbal Communication,* Garant-Uitgevers, Apeldoorn, Netherlands, 1996.

[10] A. van Dam, "Beyond WIMP," *IEEE Computer Graphics and Applications,* vol. 20, no. 1, pp. 50–51, 2000.

[11] A. Pentland, "Looking at people: sensing for ubiquitous and wearable computing," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 22, no. 1, pp. 107–119, 2005.

[12] C. Darwin, *The Expression of the Emotions in Man and Animals,* J. Murray, London, 1872.

[13] P. Ekman, and W.V. Friesen, "Constants across cultures in the face and emotion," *Journal of Personality and Social Psychology,* vol. 17, no. 2, pp. 124–129, 1971.

[14] M. Suwa, N. Sugie, and K. Fujimora, "A preliminary note on pattern recognition of human emotional expression," in *Proc. 4th International Joint Conference on Pattern Recognition,* 1978, pp. 408–410.

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

[16] K. Mase, and A. Pentland, "Recognition of facial expression from optical flow," *Institute of Electronics, Information and Communication Engineers Trans.,* vol. E74, no. 10, pp. 3474–3483, 1991.

[17] M. Pantic, and L.J.M. Rothkrantz, "Automatic analysis of facial expressions: the state of the art," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 22, no. 12, pp. 1424–1443, December, 2000.

[18] G.W. Cottrell, and J. Metcalfe, "EMPATH: Face, emotion, gender recognition using holons," *Advances in Neural Information Processing System*, vol. 3, 1991, pp. 564–571.

[19] A. Rahardja, A. Sowmya, and W.H. Wilson, "A neural network approach to component versus holistic recognition of facial expression in images," *Intelligent Robots and Computer Vision X: Algorithms and Techniques,* vol. 1607, pp. 62–70, 1991.

[20] P. Vanger, R. Honlinger, and H. Haken, "Applications of synergetics in decoding facial expression of emotion," in *Proc. IEEE International Conference on Automatic Face and Gesture Recognition,* Zurich, 1995, pp. 24–29.

[21] K. Matsuno, C.W. Lee, and S. Tsjui, "Recognition of facial expressions with potential net," in *Proc. Asian Conference on Computer Vision,* 1993, pp. 504–507.

[22] C. Padgett, and G.W. Cottrell, "Representing face images for emotion classification" in *Proc. International Conference on Advances in Neural Information Processing Systems*, 1996, pp. 894–900.

[23] M.J. Black, and Y. Yacoob, "Recognizing facial expressions in image sequences using local parameterized models of image motion," *International Journal of Computer Vision,* vol. 25, no. 1, pp. 23–48, 1997.

[24] G.J. Edwards, T.F. Cootes, and C.J. Taylor, "Face recognition using active appearance models," in *Proc. European Conference on Computer Vision,* Freiburg, Germany, June, 1998, vol. 2, pp. 581–695.

[25] T.F. Cootes, G.J. Edwards, and C.J. Taylor, "Active appearance models," in *Proc. European Conference on Computer Vision,* Freiburg, Germany, June, 1998, vol. 2, pp. 484–498.

[26] X. Feng, M. Pietikainen, and A. Hadid, "Facial expression recognition with local binary patterns and linear programming," *Pattern Recognition and Image Analysis,* vol. 15, no. 2, pp. 546–548, 2005.

[27] J. Yu, and B. Bhanu, "Evolutionary feature synthesis for facial expression recognition," *Pattern Recognition Letters,* vol. 27, pp. 1289–1298, 2006.

[28] H. Kobayashi, and F. Hara, "Recognition of mixed facial expressions and their strength by neural network," in *Proc. IEEE International Workshop on Robot and Human Communication,* Tokyo, 1992, pp. 381–386.

[29] J. Ding, M. Shimamura, H. Kobayashi, and T. Nakamura, "Neural network structures for expression recognition," in *Proc. International Joint Conference on Neural Network,* Nagoya, Japan, October 25-29, 1993, pp. 1420–1423.

[30] J. Zhao, and G. Kearney, "Classifying facial emotion by back-propagation neural networks with fuzzy inputs," in *Proc. International Conference on Neural Information Processing,* 1996, vol. 1, pp. 454–457.

[31] F. Hara, and H. Kobayashi, "Facial interaction between 3-D face robot and human beings," in *Proc. IEEE International Conference on Systems, Man and Cybernetics,* Florida, 1997, pp. 3732–3737.

[32] H. Ushida, T. Takagi, and T. Yamaguchi, "Recognition of facial expressions using the conceptual fuzzy sets," in *Proc. Second International Conference on Fuzzy Systems 1,* San Francisco, California, 1993, pp. 594–599.

[33] G.D. Kearney, and S. McKenzie, "Machine interpretation of emotion: design of a memory-based expert system for interpreting facial expressions in terms of signaled emotions (JANUS)," *Cognitive Science,* vol. 17, no. 4, pp. 589–622, 1993.

[34] P. Ekman, W.V. Friesen, and S. Tomkins, "Facial affect scoring technique: a first validity study," *Semiotica,* vol. 3, pp. 37–58, 1971.

[35] J.F. Cohn, A.J. Zlochower, J.J. Lien, and T. Kanade, "Feature-point tracking by optical flow discriminates subtle differences in facial expression," in *Proc. Third IEEE International Conference on Automatic Face and Gesture Recognition*, Nara, Japan, 1998, pp. 196–401.

[36] B.D. Lucas, and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. International Joint Conference on Artificial*

Intelligence, 1981, pp. 674–680.

[37] M. Pantic, L.J.M. Rothkrantz, "Expert system for automatic analysis of facial expressions," *Image and Vision Computing Journal,* vol. 18, no. 11, pp. 881–905, 2000.

[38] M. Kass, A. Witkin, and D. Terzopoulos, "Snake: active contour model," *International Journal of Computer Vision,* vol. 1, no. 4, pp.321–331, 1987.

[39] P. Michel, and R.E. Kaliouby, "Real time facial expression recognition in video using support vector machines," in *Proc. 5th International Conference on Multimodal Interfaces,* Vancouver, Canada, 2003, pp. 258–264.

