AGE PREDICTION USING FACIAL FEATURE EXTRACTION

BY:

SONAL JAIN	**061293**
ARUNIMA SHARMA	**061410**
NIHAL BAJWA	**061431**
NEHA ALREJA	**061533**

MAY -2010

Submitted in partial fulfillment of the Degree of Bachelor of Technology

DEPARTMENT OF COMPUTER SCIENCE

JAYPEE UNIVERSITY OF INFORMAION TECHNOLOGY-WAKNAGHAT

CERTIFICATE

This is to certify that the work entitled, "AGE PREDICTION USING FACIAL FEATURE EXTRACTION" submitted by SONAL JAIN , ARUNIMA SHARMA, NIHAL BAJWA & NEHA ALREJA in partial fulfillment for the award of degree of Bachelor of Technology in Computer Science /Information Technology of Jaypee University of Information Technology has been carried out under my supervision. This work has not been submitted partially or wholly to any other University or Institute for the award of this or any other degree or diploma.

SATISH CHANDRA
Department of Computer Science and Engineering,
Jaypee University of Information Technology
Waknaghat, Solan – 173215, India

ACKNOWLEDGMENT

Apart from the efforts by us, the success of this project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

We would like to show our greatest appreciation to Mr. Satish Chandra, the project guide. We owe our heartiest thanks to Brig.(Retd.) S.P. Ghrera(Head Of Department –Computer Science Engineering /Information Technology) who has always inspired confidence in us to take initiative. He has always been motivating and encouraging.

We would also like express our sincere gratitude to Mr. D.S. Saini (ECE Department) who helped us solve many of our problems. Finally, thanks to all our family members who supported us in our every grim phase.

Sonal Jain	Roll No. 061273	(Computer Science Engineering)
Arunima Sharma	Roll No. 061410	(Information Technology)
Nihal Bajwa	Roll No. 061431	(Information Technology)
Neha Alreja	Roll No. 061533	(Computer Science Engineering)

B.Tech (Computer Science & Engineering/ Information Technology)

Jaypee University Of Information and Technology

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

MATLAB	MATrix LABoratory
ASHCI	Age-Specific Human Computer Interaction
GUI	Graphical User Interface
SVM	Support Vector Machine
NDG	Network Data Grid
HCI	Human- Computer Interaction
ASM	Active Shape Model
MLP	Multi Layer Perceptron
EUD	Euclidian Distance Maps
UEPs	Ultimate Eroded Points
LDA	Linear Discriminant Analysis
PCA	Principal Components Analysis
CCA	Canonical Correlation Analysis
LAF	Linear Adaptive Filters
KM	Kernel Methods
SAL	Sequence Associations & Link Analysis
QC	Quality Control Technique
SEPATH	Structural Equation Modeling Path Analysis
GLM	General Linear Model
GRM	Generalized Regression Model
PLS	Partial Least Square

ABSTRACT

The aim is to categorize a person as a child, young adult or an elderly person and further to classify them into certain ranges of ages. The input will be a human image and the output will be the predicted age range. The approach includes face detection, feature extraction, calculation of ratios for training and testing.

Features to be extracted can be categorized as primary as well as secondary. Primary feature are eyes, nose, chin and lips and secondary features comprises of skin (wrinkles).

Automatic human age estimation has considerable potential applications in human computer interaction and multimedia communication.

CHAPTER -1

INTRODUCTION

1.1 CHAPTER SUMMARY

Section 1.2 of chapter 1 contains the detailed problem statement of the project and section 1.3 contains the overview of the system to be made.

1.2 PROBLEM STATEMENT

The project is to classify input images into one of three age groups: babies, young adults, and senior adults. The computations are based on cranio-facial development theory and skin wrinkle analysis.

From these features, ratios that distinguish babies from young adults and seniors are computed. In secondary feature analysis, a wrinkle geography map is used to guide the detection and measurement of wrinkles

The wrinkle index computed is sufficient to distinguish seniors from young adults and babies. A combination rule for the ratios and the wrinkle index thus permits categorization of a face into one of three classes.

1.3 OVERVIEW

Human faces convey a significant amount of semantic information for human-to-human communication. People have the ability to accurately recognize and interpret faces in real time.

Facial attributes play a crucial role in applications including multimedia communication and Human- Computer Interaction (HCI). For example, if the user's age can be estimated by a computer, an Age-Specific Human Computer Interaction (ASHCI) system may be developed. Such a system could be used for secure Internet access control in order to ensure that young children do not have access to Internet pages containing adult materials. Or, a vending machine could refuse to sell alcohol or cigarettes to minors. In image and video retrieval, retrieving photographs or videos could be restricted to a required age range.

As a kind of semantic knowledge, the age information could be very helpful for multimedia content analysis and understanding. Image-based automatic age estimation is challenging.

The main difficulty is that people age quite differently. Aging process is determined by not only the person's genes but also many external factors such as health, life style, living location and weather conditions.

The age progression displayed on faces is uncontrollable and personalized. Males and females may also age differently.

CHAPTER -2

TOOLS & TECHNIQUES

2.1 CHAPTER SUMMARY

Section 2.2 of chapter 2 comprises of the image processing techniques and its applications. It also introduces about the techniques of face detection and feature extraction. Section 2.3 contains the detailed description of tools used like MATLAB and STATISTICA (SVM).

2.2 IMAGE PROCESSING

It is a form of signal processing applied on an image such as photographs or frames of video as an input.

The output can either be an image or a set of characteristics related to the image.

Involve treating the image as a 2-dimensional signal and applying standard signal-processing techniques to it.

Usually referred as digital image processing, but optical and analog image processing are also possible.

<u>Applications</u>

- Computer vision
- Optical sorting
- Augmented Reality
- Face detection
- Feature detection
- Lane departure warning system
- Non-photorealistic rendering
- Medical image processing
- Microscope image processing
- Morphological image processing
- Remote sensing

2.2.1 FACE DETECTION

It is a computer technology that determines the locations and sizes of human faces in arbitrary (digital) images. It detects facial features and ignores anything else.

2.2.2 FACIAL FEATURE EXTRACTION:

Facial feature, aims to detect and extract specific feature such as the eyes, lips, chin and nose. It is a challenge for face recognition.

2.3 MATLAB

MATLAB (short for MATrix LABoratory) a special-purpose computer program optimized to perform engineering and scientific calculations. It is a high-performance language for technical computing.

Typical uses include:

- Math and computation
- Algorithm development
- Modeling, simulation and prototyping
- Data analysis, exploration and visualization
- Scientific and engineering graphics
- Application development, including Graphical User Interface (GUI) building

2.3.1 ADVANTAGES OF USING MATLAB

Ease of Use

- MATLAB is an interpreted language. Program may be easily written and modified with the built-in integrated development environment and debugged with the MATLAB debugger.

Platform Independence

- Supported on many different computer systems

Predefined Function

- MATLAB comes complete with an extensive library of predefined functions that provide tested and pre-packaged solutions to many basic technical tasks

2.3.2 MATLAB GRAPHICAL USER INTERFACE

A graphical user interface (GUI) is a graphical display that contains devices or components that enable a user to perform interactive tasks. To perform these tasks, the user of the GUI does not have to create a script or type commands at the command line. Often, the user does not have to know the details of the task at hand.

The GUI components can be menus, toolbars, push buttons, radio buttons, list boxes and sliders—just to name a few. In MATLAB, a GUI can also display data in tabular form or as plots, and can group related components.

The following figure illustrates a simple GUI.

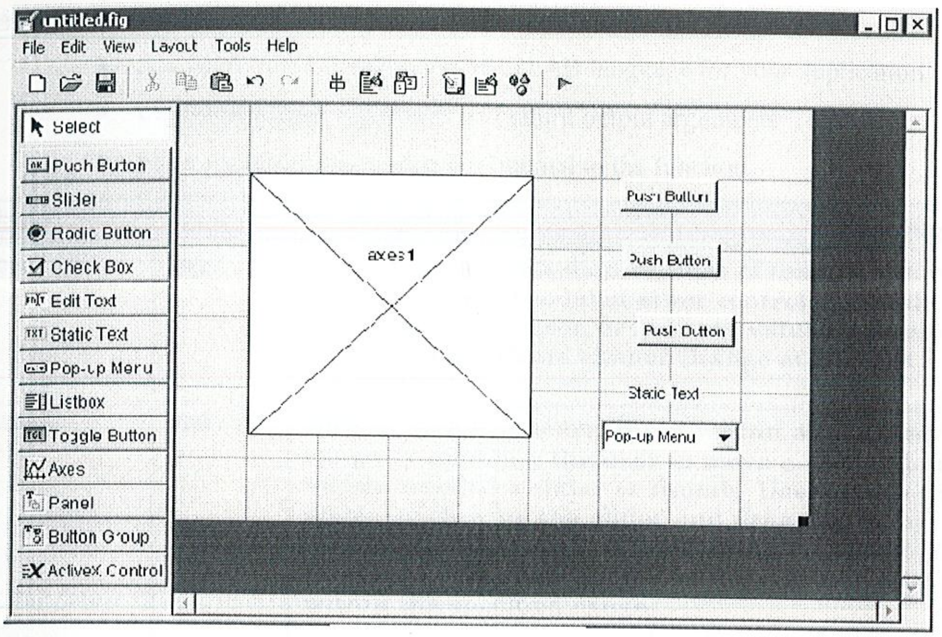


Figure 2.3.2 (i)

Laying Out a GUI

The GUIDE Layout Editor enables you to populate a GUI by clicking and dragging GUI components, such as buttons, text fields, sliders, axes, and soon, into the layout area. It also enables you to create menus and context menus for the GUI.Other tools, which are accessible from the Layout Editor, enable you to size the GUI, modify component look and feel, align components, set tab order, view a hierarchical list of the component objects, and set GUI options.

Programming a GUI

When you save your GUI layout, GUIDE automatically generates an M-file that you can use to control how the GUI works. This M-file provides code to initialize the GUI and contains a framework for the GUI callbacks—the routines that execute in response to user-generated events such as a mouse click. Using the M-file editor, you can add code to the callbacks to perform the functions you want.

Working with M-Files

M-files can be *scripts* that simply execute a series of MATLAB statements, or they can be *functions* that also accept input arguments and produce output

MATLAB functions:

- Are useful for extending the MATLAB language for your application
- Can accept input arguments and return output arguments
- Store variables in a workspace internal to the function.

Component	Icon	Description
Static Text	TXT	Static text controls display lines of text. Static text is typically used to label other controls, provide directions to the user, or indicate values associated with a slider. Users cannot change static text interactively.
Slider		Sliders accept numeric input within a specified range by enabling the user to move a sliding bar, which is called a slider or thumb. Users move the slider by clicking the slider and dragging it, by clicking in the trough, or by clicking an arrow. The location of the slider indicates the relative location within the specified range.
List Box		List boxes display a list of items and enable users to select one or more items.
Pop-Up Menu		Pop-up menus open to display a list of choices when users click the arrow.
Axes		Axes enable your GUI to display graphics such as graphs and images. Like all graphics objects, axes have properties that you can set to control many aspects of its behavior and appearance. See "Axes Properties" in the MATLAB Graphics documentation and commands such as the following for more information on axes objects: `plot`, `surf`, `line`, `bar`, `polar`, `pie`, `contour`, and `mesh`. See Functions — By Category in the MATLAB Function Reference documentation for a complete list.

Figure 2.3.2(ii)

2.4 STATISTICA

Statistica is the tool of making effective use of numerical data relating to groups of individuals or experiments. It deals with all aspects of this, including not only the collection, analysis and interpretation of such data, but also the prediction of data, in terms of the design of surveys and experiments.

Statistica is a statistics and analytics software package which provides data analysis, data management, data mining, and data visualization procedures.

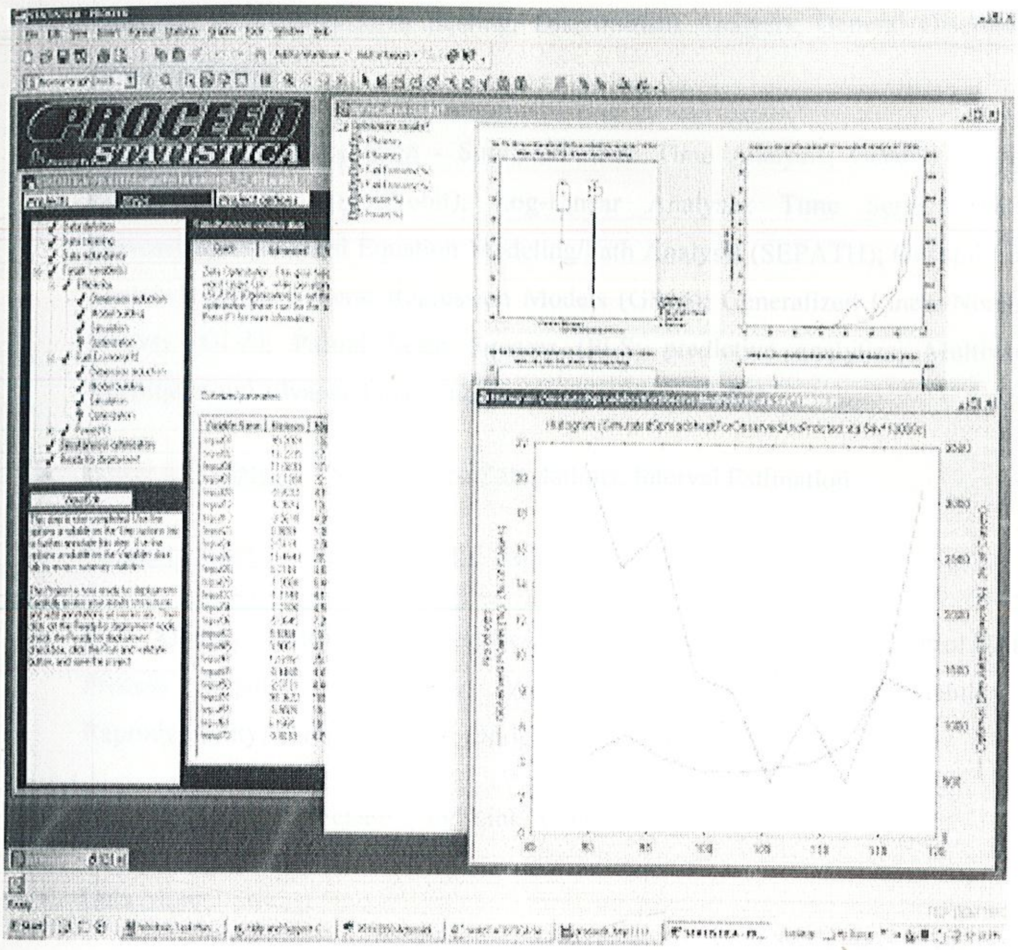


Figure 2.4(i)

It provides different packages of analytical techniques:

- Base - The base version of the software includes core statistical functions: Descriptive Statistics, Breakdowns, and Exploratory Data Analysis; Correlations; Probability Calculator; T-Tests (and other tests of group differences); Frequency Tables, Crosstabulation Tables, Stub-and-Banner Tables, Multiple Response Analysis; Multiple Regression Methods; Nonparametric Statistics; Distribution Fitting

Additional modules are available for advanced features:

- Cluster Analysis Techniques - Factor Analysis and Principle Components; Canonical Correlation Analysis; Reliability/Item Analysis; Classification Trees; Correspondence

Analysis; Multidimensional Scaling; Discriminant Analysis; General Discriminant Analysis Models

- <u>Distribution and Simulation</u> - Survival/Failure Time Analysis; General Nonlinear Estimation (and Logit/Probit); Log-Linear Analysis; Time Series Analysis, Forecasting; Structural Equation Modeling/Path Analysis (SEPATH); General Linear Models (GLM); General Regression Models (GRM); Generalized Linear/Nonlinear Models (GLZ); Partial Least Squares (PLS; predictive analytics; Multivariate techniques and advanced linear/nonlinear regression models.

- <u>Power Calculations</u> - Sample Size Calculations; Interval Estimation

- <u>Neural networks</u> - neural networks; training algorithms

- <u>QC</u> - Quality control techniques; design of experiments; Quality Control Charts; Process Capability Analysis; Weibull Analysis; Gage Repeatability & Reproducibility; Sampling Plans; Design of Experiments; Taguchi Designs

- <u>SAL</u>- Sequence, Association, and Link Analysis

- <u>Data Miner</u>

2.4.1 SUPPORT VECTOR MACHINE (SVM)

Support vector machines are a set of related supervised learning methods used for classification and regression. In simple words, given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that predicts whether a new example falls into one category or the other.

Intuitively, an SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on.

More formally, it constructs a hyperplane or set of hyperplanes in a high or infinite dimensional space, which can be used for classification, regression or other tasks. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest training datapoints of any class (so-called functional margin), since in general the larger the margin the lower the generalization error of the classifier

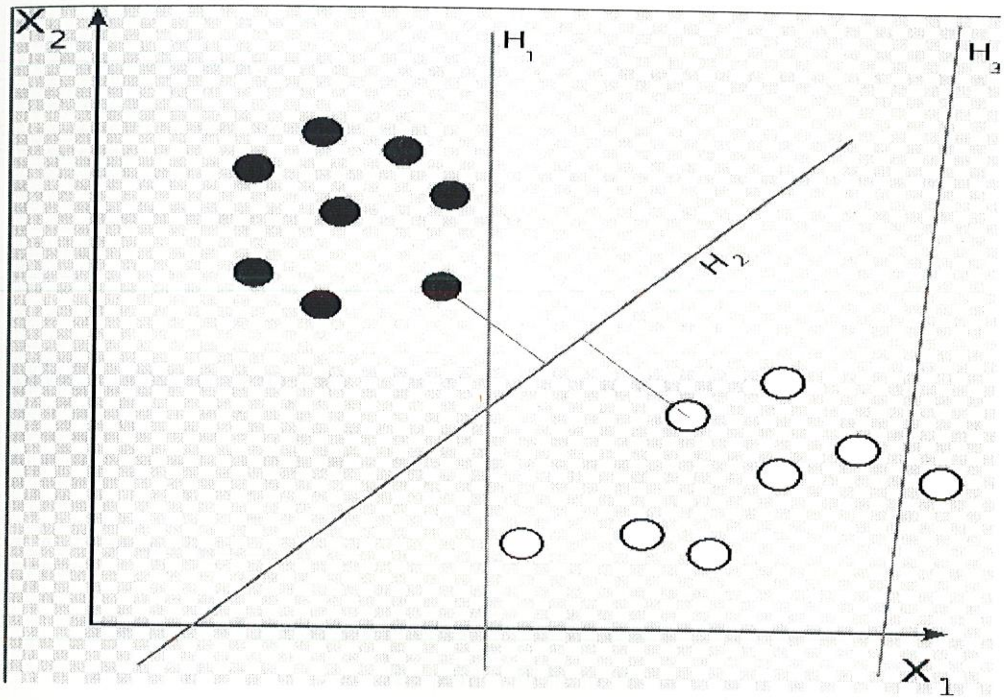


Figure 2.4.1(i)

- H3 (green) doesn't separate the 2 classes.
- H1 (blue) does, with a small margin.
- H2 (red) with the maximum margin.

2.4.2 ADVANTAGES OF USING SVM

- Excellent results on practical learning benchmarks were achieved in digit recognition, computer vision and text categorization. Today, SVMs show better results than (or comparable outcomes to) NNs and other statistical models, on the most popular benchmark problems.
- More accurate and less time complexity.
- SVM Light: SVM^{light} is an implementation of Support Vector Machines (SVMs) for the problem of pattern recognition, for the problem of regression, and for the problem of learning a ranking function.
- The algorithm has scalable memory requirements and can handle problems with many thousands of support vectors efficiently.

2.4.3 CLASSIFICATION BY SUPPORT VECTOR MACHINE

- Linear classifier :

  Suppose some given data points each belong to one of two classes, and the goal is to decide which class a *new* data point will be in. In the case of support vector machines, a data point is viewed as a p-dimensional vector (a list of p numbers), and we want to know whether we can separate such points with a $p-1$-dimensional hyperplane. This is called a linear classifier.

- Maximum margin classifier :

  There are many hyperplanes that might classify the data. However, maximum separation (margin) between the two classes is usually desired. So we choose the hyperplane so that the distance from it to the nearest data point on each side is maximized. If such a hyperplane exists, it is clearly of interest and is known as the *maximum-margin hyperplane* and such a linear classifier is known as a Maximum margin classifier.

2.4.4 KERNELS IN SVM

Kernel methods (KMs) are a class of algorithms for pattern analysis, whose best known element is the support vector machine (SVM). The general task of pattern analysis is to find and study general types of relations (for example clusters, rankings, principal components, correlations, classifications) in general types of data (such as sequences, text documents, sets of points, vectors, images, etc).

KMs approach the problem by mapping the data into a high dimensional feature space, where each coordinate corresponds to one feature of the data items, transforming the data into a set of points in a Euclidean space. In that space, a variety of methods can be used to find relations in the data. Since the mapping can be quite general (not necessarily linear, for example), the relations found in this way are accordingly very general. This approach is called the kernel trick.

KMs owe their name to the use of kernel functions, that enable them to operate in the feature space without ever computing the coordinates of the data in that space, but rather by simply computing the inner products between the images of all pairs of data in the feature space. This operation is often computationally cheaper than the explicit computation of the coordinates. Kernel functions have been introduced for sequence data, graphs, text, images, as well as vectors.

Algorithms capable of operating with kernels include SVM, Gaussian processes, Fisher's linear discriminant analysis (LDA), principal components analysis (PCA), canonical correlation analysis, ridge regression, spectral clustering, linear adaptive filters and many others.

Because of the particular culture of the research community that has been developing this approach since the mid-1990s, most kernel algorithms are based on convex optimization or eigenproblems, are computationally efficient and statistically well-founded. Typically, their statistical properties are analyzed using statistical learning theory (for example, using Rademacher complexity).

2.4.4.1 RADIAL BASIS FUNCTION

Normally a Gaussian will be used as the RBF, the output of the kernel is dependent on the Euclidean distance from support vector from the testing data point. The support vector will be the centre of the RBF and will determine the area of influence this support vector has over the data space. The following figure shows a two-dimensional version of such a kernel.

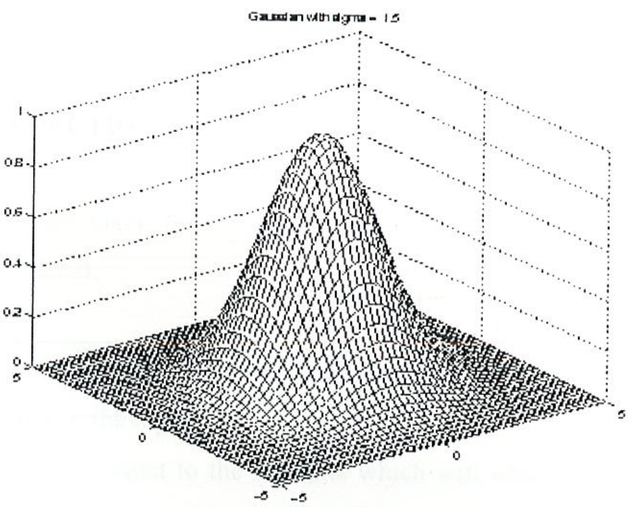


Figure: The Radius Basis Function kernel

Figure 2.4.4.1 (i)

CHAPTER -3

FEATURE DETECTION & EXTRACTION

3.1 CHAPTER SUMMARY

Section 3.2 of chapter 3 shows the database collection of the images that can be used for face detection and feature extraction. Section 3.3 contains the detailed face detection technique and how it is implemented. It also consists of the MATLAB GUI snapshots. Section 3.4 contains the detailed description of feature extraction technique, also showing the example of extracted primary features like eyes, nose , mouth , chin by MATLAB GUI snapshots.

3.2 DATABASE COLLECTION:

Initially the images are taken from the Database and their face and facial features are detected using MATLAB.

- There are six ratios that are calculated using the facial features which are demonstrated later in the report.
- These ratios act as an input to the machine which will observe the pattern from the figures of the training data.
- Training data is chosen arbitrarily and using support vector machine our machine is trained.
- Observations are made. Rest of the data is used for testing and thus the classification is predicted.

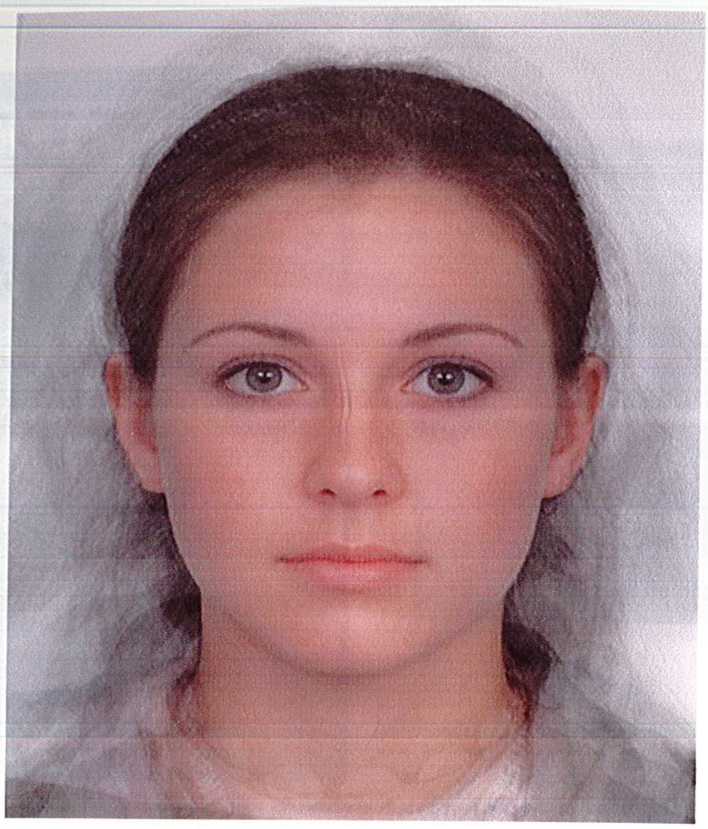
IMAGE DATABASE

Figure 3.2(i)

3.3 FACE DETECTION

Face detection is a computer technology that determines the locations and sizes of human faces in arbitrary (digital) images. It detects facial features and ignores anything else, such as buildings, trees and bodies.

Face detection can be regarded as a more general case of face localization; In face localization, the task is to find the locations and sizes of a known number of faces (usually one). In face detection, one does not have this additional information.