[40] G. Guo, and C.R. Dyer, "Learning from examples in the small sample case: face expression recognition," *IEEE Trans. Systems, Man and Cybernetics,* Part-B, vol. 35, no. 3, pp. 477–488, 2005.

[41] S.A. Dudani, "The distance-weighted *k*-nearest neighbor rule," *IEEE Trans. Systems, Man and Cybernetics,* vol. 6, pp. 325–327, 1976.

[42] A.S.M. Sohail, and P. Bhattacharya, "Classification of facial expressions using *k*-nearest neighbor classifier," A. Gagalowicz, W. Philips (Eds.), *Advances in Computer Vision and Computer Graphics*, Lecture Notes in Computer Science, vol. 4418, pp. 555–566, Springer-Verlag, Berlin Heidelberg New York, 2007.

[43] A.S.M. Sohail, and P. Bhattacharya, "Detection of facial feature points using anthropometric face model," in *Proc. IEEE International Conference on Signal-Image Technology and Internet-Based Systems,* Hammamet, Tunisia, 2006, pp. 656–665.

[44] M. Yoneyama, Y. Iwano, A. Ohtake, and K. Shirai, "facial expressions recognition using discrete hopfield neural networks," in *Proc. IEEE International Conference on Image Processing,* 1997, vol. 3, pp. 117–120.

[45] I. Kanter, and H. Sompolinsky, "Associative recall of memory without errors," *Physical Review,* vol. 35, no. 1, pp. 380–392, 1987.

[46] I.A. Essa, "Analysis, interpretation and synthesis of facial expressions," Ph.D. Dissertation, Massachusetts Institute of Technology, Boston, USA, 1995.

[47] I.A. Essa, and A. Pentland, "Coding, analysis, interpretation and recognition of facial expressions," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 19, no. 7, pp. 757–763, 1997.

[48] Y. Yacoob, and L. Davis, "Recognizing human facial expression," *University of Maryland*, Technical Report CS-TR-3265, May, 1994.

[49] M. Rosenblum, Y. Yacoob, and L.S. Davis, "Human emotion recognition from motion using a radial basis function network architecture," *University of Maryland*, Technical Report CS-TR-3304, June, 1994.

[50] M. Wang, Y. Iwai, and M. Yachida, "Expression recognition from time sequential facial images by use of expression change model," in *Proc. Third IEEE International Conference on Automatic Face and Gesture Recognition,* Nara, Japan, 1998, pp. 324–329.

[51] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery, *Numerical Recipes in C,* Cambridge University Press, New York, USA, 1992.

[52] Z. Zhang, M. Lyons, M. Schuster, and S. Akamatsu, "Comparison between geometry-based and gabor wavelets-based facial expression recognition using multi-layer perceptron," in *Proc. Third IEEE International Conference on Automatic Face and Gesture Recognition*, Nara, Japan, 1998, pp. 454–459.

[53] M. Riedmiller, and H. Braun, "A direct adaptive method for faster back-propagation learning: The RPROP algorithm," in *Proc. IEEE International Conference on Neural Networks,* San Francisco, California, 1993, pp. 586–591.

[54] M.J. Lyons, J. Budynek, and S. Akamatsu, "Automatic classification of single facial imges," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 21, no. 12, pp. 1357–1362, 1999.

[55] S. Cheng, M. Chen, H. Chang, and T. Chao, "Semantic-based facial expression recognition using analytical hierarchy process," *Expert Systems with Application,* vol. 33, pp. 86–95, 2007.

[56] T.L. Saaty, *The Analytic Hierarchy Process,* McGraw-Hill, New York, 1980.

[57] A.S.M. Sohail, and P. Bhattacharya, "Localization of facial feature regions using anthropometric face model," in *Proc. First International Conference on Multidisciplinary Information Sciences and Technologies,* Merida, Spain, October, 2006.

[58] A.S.M. Sohail, and P. Bhattacharya, "Automated lip contour detection using the level set segmentation method," accepted for inclusion in *Proc. 14th International Conference on Image Analysis and Processing,* Modena, Italy, September, 2007.

[59] J. Platt, N. Cristianini, and J. Shawe-Taylor, "Large margin DAGs for multiclass classification," *Advances in Neural Information Processing Systems,* vol. 12, pp. 547–553, 2000.

[60] E. Alpaydin, *Introduction to Machine Learning,* MIT Press, Cambridge, 2004.

[61] V. Bruce, *Recognizing Faces,* Lawrence Erlbaum, Hove, East Sussex, 1986.

[62] H. Yamada, "Visual information for categorizing facial expressions of emotions," *Applied Cognitive Psychology*, vol. 7, pp. 257–270, 1993.

[63] P. Ekman, *Emotion in the Human Face,* Cambridge University Press, Cambridge, 1982.

[64] A.J. Fridlund, P. Ekman, and H. Oster, "Facial expressions of emotion: review

literature 1970–1983," A.W. Siegman, S. Feldstein (Eds.), *Nonverbal Behavior and Communication,* pp. 143–224, Lawrence Erlbaum Associates, Hillsdale, NJ, 1987.

[65] C.E. Izard, *The Face of Emotion,* Appleton-Century-Crofts, New York, 1971.

[66] J.A. Russell, "Is there universal recognition of emotion from facial expression? A review of cross-cultural studies," *Psychological Bulletin,* vol. 115, no. 1, pp. 102–141, 1994.

[67] P. Ekman, and W.V. Friesen, *Facial Action Coding System (FACS) Manual,* Consulting Psychologists Press, Palo Alto, 1978.

[68] G. Faigin, *The Artist's Guide to Facial Expressions,* Watson-Guphill Publications, 1990.

[69] G. Johansson, "Visual perception of biological motion and a model for its analysis," *Perception and Psychophysics,* vol. 14, pp. 201–211, 1973.

[70] J.N. Bassili, "Facial motion in the perception of faces and of emotional expression," *Journal of Experimental Psychology: Human Perception and Performance,* vol. 4, pp. 373–379, 1978.