Original Image

Figure 3.3(i)

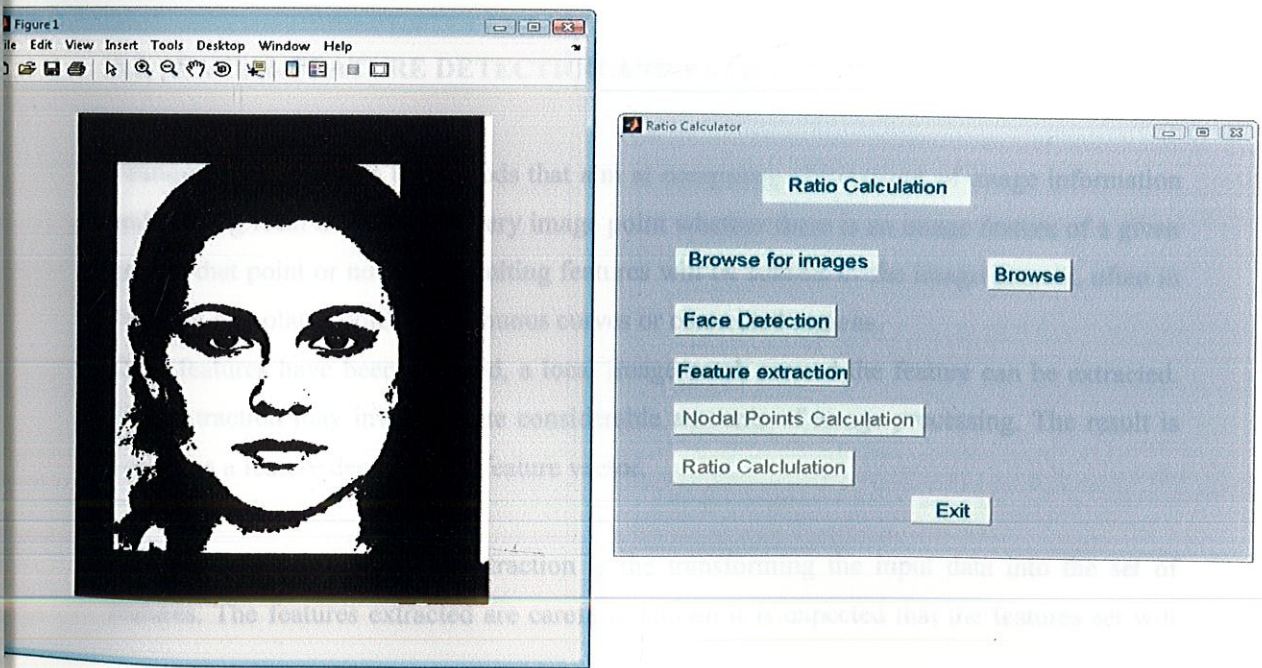
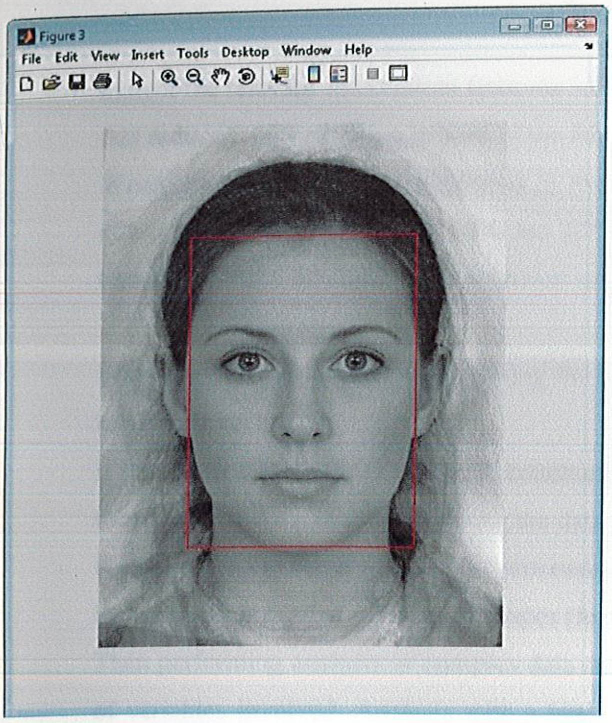


Figure 3.3(ii)



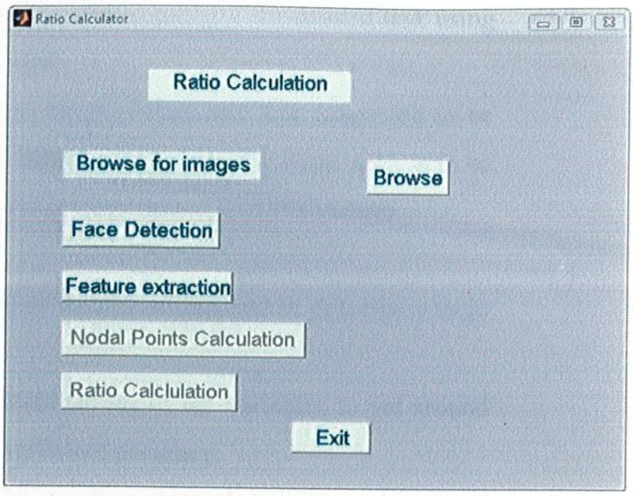


Image showing detected face

Figure 3.3(iii)

3.4 FACIAL FEATURE DETECTION AND EXTRACTION

Feature detection refers to methods that aim at computing abstractions of image information and making local decisions at every image point whether there is an image feature of a given type at that point or not. The resulting features will be subsets of the image domain, often in the form of isolated points, continuous curves or connected regions.

Once features have been detected, a local image patch around the feature can be extracted. This extraction may involve quite considerable amounts of image processing. The result is known as a feature descriptor or feature vector.

To be more precise, Feature extraction is the transforming the input data into the set of features. The features extracted are carefully chosen it is expected that the features set will extract the relevant information from the input data in order to perform the desired task using this reduced representation instead of the full size input.

When the input data to an algorithm is too large to be processed and it is suspected to be notoriously redundant (much data, but not much information) then the input data will be transformed into a reduced representation set of features (also named features vector).

Feature extraction involves simplifying the amount of resources required to describe a large set of data accurately.

It is a general term for methods of constructing combinations of the variables to get around these problems while still describing the data with sufficient accuracy.

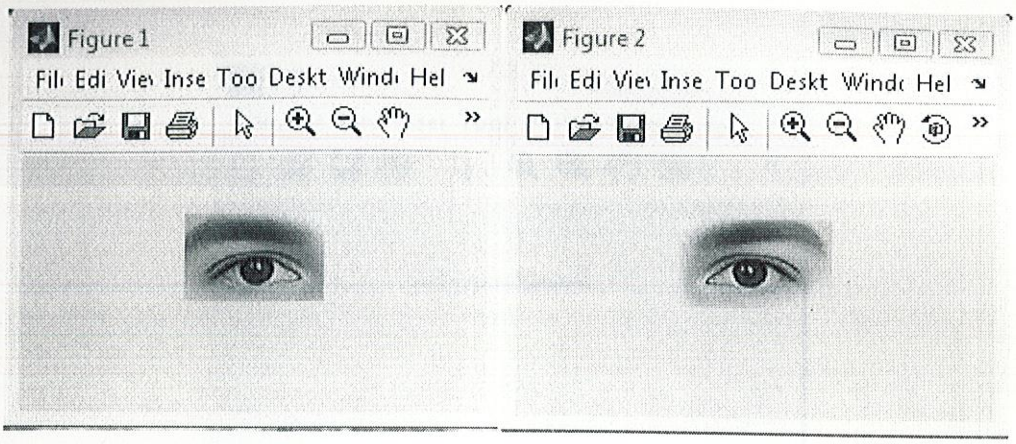
It can be used in the area of image processing which involves using algorithms to detect and isolate various desired portions or shapes (features) of a digitized image.

Then performing analysis of complex data one of the major problems stems from the number of variables involved. Analysis with a large number of variables generally requires a large amount of memory and computation power or a classification algorithm which overfits the training sample and generalizes poorly to new samples.

3.4.1 EXTRACTED FEATURES

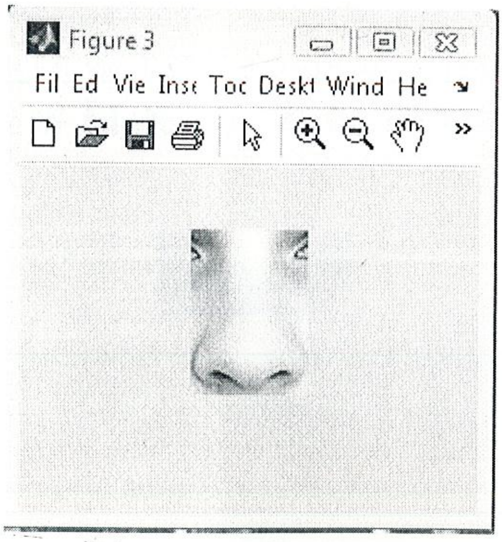
Primary Features:

- Left Eye
- Right Eye
- Nose
- Chin
- Lips



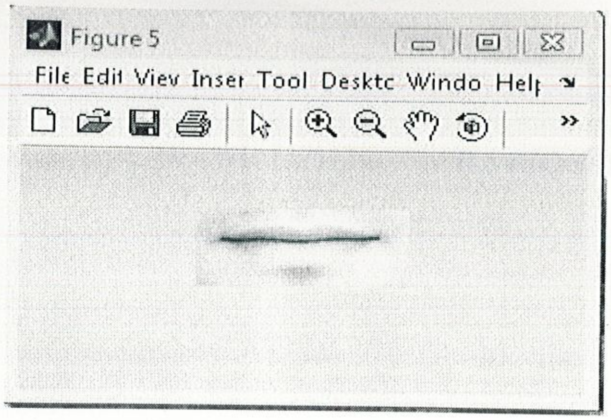
LEFT EYE **RIGHT EYE**

Figure 3.4.1(i)



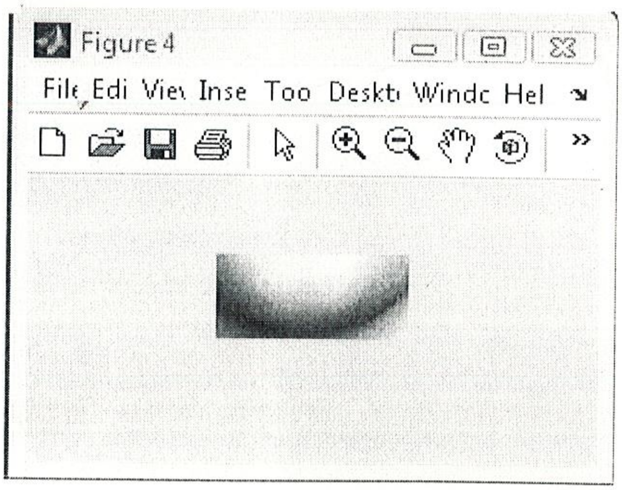
NOSE

Figure 3.4.1(ii)



LIPS

Figure 3.4.1(iii)



CHIN

Figure 3.4.1(iv)

CHAPTER -4

RATIO CALCULATION

4.1 CHAPTER SUMMARY

Section 4.2 contains the detailed description of nodal points computation. It also contains all the MATLAB GUI snapshots of the masks and the correlated images of the primary features. Section 4.3 shows the geometry of face and defines the distances considered to calculate the ratios.

Section 4.4 shows the 5 important ratios calculated and their significance. It also contains the MATLAB GUI snapshots of the same.

4.2 NODAL POINTS CALCULATION:

- Extracted the primary features
- Created masks for all the features.
- Comparison done with the original ones.
- Correlation between the two images was computed.
- From the correlation image the top, bottom, left and right co-ordinates were calculated.
- The centre point was computed for the left eye, right eye, mouth and bottom point for nose and chin using the above co-ordinates.

Mask

A mask is a custom user interface for a subsystem that hides the subsystem's contents, making it appear to the user as an atomic block with its own icon and parameter dialog box.

Correlation

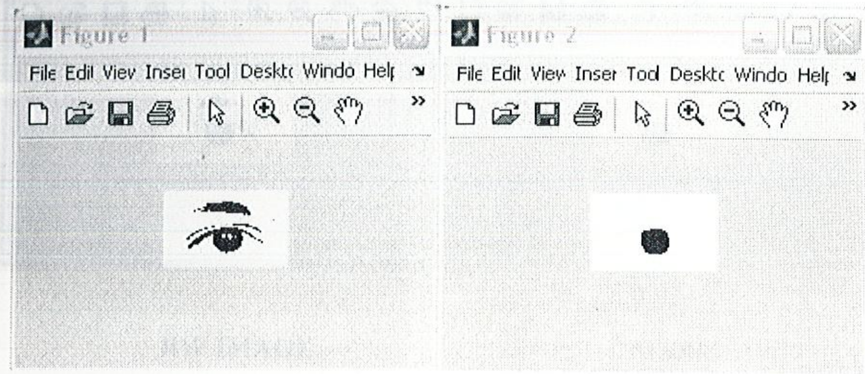
The operation called correlation is closely related to convolution. In correlation, the value of an output pixel is also computed as a weighted sum of neighboring pixels. The difference is that the matrix of weights, in this case called the correlation kernel, is not rotated during the computation.

It produces a matrix of correlation coefficients for a data matrix where each column represents a variable. The correlation coefficients are a normalized measure of the strength of the linear relationship between two variables and range between -1 and 1, where:

- -1 means that one column of data has a negative linear relationship to another column of data.
- 0 means there is no linear relationship between the data columns.
- 1 means that there is a positive linear relationship between the data columns.

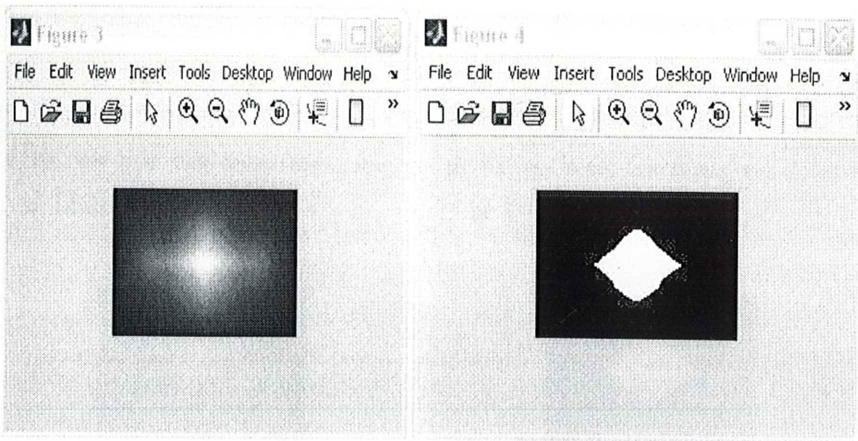
4.2.1 NODAL POINTS GUI

LEFT EYE:



BW IMAGE **MASK**

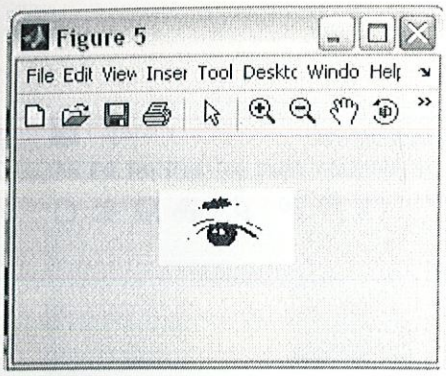
Figure 4.2.1(i)



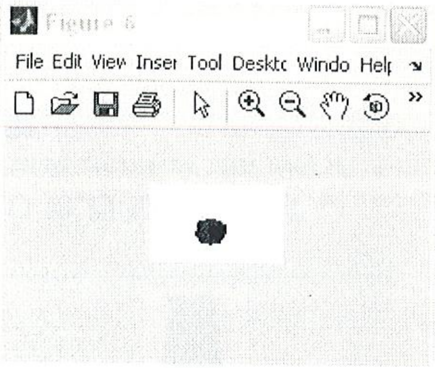
CORRELATION **BW IMAGE OF CORRELATION**

Figure 4.2.1(ii)

RIGHT EYE:



BW IMAGE



MASK

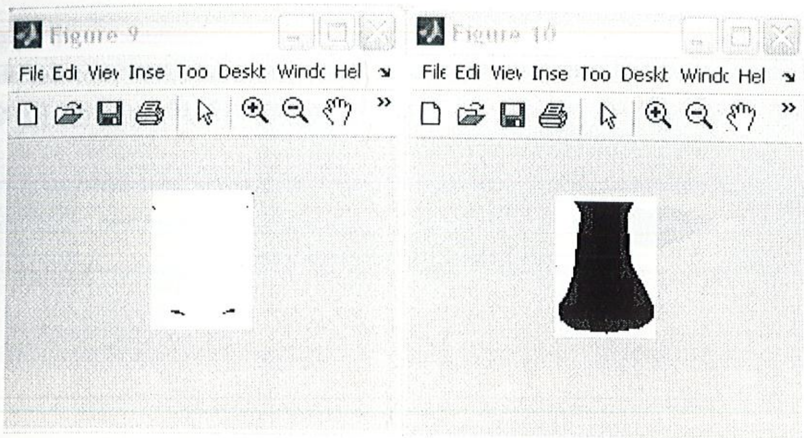
Figure 4.2.1(iii)

CORRELATION

BW IMAGE OF CORRELATION

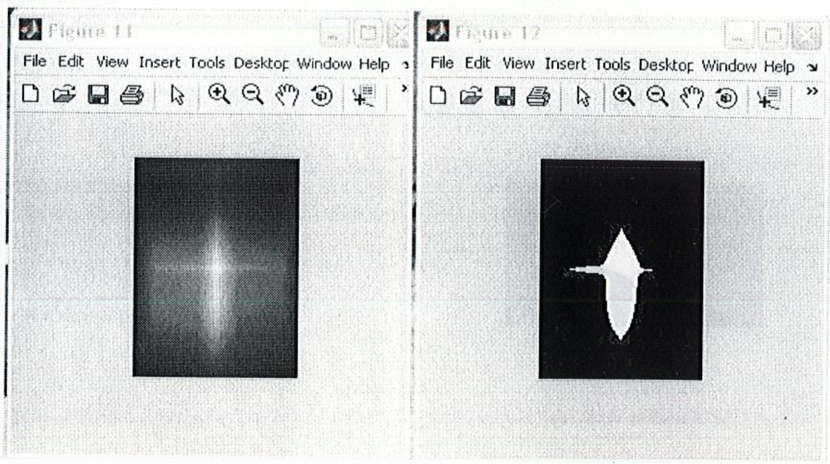
Figure 4.2.1(iv)

NOSE:



BW IMAGE **MASK**

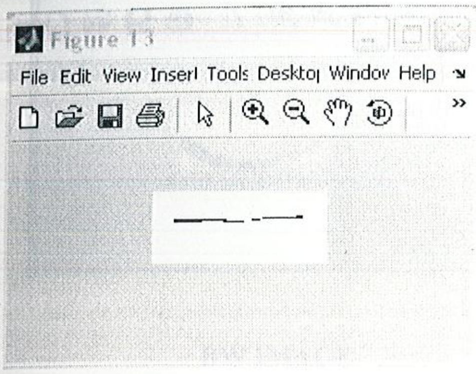
Figure 4.2.1(v)

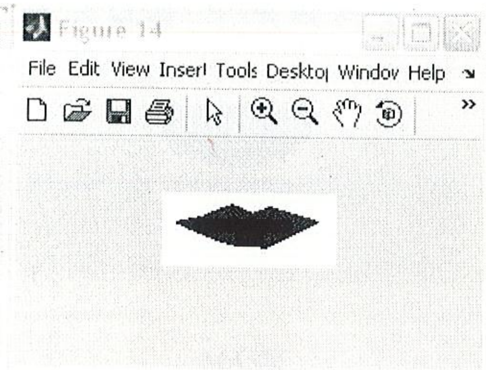


CORRELATION **BW IMAGE OF CORRELATION**

Figure 4.2.1(vi)

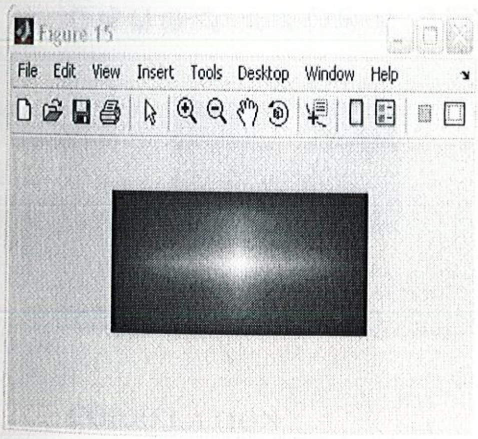
LIPS:

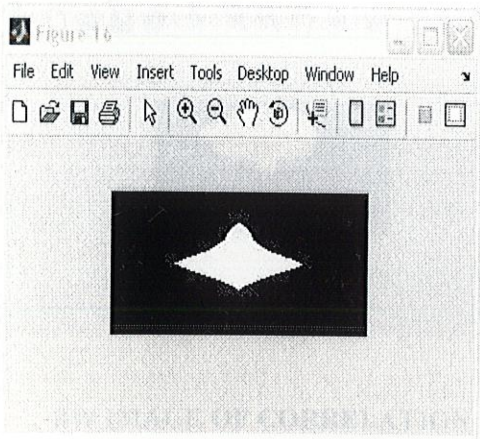




BW IMAGE **MASK**

Figure 4.2.1(vii)

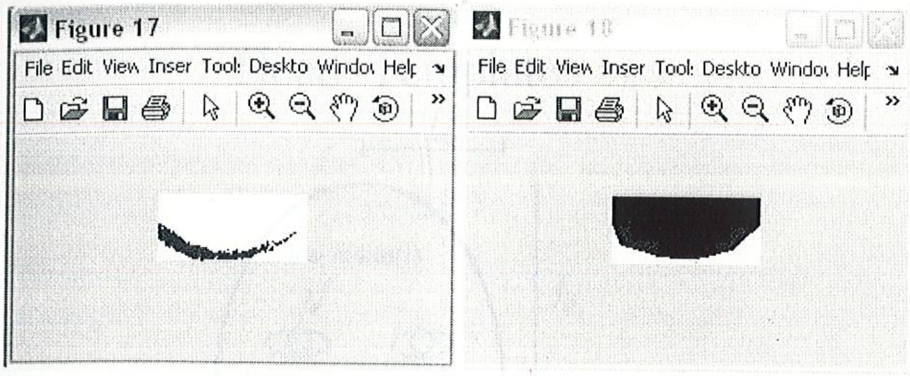




CORRELATION **BW IMAGEOF CORRELATION**

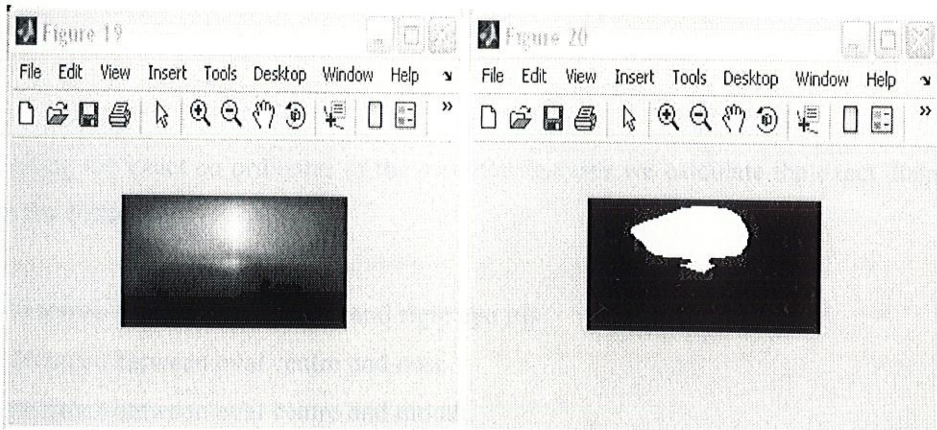
Figure 4.2.1(viii)

CHIN:



BW IMAGE **MASK**

Figure 4.2.1(ix)



CORRELATION **BW IMAGE OF CORRELATION**

Figure 4.2.1(x)

4.3 DISTANCE CALCULATION

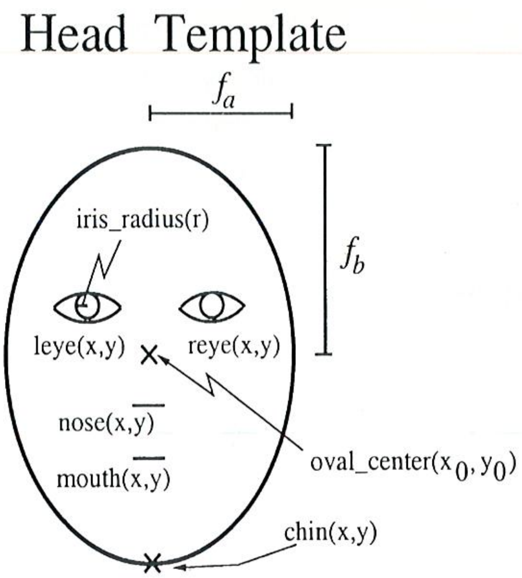


Figure 4.3(i)

After finding the exact co ordinates of the essential features we calculate the exact distances between the following:

1. Distance between left eye iris and right eye iris
2. Distance between oval centre and nose
3. Distance between oval centre and mouth
4. Distance between oval centre and chin

4.3.1 RATIOS

After calculating these distances we calculated the following ratios:

ratio1 = distance between right iris-left iris upon distance between center-nose

ratio2 = between right iris-left iris and center-mouth

ratio3 = between right iris-left iris and center-chin

ratio4 = between center-nose and center-mouth

ratio5 = between center-mouth and center-chin

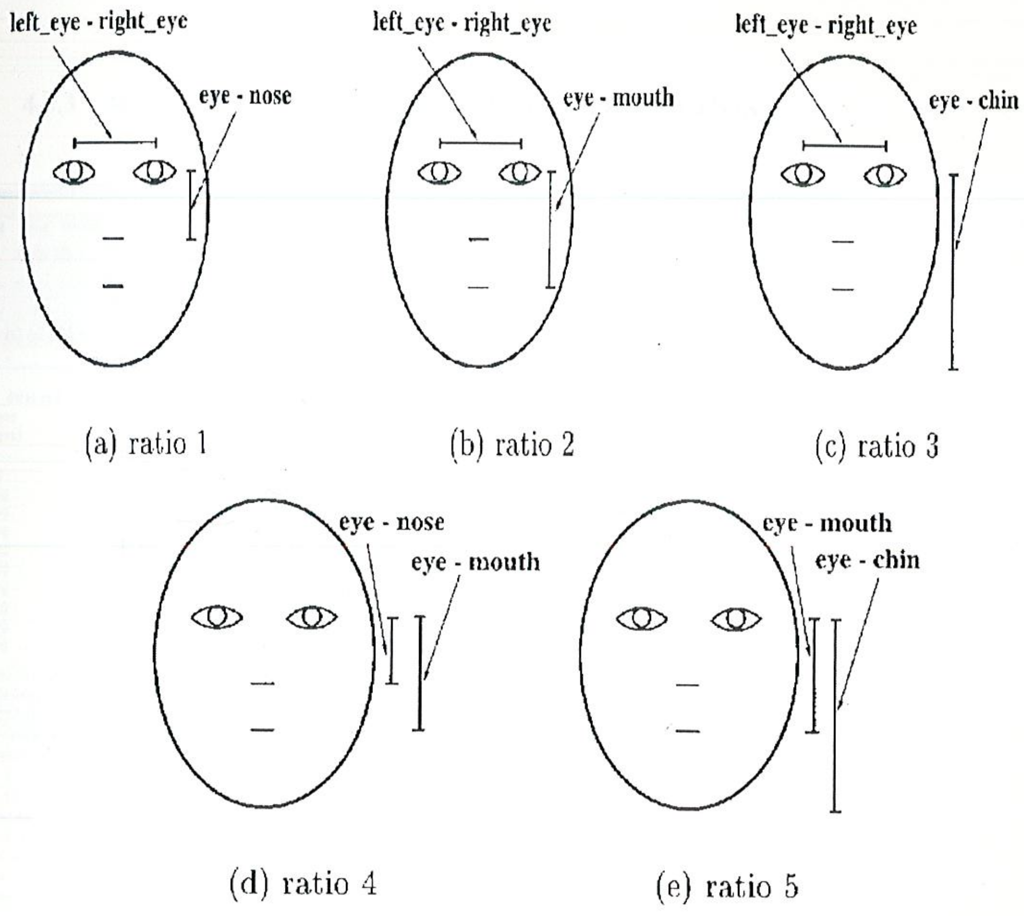


Figure 4.3.1(i)

4.3.2 SIGNIFICANCE OF EACH RATIO

- The most promising is Ratio 1. This ratio uses features which are not affected by any facial expressions or facial motions. However, it too is subject to imprecise localization. If it can be made robust to shading, shadowing, and occlusion effects, it should serve as a good classifier.
- Ratio 2 appears to be the ratio that can be measured reliably and also shows promise in providing reliable classification.
- Ratios 3, 4, and 5 are not as promising.