[71] D. Terzopoulos, and K. Waters, "Analysis and synthesis of facial image sequences using physical and anatomical models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 6, pp. 569–579, 1993.

[72] N.M. Thalmann, P. Kalra, and M. Escher, "Face to virtual face," *Proceedings of the IEEE,* vol. 86, no. 5, pp. 870–883, 1998.

[73] L. Xhang, and P. Lenders, "Knowledge-based eye detection for human face recognition," in *Proc. Fourth IEEE International Conference on Knowledge-Based Intelligent Engineering Systems and Allied Technologies,* Brighton, UK, August 2000, vol. 1, pp. 117–120.

[74] M. Rizon, and T. Kawaguchi, "Automatic eye detection using intensity and edge information," in *Proc. IEEE TENCON 2000,* vol. 2, pp. 415–420.

[75] A.L. Yuille, P.W. Hallinan, and D.S. Cohen, "Feature extraction from faces using deformable templates," International *Journal of Computer Vision*, vol. 8, no. 2, pp. 99–111, 1992.

[76] R. Brunelli, and T. Poggio, "Face recognition: Features versus templates," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 15, no. 10, pp. 1042–1062, 1993.

[77] R. Herpers, and G. Sommer, "An Attentive Processing Strategy for the Analysis of Facial Features," H. Wechsler et al. (eds.), *Face Recognition: From Theory to Applications*, pp. 457–468, Springer-Verlag, Berlin Heidelberg New York, 1998.

[78] M. Pardas, and M. Losada, "Facial parameter extraction system based on active contours," in *Proc. International Conference on Image Processing,* Thessaloniki, Greece, October 2001, vol. 1, pp. 1058–1061.

[79] T. Kawaguchi, D. Hidaka, and M. Rizon, "Detection of eyes from human faces by Hough Transform and separability filter," in *Proc. International Conference on Image Processing,* Vancouver, Canada, 2000, pp.49–52.

[80] C.A. Perez, A. Palma, C.A. Holzmann, and C. Pena, "Face and eye tracking algorithm based on digital image processing," in *Proc. IEEE International Conference on Systems, Man and Cybernetics,* Tucson, Arizona, October 2001, vol. 2, pp. 1178–1183.

[81] R.L. Hsu, M. Abdel-Mottaleb, and A.K. Jain, "Face detection in color images," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 24, no. 5, pp. 696–706, 2002.

[82] H.C. Kim, D. Kim, and S.Y. Bang, "A PCA mixture model with an efficient model selection method," in *Proc. IEEE International Joint Conference on Neural*

Networks, Washington , DC , July 2001, vol. 1, pp. 430–435.

[83] S. Phimoltares, C. Lursinsap, and K. Chamnongthai, "Locating essential facial features using neural visual model," in *Proc. First International Conference on Machine Learning and Cybernetics,* Beijing, China, November 2002, pp. 1914–1919.

[84] H.W. Lee, S.K, Kil, Y. Han, and S.H. Hong, "Automatic face and facial feature detection," in *Proc. IEEE International Symposium on Industrial Electronics,* Pusan, Korea, June 2001, pp. 254–259.

[85] P.I. Wilson, and J. Fernandez, "Facial feature detection using Haar classifiers," *Journal of Computing Sciences in Colleges,* vol. 21, no. 4, 2006.

[86] M. Dooley, "Anthropometric modeling programs – a survey," *IEEE Computer Graphics and Applications*, vol. 2, pp. 17–25, November 1982.

[87] L.G. Farkas, *Anthropometry of the Head and Face*, Raven Press, New York, 1994.

[88] S.L. Rogers, *Personal Identification from Human Remains*, Charles C. Thomas Publisher Limited, Springfield, Illinois, 1984.

[89] A. Hrdlicka, Practical *Anthropometry*, AMS Press, New York, 1972.

[90] C. Gordon, *Anthropometric Survey of U.S. Army Personnel: Methods and Summary Statistics 1988*, United States Army Natick Research, Development and Engineering Center, 1989.

[91] L.G. Farkas, *Anthropometric Facial Proportions in Medicine*, Thomas Books, Springfield, Illinois, 1987.

[92] I. Fasel, B. Fortenberry, and J. Movellan, "A generative framework for real time object detection and classification," *Computer Vision and Image Understanding,* vol. 98, pp, 182–210, 2005.

[93] J. Friedman and T. Hastie, and R. Tibshirani, "Additive logistic regression: a statistical view of boosting," Technical Report, Department of Statistics, Stanford University, 1998. Available online: http://citeseer.ist.psu.edu/friedman98additive.html

[94] Y. Freund, and R. Schapire, "A short introduction to boosting," *Journal of the Japanese Society for Artificial Intelligence,* vol. 14, no. 5, pp. 771–780, 1999.

[95] B.W. Silverman, *Density Estimation for Statistics and Data Analysis,* Chapman and Hall, 1986.

[96] H. Rowley, S. Baluja, and T. Kanade, "Neural network-based face detection," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 1, no. 20, pp. 23–28, 1998.

[97] P. Viola, and M.J. Jones, "Robust Real-Time Face Detection," *International. Journal of Computer Vision,* vol. 57, no. 2, pp. 137–154, 2004.

[98] G. Shakhnarovich, P. Viola, and B. Moghaddam, "A unified learning framework for real-time face detection and classification," *in Proc. 5th IEEE International Conference on Automatic Face and Gesture Recognition,* Washington, D.C., USA, May 2002, pp. 14–21.

[99] K.K. Sung, and T. Poggio, "Example based learning for view-based human face detection," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 20, no. 1, pp. 39–51, 1998.

[100] J. Lyons, S. Akamatsu, M. Kamachi, and J. Gyoba, "Coding facial expressions with Gabor wavelets," in *Proc. Third IEEE International Conference on Automatic Face and Gesture Recognition,* Nara, Japan, 1998, pp. 200–205.