4.3.3 MATLAB GUI SHOWING CALCULATED RATIOS

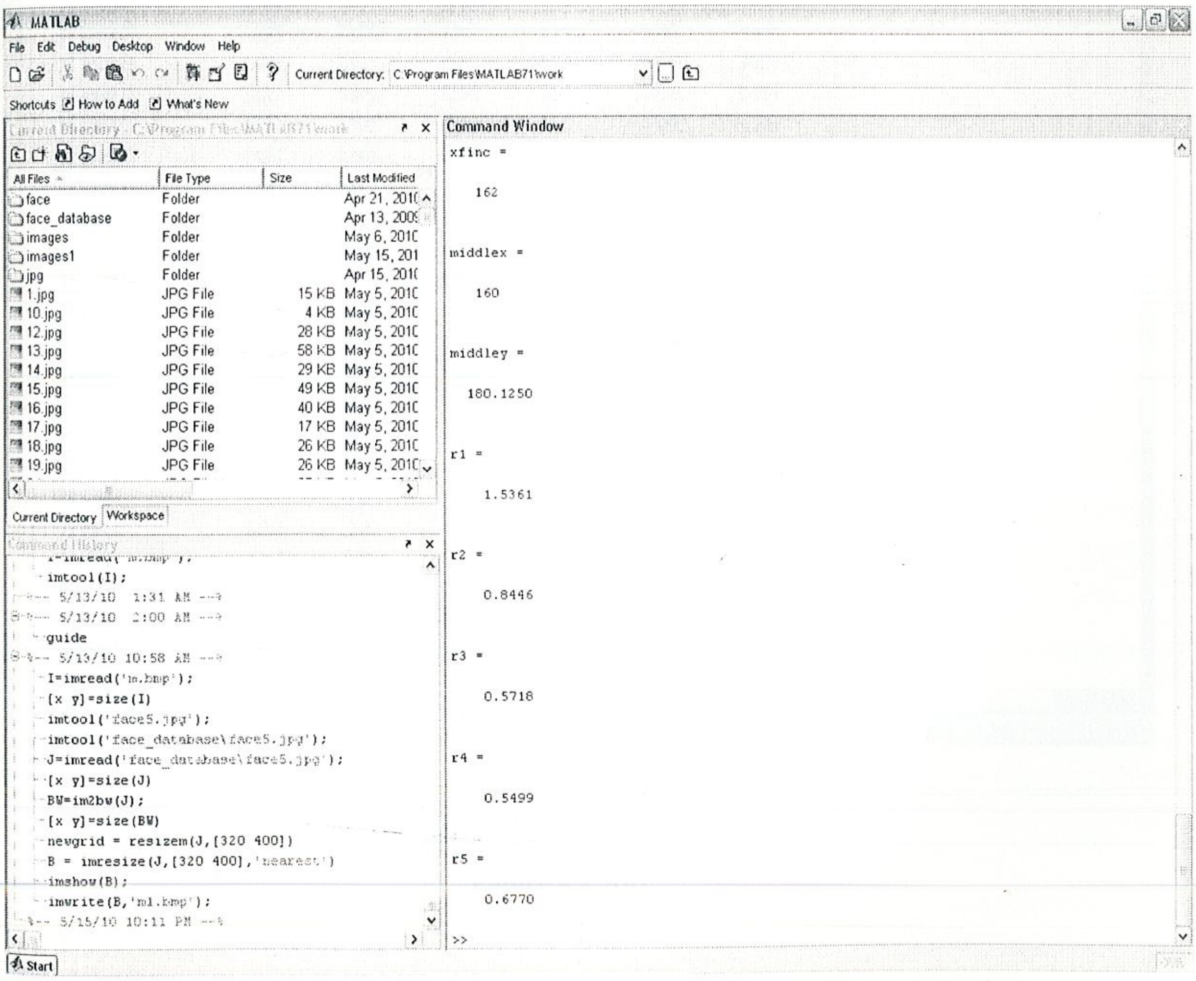


Figure 4.3.3(i)

	subject	ratio 1(E-N)	ratio 2(E-M)	ratio 3(E-C)	ratio 4(N-M)	ratio 5(M-C)	ratio 6(H-C)
2	young 1	1.614	0.9583	0.5786	0.5938	0.6038	0.5867
3	young 2	1.5188	0.8898	0.5506	0.5859	0.6188	0.613
4	young 3	1.5328	0.9154	0.5559	0.5493	0.6073	0.6082
5	young 4	1.4777	0.8747	0.5737	0.5362	0.6121	0.5611
6	young 5	1.5263	0.8863	0.5523	0.4552	0.6221	0.5889
7	young 6	1.5469	0.7941	0.5467	0.5231	0.6012	0.5678
8	young 7	1.4993	0.9477	0.5808	0.5112	0.6128	0.6138
9	young 8	1.4589	0.9723	0.5827	0.5194	0.5994	0.5663
10	young 9	1.5849	0.8863	0.4985	0.4287	0.5919	0.5863
11	young 10	1.5346	0.9326	0.5362	0.5325	0.5898	0.5496
12	young 11	1.4859	0.8825	0.4551	0.5854	0.6112	0.5875
13	young 12	1.5782	0.8621	0.5562	0.5421	0.5988	0.5692
14	young 13	1.6152	0.9582	0.5912	0.5226	0.6015	0.5986
15	young 14	1.5263	0.8956	0.5336	0.5361	0.611	0.5826
16	young 15	1.4263	0.8912	0.5489	0.5746	0.6025	0.5967
17	young 16	1.5236	0.9825	0.5366	0.5435	0.6171	0.6015
18	young 17	1.5426	0.9845	0.5933	0.5624	0.6252	0.5996
19	young 18	1.5289	0.8756	0.4826	0.4486	0.6042	0.5863
20	young 19	1.4286	0.8881	0.5625	0.4552	0.6155	0.6013
21	young 20	1.6132	0.8963	0.5362	0.5532	0.6023	0.5933
22	child 1	1.6208	0.991	0.6054	0.6115	0.6109	0.6155
23	child 2	1.7939	1.0502	0.6196	0.5854	0.6585	0.6252
24	child 3	1.6723	0.9893	0.6252	0.6159	0.6325	0.6082
25	child 4	1.6658	0.9753	0.6015	0.6051	0.6253	0.6138
26	child 5	1.6952	1.026	0.6013	0.5826	0.6425	0.5919
27	child 6	1.6875	1.1152	0.6138	0.6133	0.528	0.6196
28	child 7	1.7852	0.9852	0.5863	0.6013	0.6592	0.5912
29	child 8	1.6583	0.9778	0.5898	0.5994	0.6342	0.6025
30	child 9	1.7259	1.102	0.6155	0.5663	0.6574	0.6121
31	child 10	1.7458	1.1132	0.6082	0.6082	0.6354	0.6129
32	child 11	1.6359	1.0563	0.5779	0.5994	0.6551	0.6133

Excel sheet containing the ratios

Figure 4.3.3(ii)

CHAPTER 5

TRAINING AND TESTING OF CALCULATED RATIOS

5.1 CHAPTER SUMMARY

Section 5.2 details the machine learning and its classification , kernels etc. Section 5.3 contains the various snapshots of SVM training and testing of data.

5.2 MACHINE LEARNING

It is a scientific discipline that is concerned with the design and development of algorithms that allow computers to learn based on data, such as from sensor data or databases. A major focus of machine learning research is to automatically learn to recognize complex patterns and make intelligent decisions based on data.

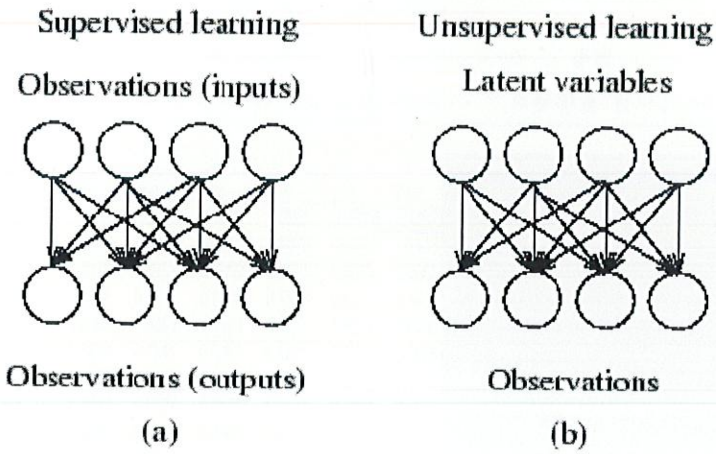
It can be categorized as

- Supervised Learning
- Unsupervised Learning

In supervised learning, the model defines the effect one set of observations, called inputs, has on another set of observations, called outputs. In other words, the inputs are assumed to be at the beginning and outputs at the end of the causal chain. The models can include mediating variables between the inputs and outputs.

In unsupervised learning, all the observations are assumed to be caused by latent variables, that is, the observations are assumed to be at the end of the causal chain. In practice, models for supervised learning often leave the probability for inputs undefined. This model is not needed as long as the inputs are available, but if some of the input values are missing, it is not possible to infer anything about the outputs.

If the inputs are also modeled, then missing inputs cause no problem since they can be considered latent variables as in unsupervised learning.

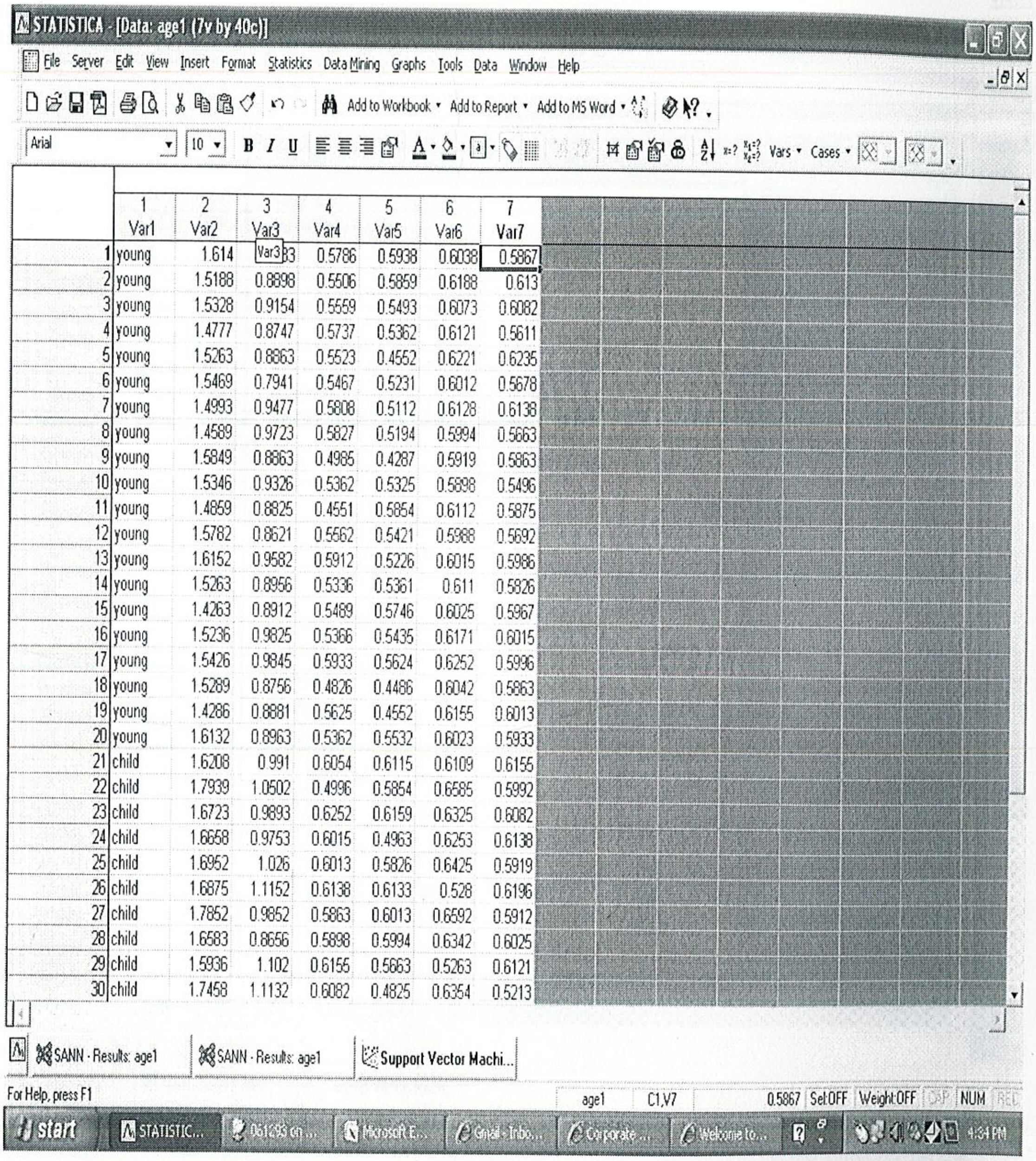


The causal structure of (a) supervised and (b) unsupervised learning

Figure 5.2(i)

In supervised learning, one set of observations, called inputs, is assumed to be the cause of another set of observations, called outputs, while in unsupervised learning all observations are assumed to be caused by a set of latent variables.

5.3 SVM SNAPSHOTS FOR TRAINING AND TESTING



	1 Var1	2 Var2	3 Var3	4 Var4	5 Var5	6 Var6	7 Var7
1	young	1.614	[illegible]	0.5786	0.5938	0.6038	0.5867
2	young	1.5188	0.8898	0.5506	0.5859	0.6188	0.613
3	young	1.5328	0.9154	0.5559	0.5493	0.6073	0.6082
4	young	1.4777	0.8747	0.5737	0.5362	0.6121	0.5611
5	young	1.5263	0.8863	0.5523	0.4552	0.6221	0.6235
6	young	1.5469	0.7941	0.5467	0.5231	0.6012	0.5678
7	young	1.4993	0.9477	0.5808	0.5112	0.6128	0.6138
8	young	1.4589	0.9723	0.5827	0.5194	0.5994	0.5663
9	young	1.5849	0.8863	0.4985	0.4287	0.5919	0.5863
10	young	1.5346	0.9326	0.5362	0.5325	0.5898	0.5496
11	young	1.4859	0.8825	0.4551	0.5854	0.6112	0.5875
12	young	1.5782	0.8621	0.5562	0.5421	0.5988	0.5692
13	young	1.6152	0.9582	0.5912	0.5226	0.6015	0.5986
14	young	1.5263	0.8956	0.5336	0.5361	0.611	0.5826
15	young	1.4263	0.8912	0.5489	0.5746	0.6025	0.5967
16	young	1.5236	0.9825	0.5366	0.5435	0.6171	0.6015
17	young	1.5426	0.9845	0.5933	0.5624	0.6252	0.5996
18	young	1.5289	0.8756	0.4826	0.4486	0.6042	0.5863
19	young	1.4286	0.8881	0.5625	0.4552	0.6155	0.6013
20	young	1.6132	0.8963	0.5362	0.5532	0.6023	0.5933
21	child	1.6208	0.991	0.6054	0.6115	0.6109	0.6155
22	child	1.7939	1.0502	0.4996	0.5854	0.6585	0.5992
23	child	1.6723	0.9893	0.6252	0.6159	0.6325	0.6082
24	child	1.6658	0.9753	0.6015	0.4963	0.6253	0.6138
25	child	1.6952	1.026	0.6013	0.5826	0.6425	0.5919
26	child	1.6875	1.1152	0.6138	0.6133	0.528	0.6196
27	child	1.7852	0.9852	0.5863	0.6013	0.6592	0.5912
28	child	1.6583	0.8656	0.5898	0.5994	0.6342	0.6025
29	child	1.5936	1.102	0.6155	0.5663	0.5263	0.6121
30	child	1.7458	1.1132	0.6082	0.4825	0.6354	0.5213