[101] T. Kanade, J.F. Cohn, and Y. Tian, "Comprehensive database for facial expression analysis," in *Proc. 4th IEEE International Conference on Automatic Face and Gesture Recognition*, Grenoble, France, 2000, pp. 46–53.

[102] R.C. Gonzalez, and R.E. Woods, *Digital Image Processing,* 2nd edn., Prentice Hall, New Jersey, 2002.

[103] N. Otsu, "A threshold selection method from gray level histograms," *IEEE Trans. Systems, Man, and Cybernetics,* vol. 9, no. 1, pp. 62–66, 1979.

[104] G.X. Ritter, and J.N. Wilson, *Handbook of Computer Vision Algorithms in Image Algebra*, CRC Press, Boca Raton, USA, 1996.

[105] A. Mcandrew, *Introduction to Digital Image Processing with Matlab,* Thomson Course Technology, 2004

[106] M. Sonka, V. Hlavac, and R. Boyle, *Image Processing, Analysis and Machine Vision,* International Thomson Computer Press, Belmont, CA 94002, 1993.

[107] J.R. Parker, *Algorithms for Image Processing and Computer Vision,* John Wiley & Sons, 1997.

[108] D. Marr, and E. Hildreth, "Theory of edge detection," *Royal Society of London,* vol. B 207 pp. 187–217, 1980.

[109] W.K. Pratt, *Digital Image Processing,* 4th edn., John Wiley & Sons, 2007.

[110] S. Osher, and J.A. Sethian, "Fronts propagating with curvature dependent speed: algorithms based on Hamilton-Jacobi formulations," *Journal of Computational Physics,* vol. 79, no. 1, pp. 12–49, 1988.

[111] V. Caselles, F. Catte, T. Coll, and F. Dibos, "A geometric model for active contours in image processing," *Numerische Mathematik,* vol. 66, no. 1, pp. 1–31, 1993.

[112] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic active contours," *International Journal of Computer Vision*, vol. 22, no. 1, pp. 61–79, 1997.

[113] R. Malladi, J.A. Sethian, and B.C. Vemuri, "Shape modeling with front propagation: a level set approach," *IEEE Trans. Pattern Analysis and Machine*

Intelligence, vol. 17, no. 2, pp. 158–175, 1995.

[114] S. Osher, and R. Fedkiw, *Level Set Methods and Dynamic Implicit Surfaces*, Springer-Verlag, New York, 2002.

[115] D. Peng, B. Merriman, S. Osher, H. Zhao, and M. Kang, "A PDE-based fast local level set method", *Journal of Computational Physics*, vol. 155, no. 2, pp. 410–438, 1999.

[116] M. Sussman, and E. Fatemi "An efficient, interface-preserving level set redistancing algorithm and its application to interfacial incompressible fluid flow," *SIAM Journal on Scientific Computing*, vol. 20, no.4, pp. 1165–1191, 1999.

[117] J.A. Sethian, *Level Set Methods and Fast Marching Methods*, Cambridge University Press, Cambridge, 1999.

[118] V.I. Arnold, *Geometrical Methods in the Theory of Ordinary Differential Equations,* Springer-Verlag, NY, 1983.

[119] L. Evans, *Partial Differential Equations,* Providence: American Mathematical Society, 1998.

[120] B. Vemuri, and Y. Chen, "Joint image registration and segmentation," S. Osher, and N. Paragios, (Eds.), *Geometric Level Set Methods in Imaging, Vision, and Graphics*, pp. 251–269, Springer-Verlag, Berlin Heidelberg New York, 2003.

[121] V. Vapnik, *Statistical Learning Theory,* Wiley, New York, 1998.

[122] K. Roy, "Iris recognition using Support Vector Machines," Master's Thesis, Concordia University, Montreal, Canada, 2006.

[123] V.N. Vapnik, The Nature of Statistical Learning Theory, 2nd edn., Springer-Verlag, New York, 2000.

[124] C. Cortes, and V. Vapnik, "Support-vector networks," *Machine Learning,* vol. 20,

no. 3, pp. 273-297, 1995.

[125] S.Y. Kung, M.W. Mak, and S.H. Lin, *Biometric Authentication: A Machine Learning Approach,* 1st edn., Prentice Hall Information and System Science Series, USA, 2005.

[126] B. Schölkopf, C. Burges, and V. Vapnik, "Extracting support data for a given task," in *Proc. First International Conference on Knowledge Discovery and Data Mining,* Menlo Park, CA, 1995, pp. 252–257.

[127] S. Knerr, L. Personnaz, and G. Dreyfus, "Single-layer learning revisited: a stepwise procedure for building and training a neural network," F. Fogelman Soulié and J. Hérault (eds.), *Neurocomputing: Algorithms, Architectures and Applications*, NATO ASI Series, vol. F68, pp. 41–50, Springer-Verlag, New York, 1990.

[128] J.H. Friedman, "Another approach to polychotomous classification," Technical Report, Department of Statistics, Stanford University, 1996. Available online: http://www-stat.stanford.edu/~jhf/ftp/poly.ps.Z

[129] J. Platt, "Fast training of support vector machines using sequential minimal optimization," B. Schölkopf, C.J.C. Burges, and A.J. Smola (eds.), *Advances in Kernel Methods - Support Vector Learning,* pp. 185–208. MIT Press, Cambridge, MA, 1999.

[130] R. Shinghal, *Pattern Recognition Techniques and Applications,* 1st edn., Oxford University Press, New Delhi, 2006.

[131] Y.S. Huang, and C.Y. Suen, "A method of combining multiple experts for the recognition of unconstrained handwritten numerals," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 17, no. 1, pp.90–94, 1995.