Figure 5.3(i)

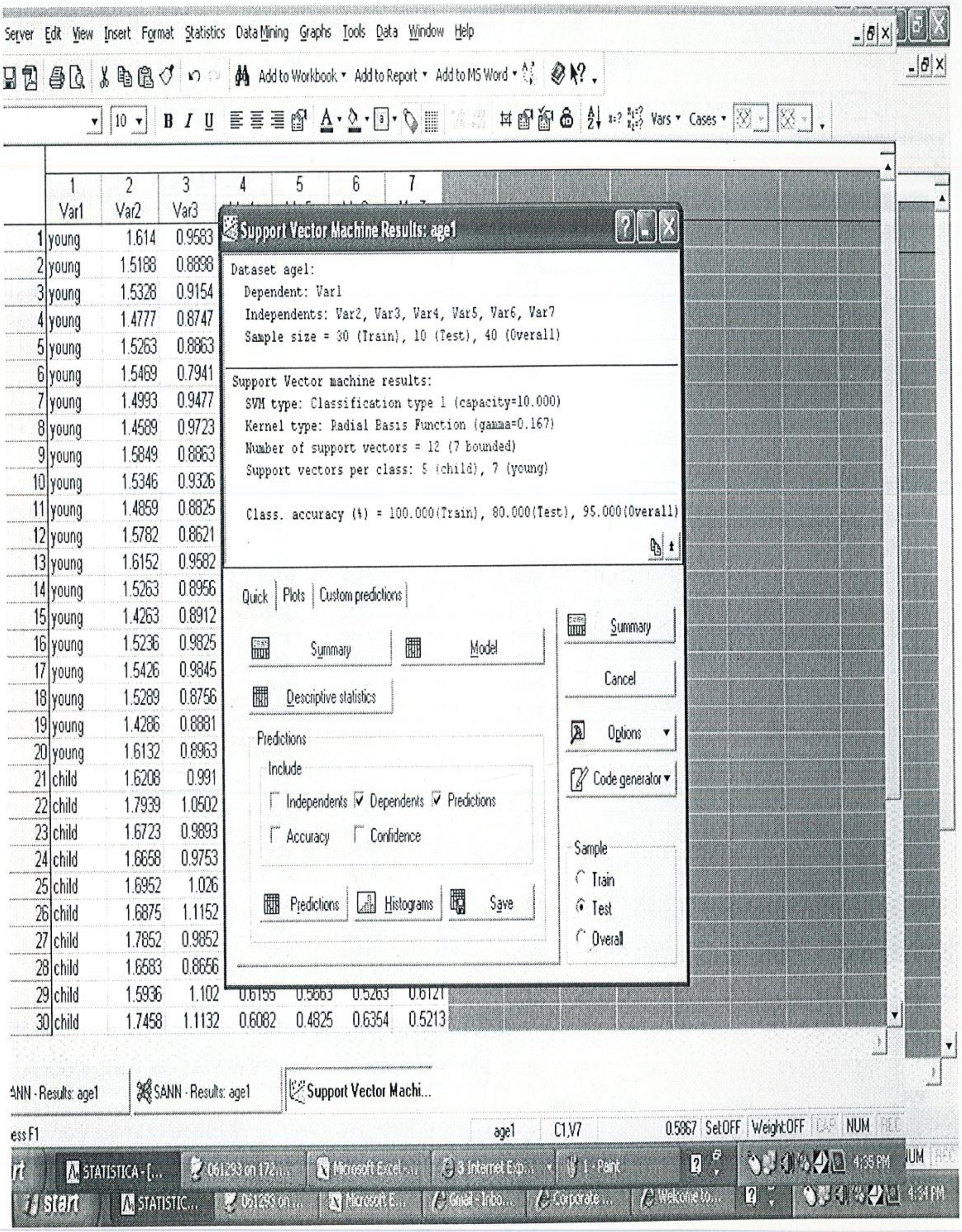
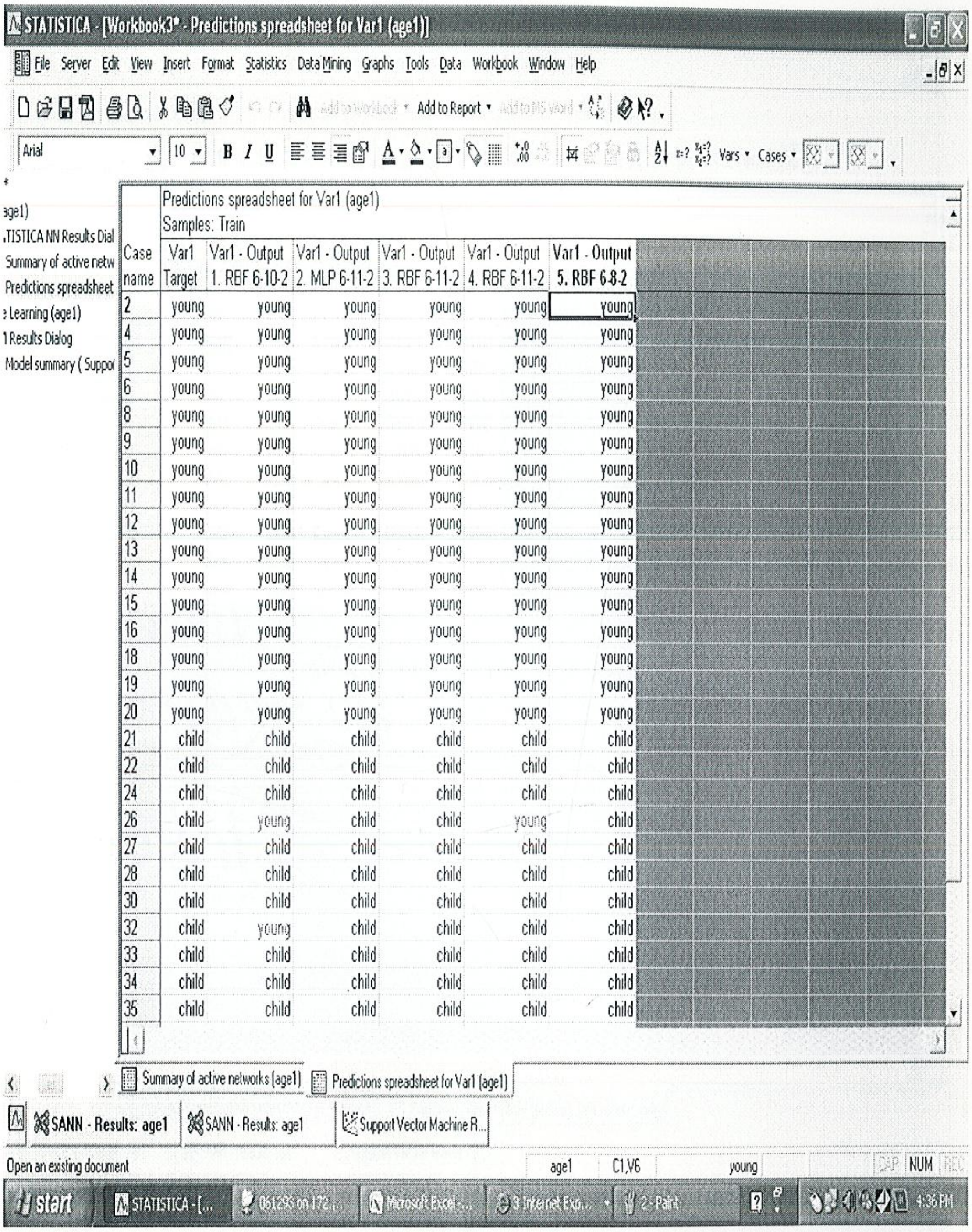
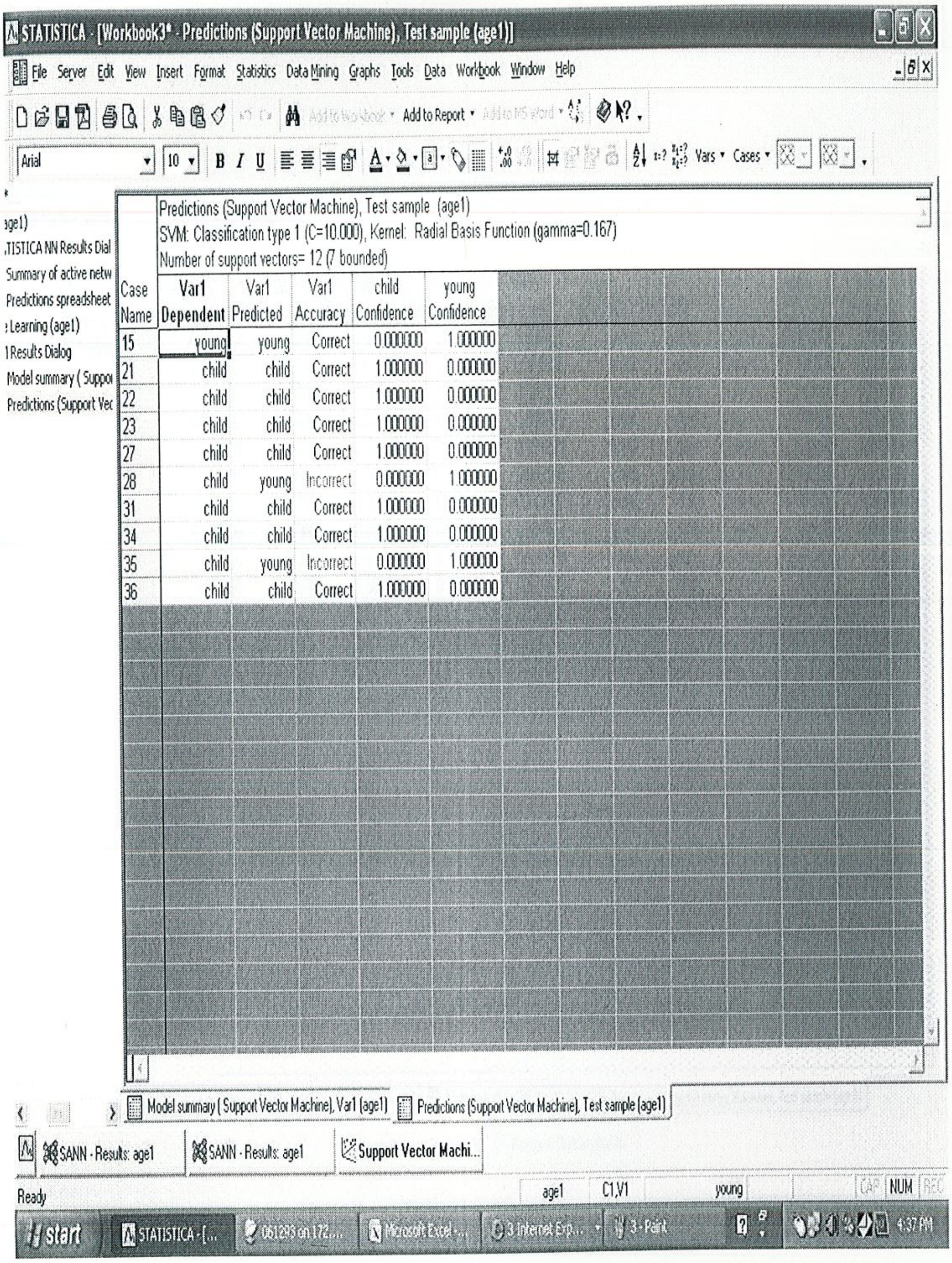

Support Vector Machine Results: age1
Dataset age1:
Dependent: Var1
Independents: Var2, Var3, Var4, Var5, Var6, Var7
Sample size = 30 (Train), 10 (Test), 40 (Overall)
Support Vector machine results:
SVM type: Classification type 1 (capacity=10.000)
Kernel type: Radial Basis Function (gamma=0.167)
Number of support vectors = 12 (7 bounded)
Support vectors per class: 5 (child), 7 (young)
Class. accuracy (%) = 100.000(Train), 80.000(Test), 95.000(Overall)

Figure 5.3(ii)



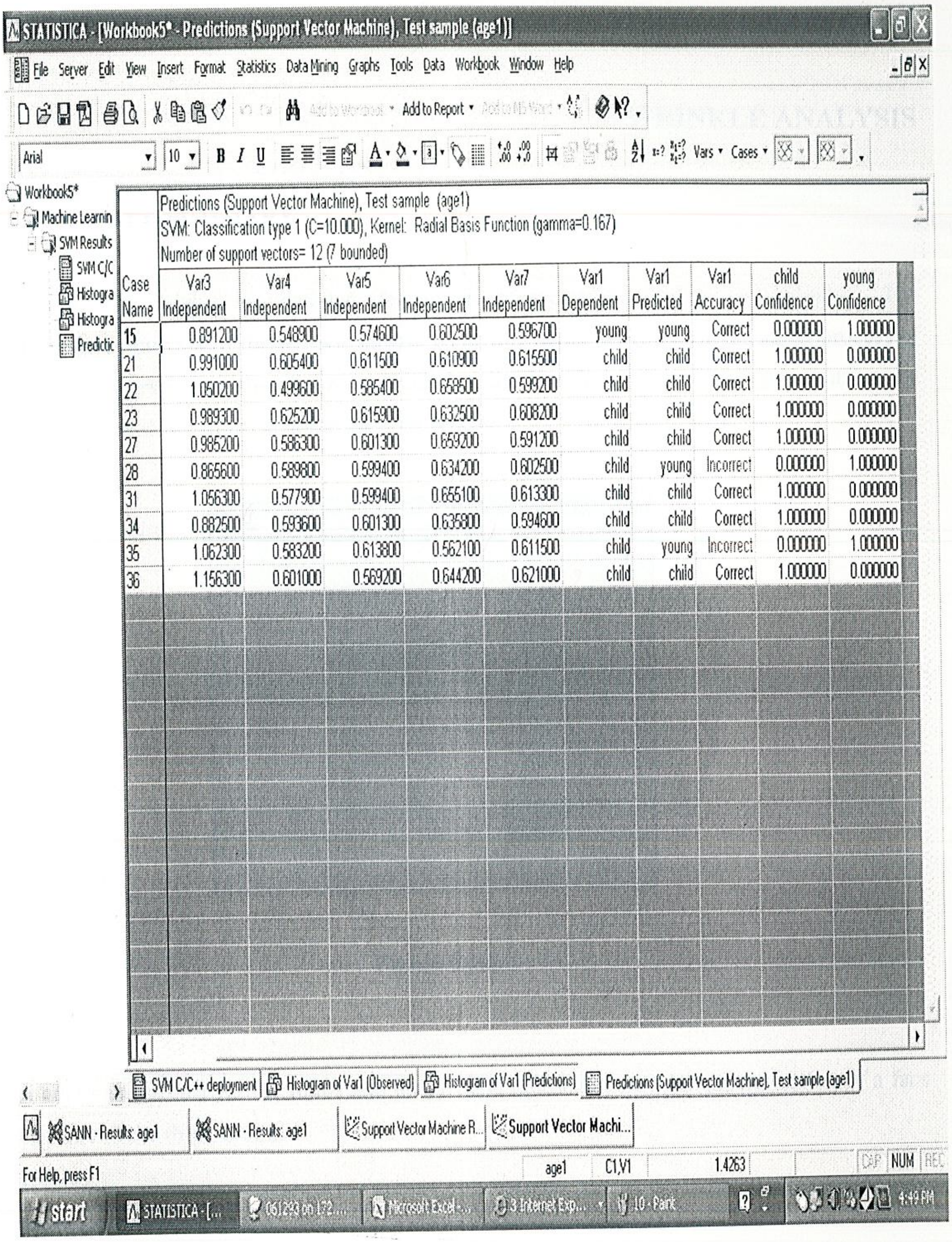
Case name	Var1 Target	Var1 - Output 1. RBF 6-10-2	Var1 - Output 2. MLP 6-11-2	Var1 - Output 3. RBF 6-11-2	Var1 - Output 4. RBF 6-11-2	Var1 - Output 5. RBF 6-8-2
2	young	young	young	young	young	young
4	young	young	young	young	young	young
5	young	young	young	young	young	young
6	young	young	young	young	young	young
8	young	young	young	young	young	young
9	young	young	young	young	young	young
10	young	young	young	young	young	young
11	young	young	young	young	young	young
12	young	young	young	young	young	young
13	young	young	young	young	young	young
14	young	young	young	young	young	young
15	young	young	young	young	young	young
16	young	young	young	young	young	young
18	young	young	young	young	young	young
19	young	young	young	young	young	young
20	young	young	young	young	young	young
21	child	child	child	child	child	child
22	child	child	child	child	child	child
24	child	child	child	child	child	child
26	child	young	child	child	young	child
27	child	child	child	child	child	child
28	child	child	child	child	child	child
30	child	child	child	child	child	child
32	child	young	child	child	child	child
33	child	child	child	child	child	child
34	child	child	child	child	child	child
35	child	child	child	child	child	child

Figure 5.3(iii)



Case Name	Var1 Dependent	Var1 Predicted	Var1 Accuracy	child Confidence	young Confidence
15	young	young	Correct	0.000000	1.000000
21	child	child	Correct	1.000000	0.000000
22	child	child	Correct	1.000000	0.000000
23	child	child	Correct	1.000000	0.000000
27	child	child	Correct	1.000000	0.000000
28	child	young	Incorrect	0.000000	1.000000
31	child	child	Correct	1.000000	0.000000
34	child	child	Correct	1.000000	0.000000
35	child	young	Incorrect	0.000000	1.000000
36	child	child	Correct	1.000000	0.000000

Figure 5.3(iv)



Case Name	Var3 Independent	Var4 Independent	Var5 Independent	Var6 Independent	Var7 Independent	Var1 Dependent	Var1 Predicted	Var1 Accuracy	child Confidence	young Confidence
15	0.891200	0.548900	0.574600	0.602500	0.596700	young	young	Correct	0.000000	1.000000
21	0.991000	0.605400	0.611500	0.610900	0.615500	child	child	Correct	1.000000	0.000000
22	1.050200	0.499600	0.585400	0.658500	0.599200	child	child	Correct	1.000000	0.000000
23	0.989300	0.625200	0.615900	0.632500	0.608200	child	child	Correct	1.000000	0.000000
27	0.985200	0.586300	0.601300	0.659200	0.591200	child	child	Correct	1.000000	0.000000
28	0.865600	0.589800	0.599400	0.634200	0.602500	child	young	Incorrect	0.000000	1.000000
31	1.056300	0.577900	0.599400	0.655100	0.613300	child	child	Correct	1.000000	0.000000
34	0.882500	0.593600	0.601300	0.635800	0.594600	child	child	Correct	1.000000	0.000000
35	1.062300	0.583200	0.613800	0.562100	0.611500	child	young	Incorrect	0.000000	1.000000
36	1.156300	0.601000	0.569200	0.644200	0.621000	child	child	Correct	1.000000	0.000000

Figure 5.3(v)

CHAPTER -6

WRINKLE ANALYSIS

6.1 CHAPTER SUMMARY

Section 6.2 contains the detailed description of secondary feature i.e. wrinkle . The subsections shows how the wrinkle length, width and depth are useful in the aging process of an elderly person and thus helps in predicting the age. The subsections also contains the overall architecture of the wrinkle analysis process.

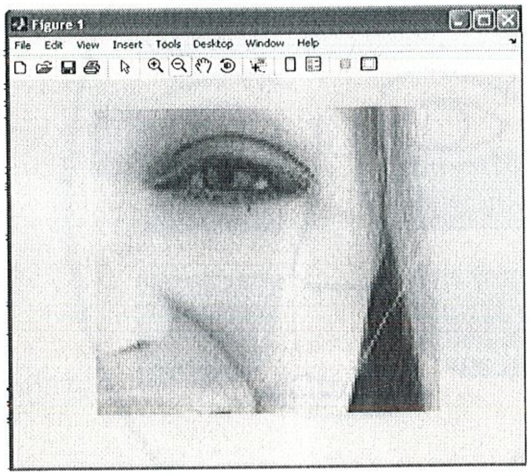


Figure 6.1(i)

A combination rule for the ratios and the wrinkle index thus permits categorization of a face into one of the three classes.

6.2 WRINKLE ANALYSIS

Skin is the outer layer of the human body and has long attracted a great deal of attention, since its appearance conveys useful information on the health condition of the subject. In our project we propose a skin age estimation scheme based on its wrinkle features such as length, width and depth, which represents the physical condition of skin statistically and quantitatively.

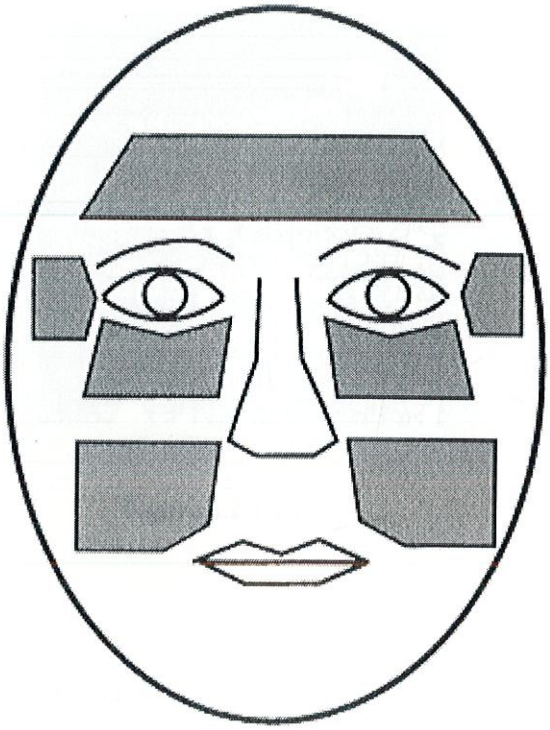


Figure 6.2(i)

It is on the basis of:

- Wrinkle length estimation
- Wrinkle width estimation
- Wrinkle depth estimation

6.2.1 WRINKLE LENGTH ESTIMATION

6.2.1.1 WATERSHED ALGORITHM

It basically segments images into regions according to the topology of the images.

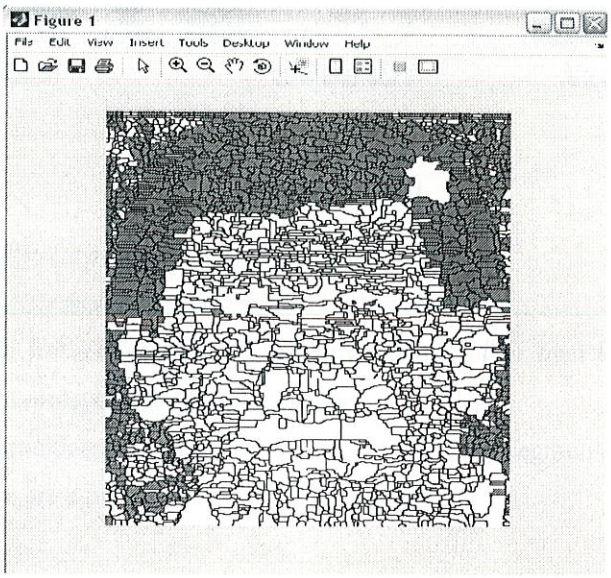


Figure 6.2.1.1(i)

6.2.1.2 WATERSHED PROCESS

- Skeletonise
- Euclidian distance maps(EUD)
- Ultimate eroded points (UEPs)
- Watershed

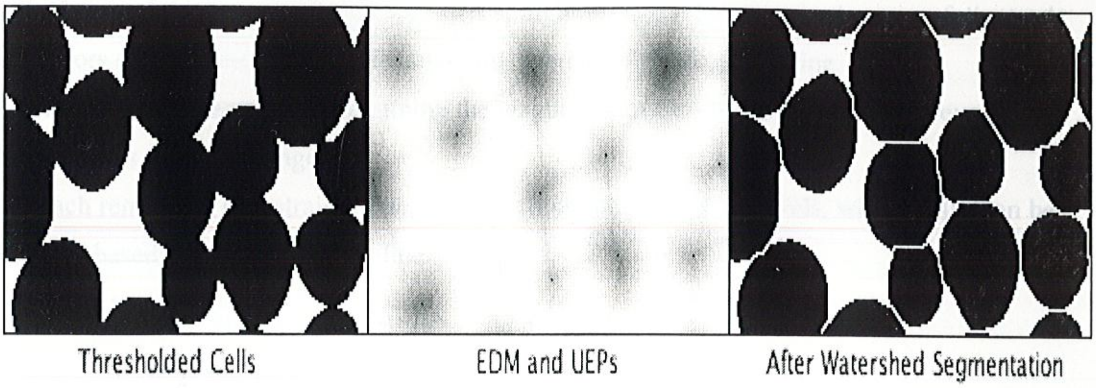


Figure 6.2.1.2(i)

- Now we are further focusing on the border lines rather than the regions, since the border lines correspond to the skin wrinkles.
- Watershed transformation performs 1-pixel line-based segmentation. To improve its accuracy, we are a new method, viz., *line sieving.*

Modifications in Watershed algorithm

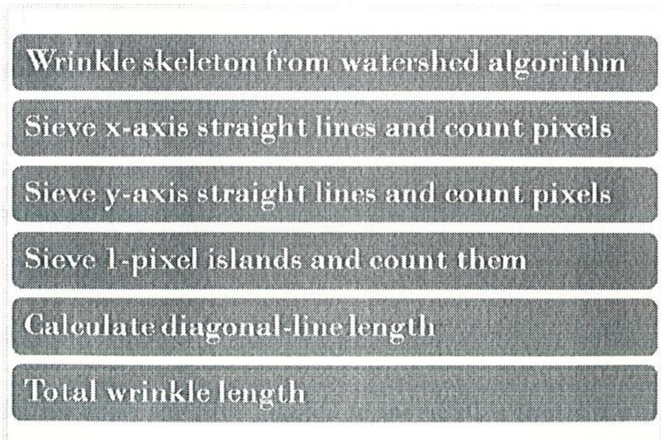


Figure 6.2.1.2(ii)

Calculation of the length of straight lines is straightforward. However, in the case of diagonal line errors are possible if we simply count the number of pixels on the line.

In order to reduce errors in determining the length of diagonal lines, we will first sieve all the straight lines from the image.

For each remaining non-straight line, we will count the number of pixels, which will then be adjusted based on the slope of the line.

6.2.2 WRINKLE WIDTH ESTIMATION

Algorithm

- procedure Thicken-by-morphological-growing
- *Input: grayscale image G(x,y),*
- binary wrinkle skeleton S(x,y)
- *Parameter: noise n, threshold t*
- *Output: grayscale wrinkle map W(x,y)*
- Do
- affectedPixel=0
- W(x,y) =Dilation(S(x,y))-S(x,y)
- for each point p(x,y) in W(x,y)
- if p(x,y) = 1 and G(x,y) > t then
- S(x,y) =1
- affectedPixel++
- End if
- End for
- While affectedPixel > n
- W(x,y) = S(x,y)

In the algorithm, we apply iterative dilation to the wrinkle skeleton image, until the skeleton lines have been fully recovered and represent the actual wrinkle widths. To this end, the comparison between the partially recovered image and original grayscale image is iterated until the difference between them is reasonably small.

6.2.3 WRINKLE DEPTH ESTIMATION

For more accurate skin age estimation, we considered an additional wrinkle feature, the wrinkle depth. On a typical skin image, the wrinkle region has a darker color. Also, a deeper wrinkle has a darker color, due to the interference with the light source. Based on this fact, we can define the wrinkle depth as the average color difference between the wrinkle region and the non-wrinkle region.

Equation shows detailed steps for calculating the wrinkle depth for the image *I*. *C = {c1,...,cn} denotes* randomly selected points from the skeleton lines. Since the skeleton lines are the approximate center lines of wrinkles, these points constitute various center points of wrinkle regions. For each center point c*i, we calculate the maximum* color difference for all the points whose distance from c*i is* half the wrinkle width plus *δ. The purpose of this is to get* the color information from its adjacent non-wrinkle region.

D(I)=1/n∑(i=1 to n) max(|I(si)-I(ci)|)

where |s-ci|=1/2 avg(width) + δ for □ ci € C

If we average their color differences for *n center points* , we get the wrinkle depth of the image.