[132] "The Machine Perception Toolbox," developed by the Machine Perception Laboratory, University of California San Diego. Available online: http://mplab.ucsd.edu/grants/project1/free-

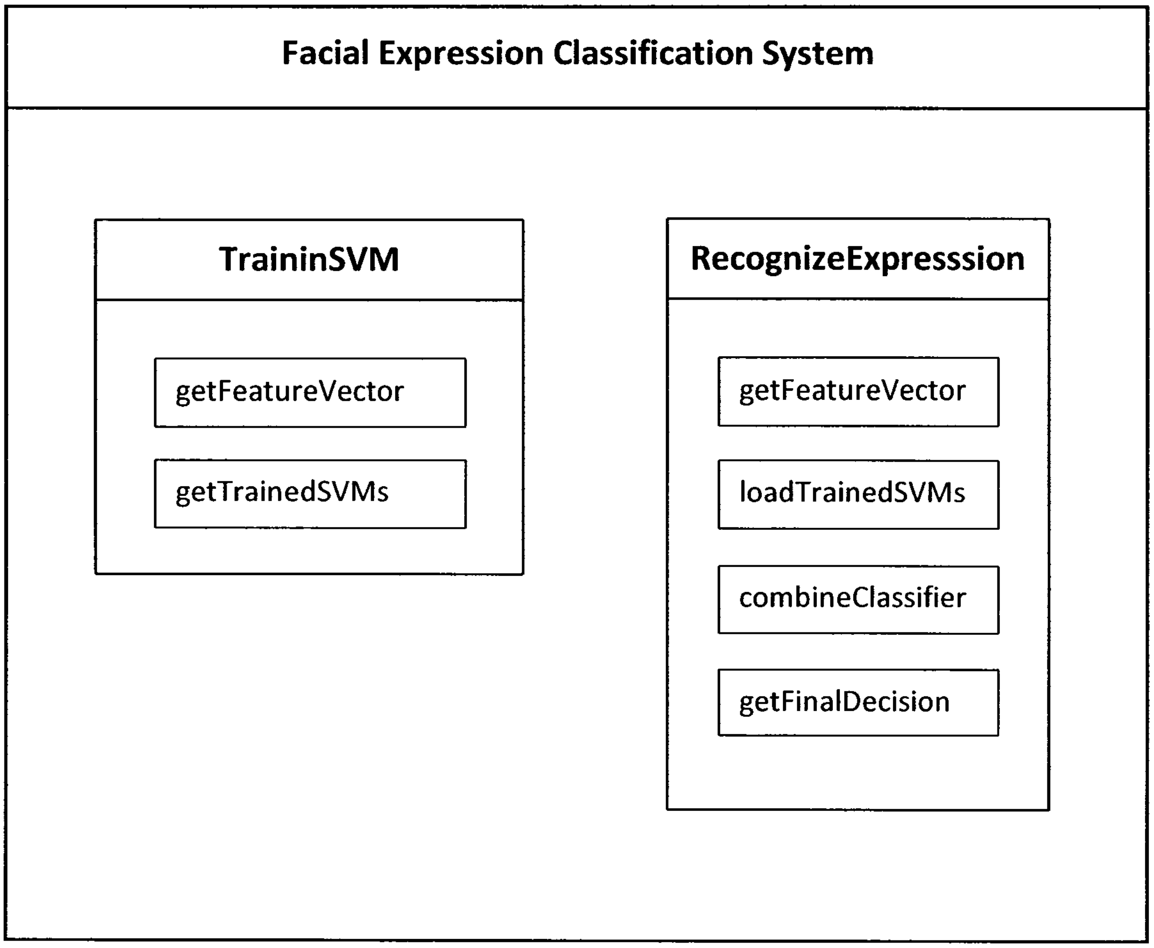
software/mptwebsite/introduction.html

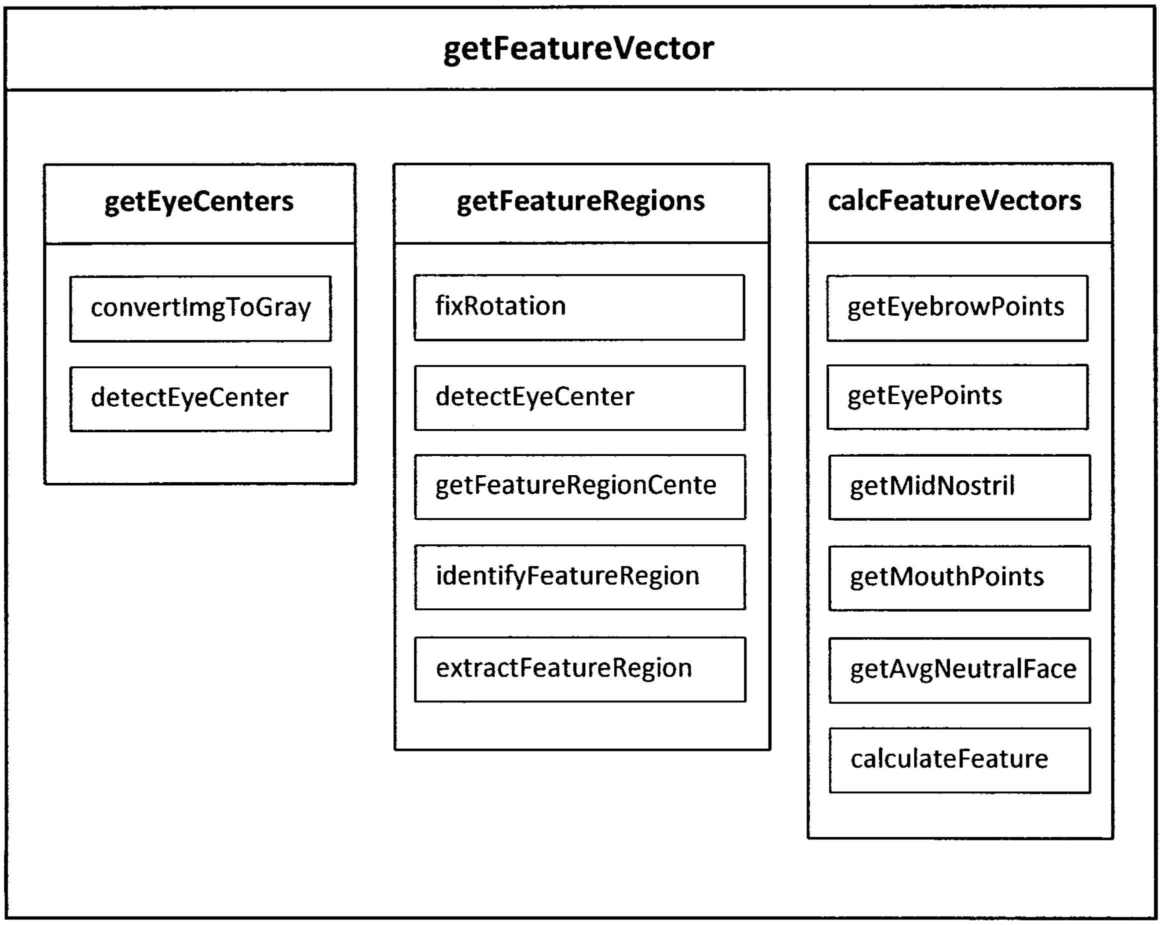
[133] "PRTools," developed by the Delft Pattern Recognition Group, Faculty of Applied Sciences, Delft University of Technology. Available online: http://www.prtools.org/

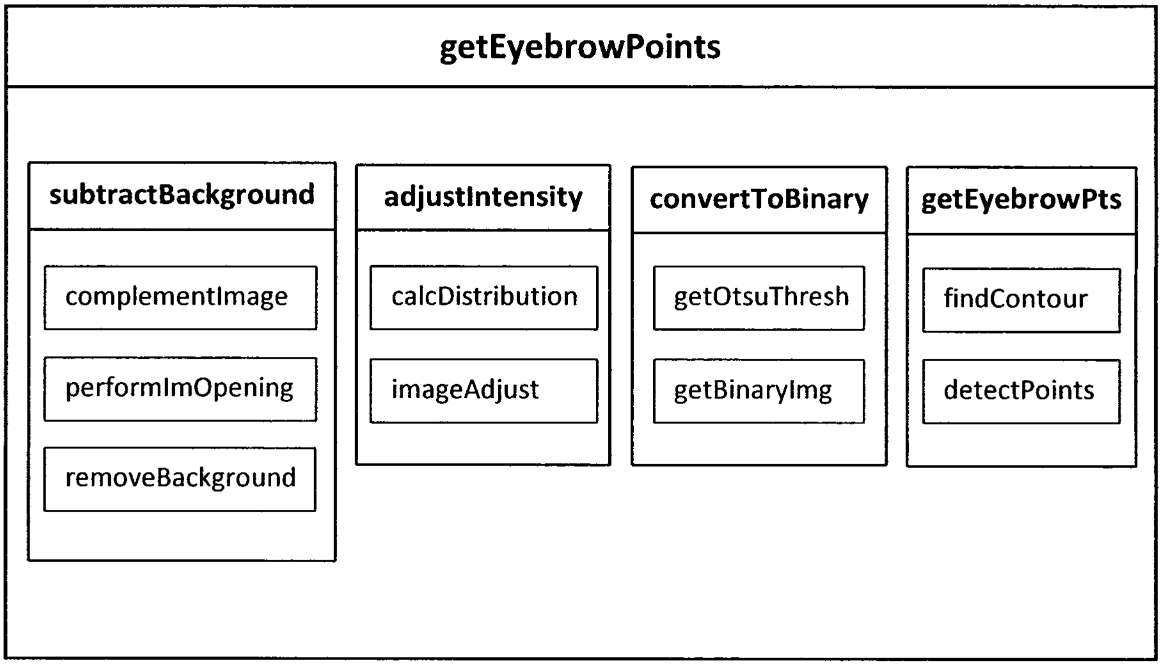
[134] Matlab code of "Level set method without re-initialization," developed by Chunming Li, Institute of Imaging Science, Vanderbilt University. Available online: http://vuiis.vanderbilt.edu/~licm/

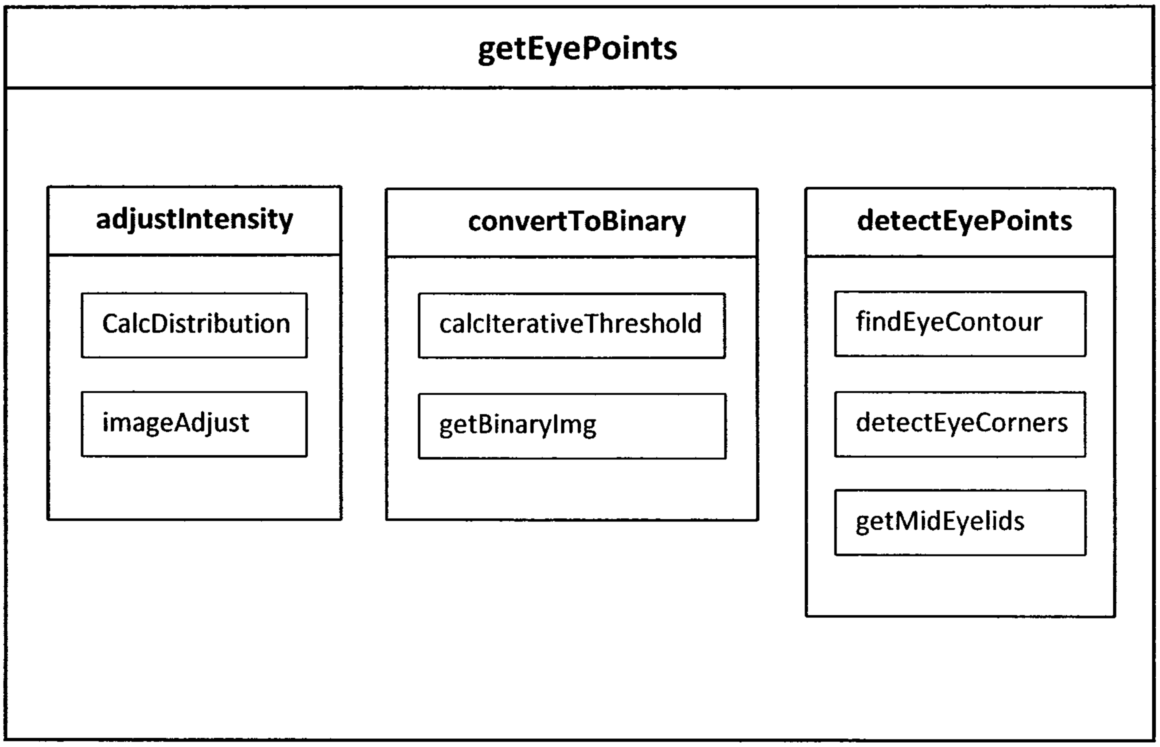
Appendix A

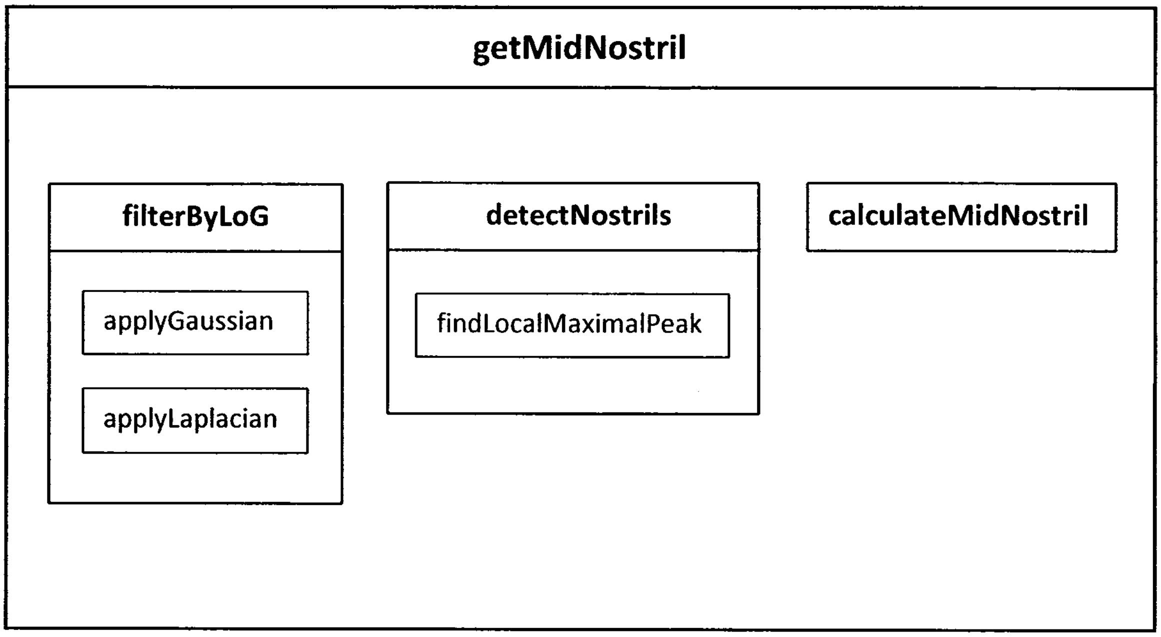
Prototype of the proposed automated facial expression classification system has been developed using Matlab® 7. Functional overview of the prototyped system is provided below:

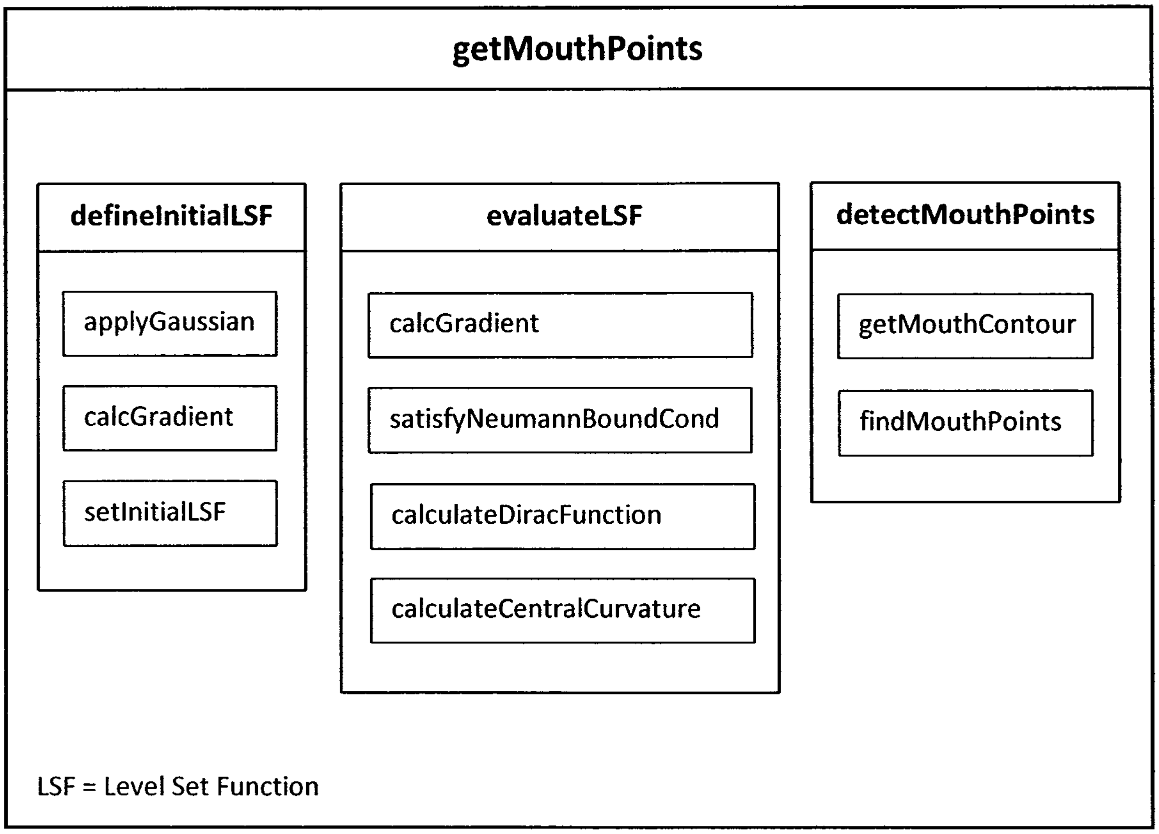


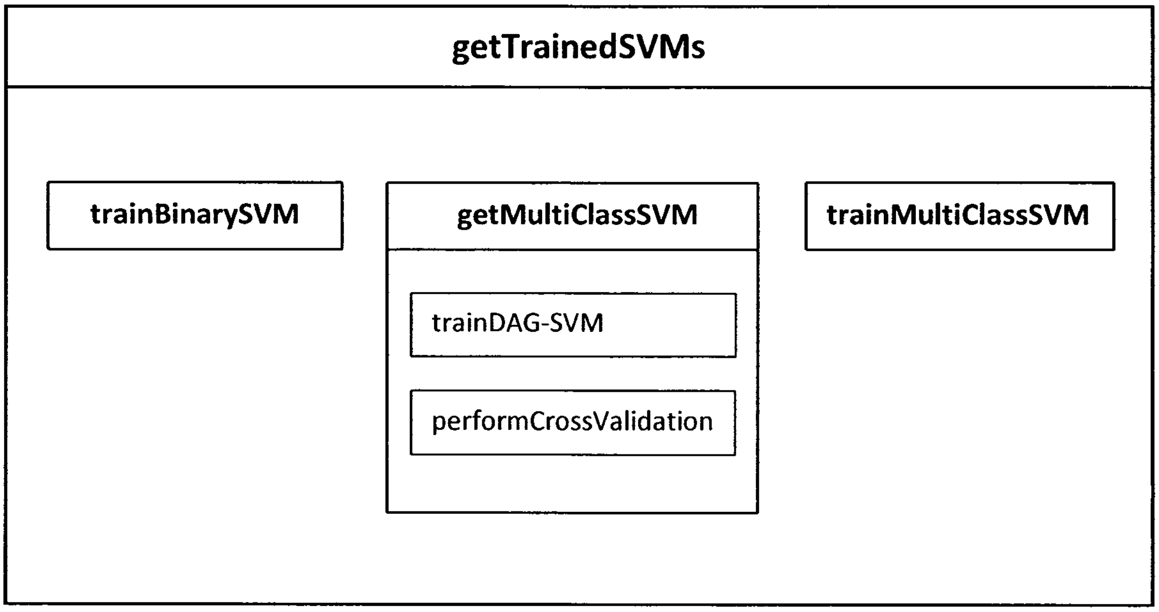

getFeatureVector
getEyeCenters
convertImgToGray
detectEyeCenter
getFeatureRegions
fixRotation
detectEyeCenter
getFeatureRegionCente
identifyFeatureRegion
extractFeatureRegion
calcFeatureVectors
getEyebrowPoints
getEyePoints
getMidNostril
getMouthPoints
getAvgNeutralFace
calculateFeature


getEyebrowPoints
subtractBackground
complementImage
performImOpening
removeBackground
adjustIntensity
calcDistribution
imageAdjust
convertToBinary
getOtsuThresh
getBinaryImg
getEyebrowPts
findContour
detectPoints


getEyePoints
adjustIntensity
CalcDistribution
imageAdjust
convertToBinary
calcIterativeThreshold
getBinaryImg
detectEyePoints
findEyeContour
detectEyeCorners
getMidEyelids


getMidNostril
filterByLoG
applyGaussian
applyLaplacian
detectNostrils
findLocalMaximalPeak
calculateMidNostril


getMouthPoints
defineInitialLSF
applyGaussian
calcGradient
setInitialLSF
evaluateLSF
calcGradient
satisfyNeumannBoundCond
calculateDiracFunction
calculateCentralCurvature
detectMouthPoints
getMouthContour
findMouthPoints
LSF = Level Set Function


getTrainedSVMs
trainBinarySVM
getMultiClassSVM
trainDAG-SVM
performCrossValidation
trainMultiClassSVM

Appendix B

The following research papers originated from the research work described in this thesis have been published or accepted for publication as well as presentation in the proceedings of different international conferences.

[1] A.S.M. Sohail, and P. Bhattacharya, "Support Vector Machines applied to automated categorization of facial expressions," accepted for inclusion and presentation in *Third Indian International Conference on Artificial Intelligence (IICAI-2007),* December 17-19, 2007, Pune, India. Proceeding to Appear on Lecture Notes in Computer Science (LNCS), Springer-Verlag, Berlin Heidelberg New York.

[2] A.S.M. Sohail, and P. Bhattacharya, "Detection of facial feature points using anthropometric face model," E. Damiani, A. Dipanda, K. Yetongnon, L. Legrand, and P. Schelkens, (Eds.), *Signal Processing for Image Enhancement and Multimedia Processing,* Multimedia Systems and Applications Series, vol. 34, Springer-Verlag, Berlin Heidelberg New York, November 2007.

[3] A.S.M. Sohail, and P. Bhattacharya, "Classifying facial expressions using point-based analytic face model and Support Vector Machines," accepted for inclusion and presentation in *IEEE International Conference on Systems, Man, and Cybernetics (SMC 2007),* October 7-10, 2007, Montreal, Canada.

[4] A.S.M. Sohail, and P. Bhattacharya, "Automated lip contour detection using the level set segmentation method," accepted for inclusion in *14th International Conference on Image Analysis and Processing (ICIAP-2007),* September 10-14, 2007, Modena, Italy.

[5] A.S.M. Sohail, and P. Bhattacharya, "Classification of facial expressions using *k*-Nearest Neighbor classifier," A. Gagalowicz, W. Philips (Eds.), *Advances in Computer Vision and Computer Graphics,* Lecture Notes in Computer Science, vol. 4418, pp. 555–566, Springer-Verlag, Berlin Heidelberg New York, 2007.

[6] A.S.M. Sohail, and P. Bhattacharya, "Localization of facial feature regions using anthropometric face model," in Proc. *First International Conference on Multidisciplinary Information Sciences and Technologies (INSCIT2006),* October 25-28, 2006, Merida, Spain.