6.2.4 SYSTEM ARCHITECTURE

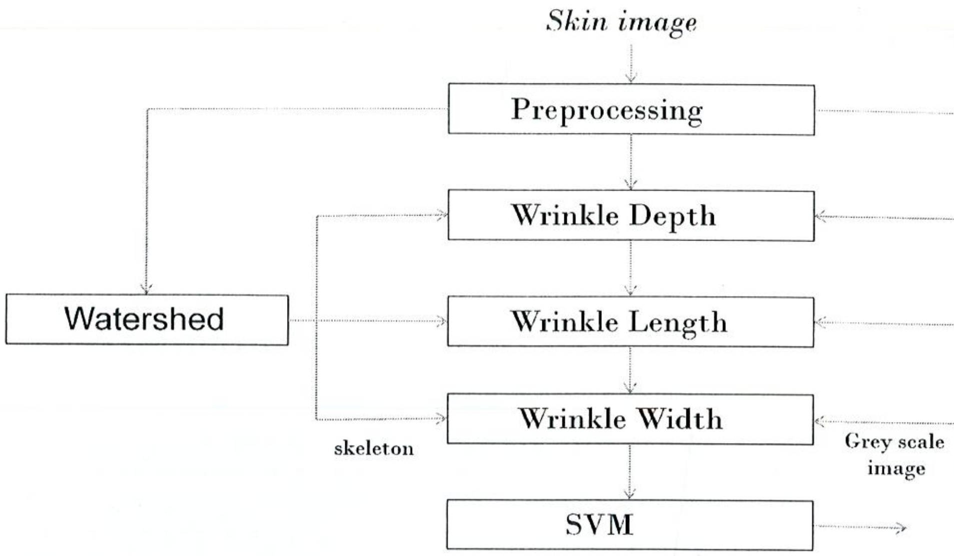


Figure 6.2.4(i)

CHAPTER 7

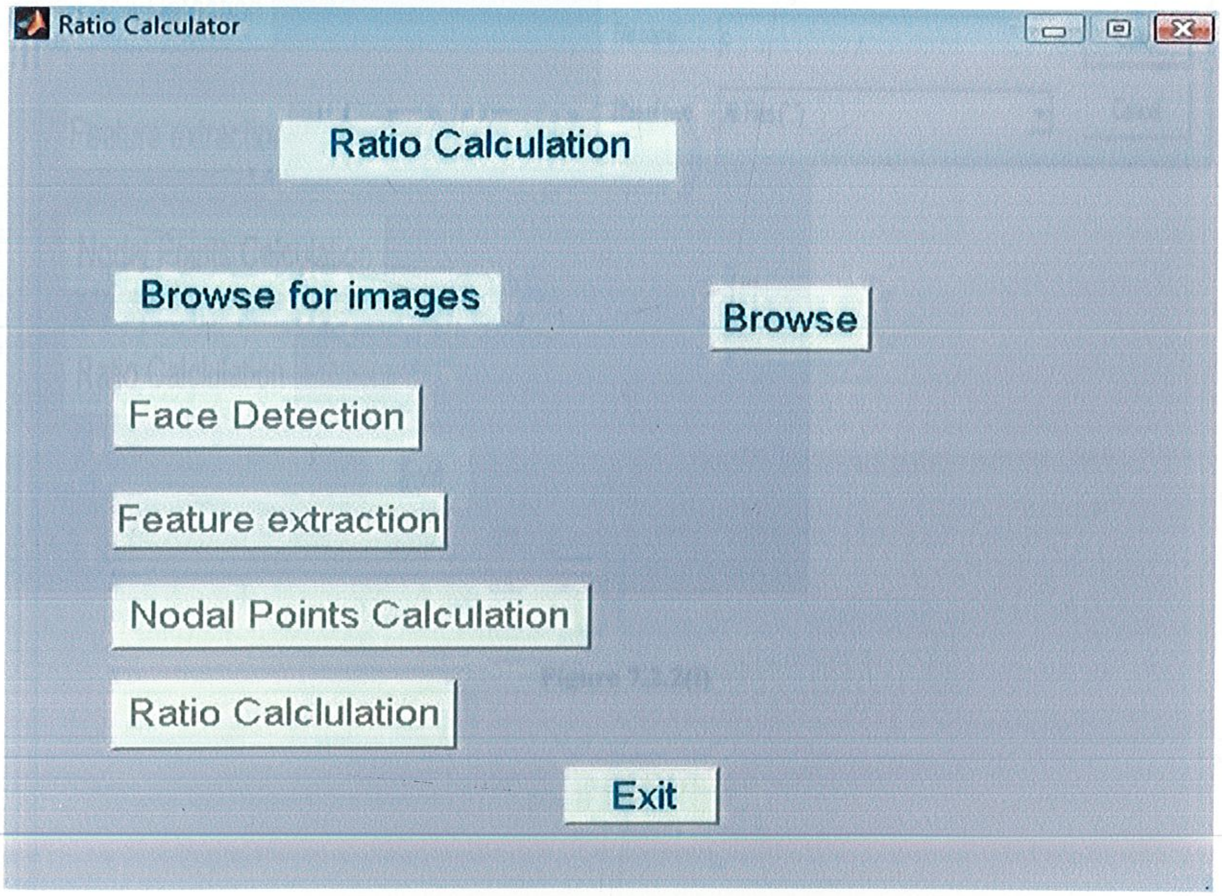
FLOWCHARTS, DFDs AND CODES AND SNAPSHOTS OF GUIs

7.1 CHAPTER SUMMARY

Section 7.2 of chapter 7 contains the main GUI snapshots of the project. Section 7.3 contains the flowchart showing all the stages involved. Section 7.4 contains the dataflow diagrams of the entire process. Finally Section 7.5 contains the main codes used in the project.

7.2 SNAPSHOTS

7.2.1 MAIN GUI (1)



7.2.2 MAIN GUI (2)

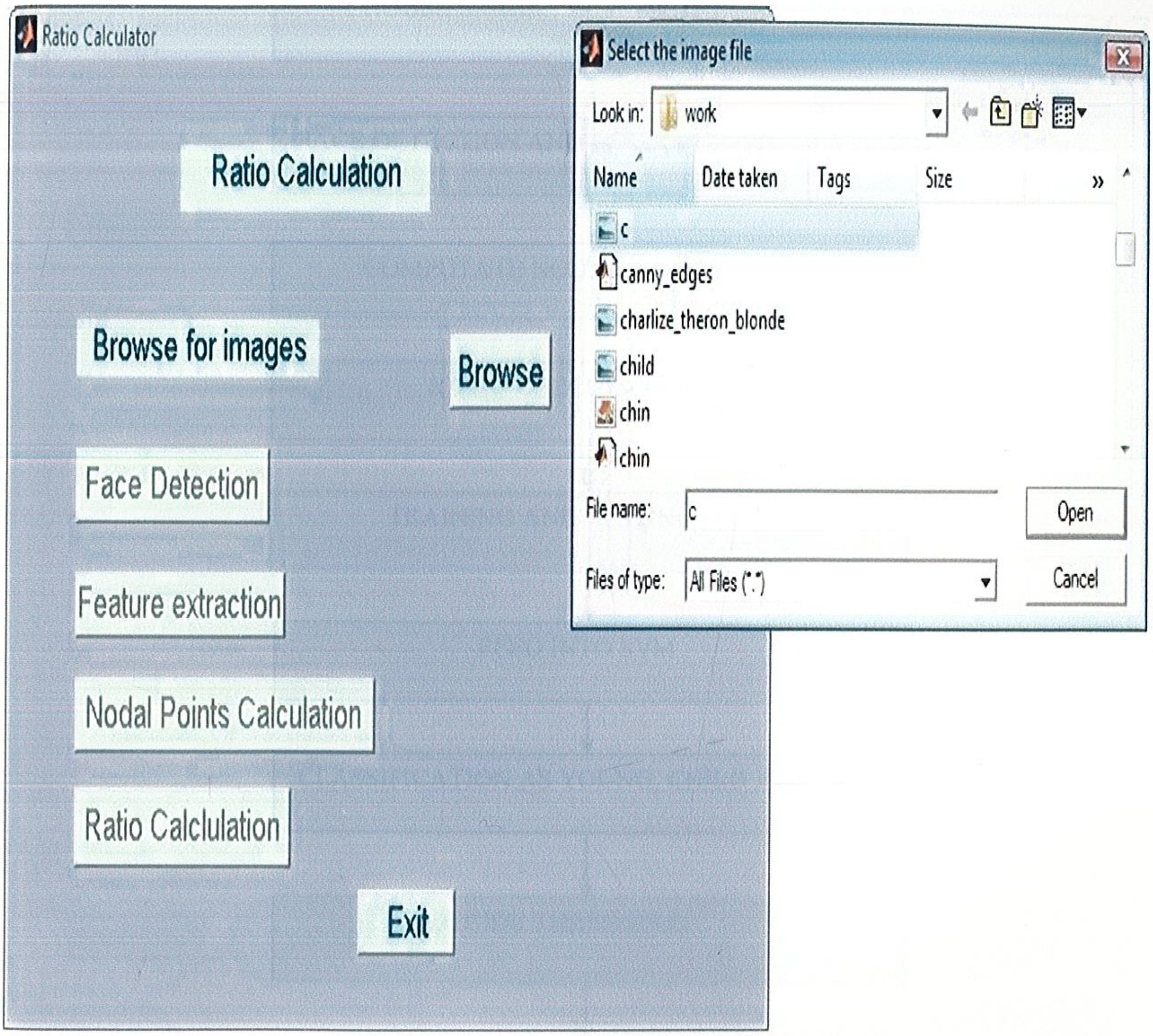


Figure 7.2.2(i)

7.3 FLOWCHART

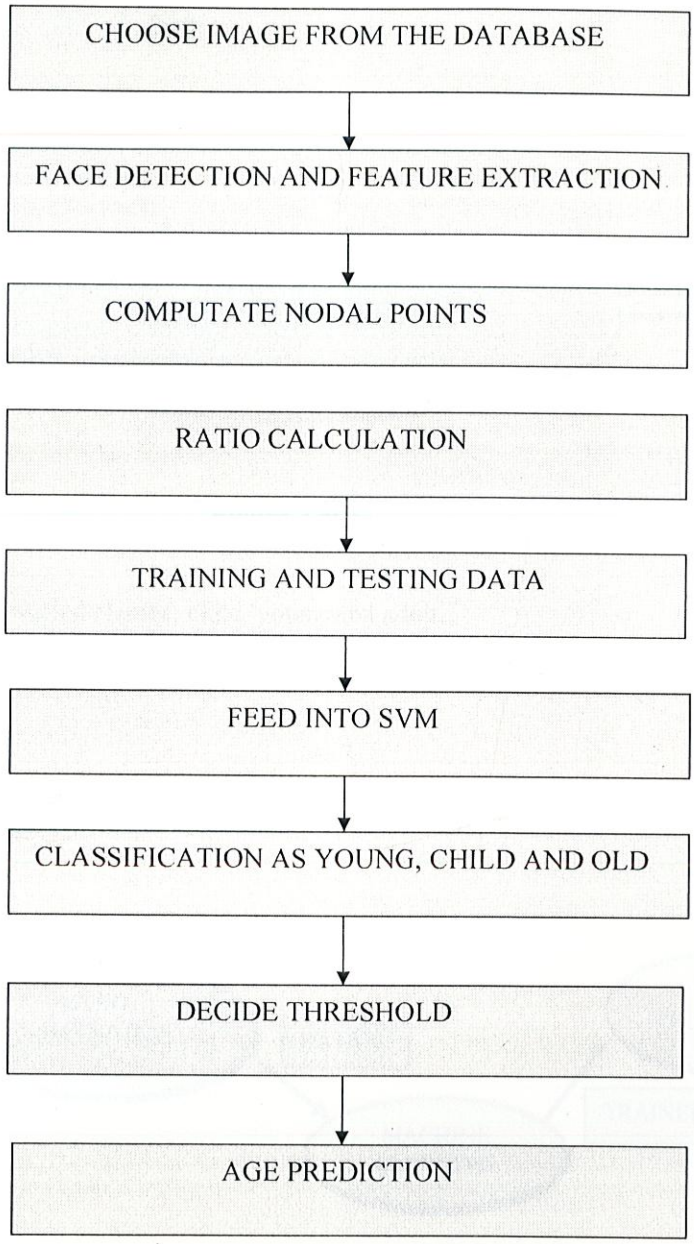


Figure 7.3(i)

7.4 DFDs

7.4.1 CONTEXT LEVEL DFD

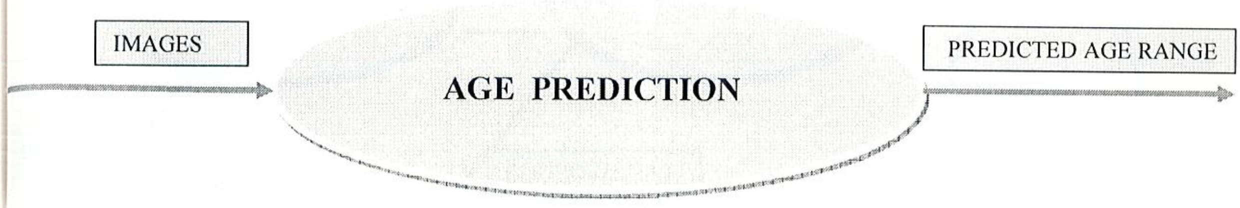


Figure 7.4(i)

Input: Images of varied classes: child, young and adult.

Output: Predicted age range.

7.4.2 LEVEL 1 DFD

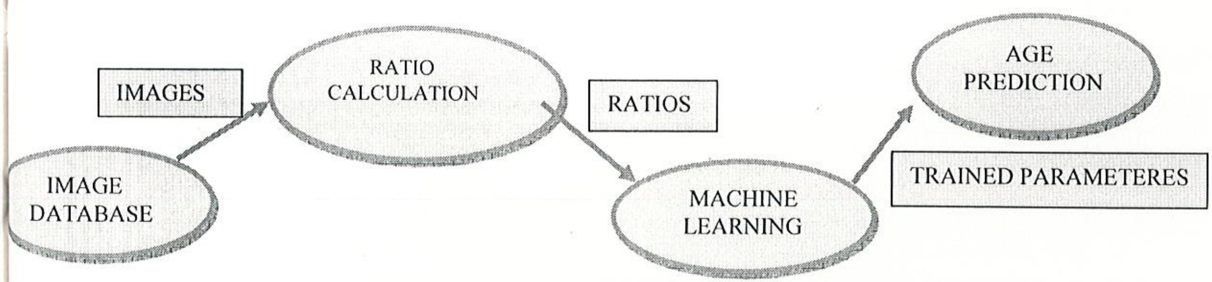


Figure 7.4(ii)

7.4.3 LEVEL 2 DFD

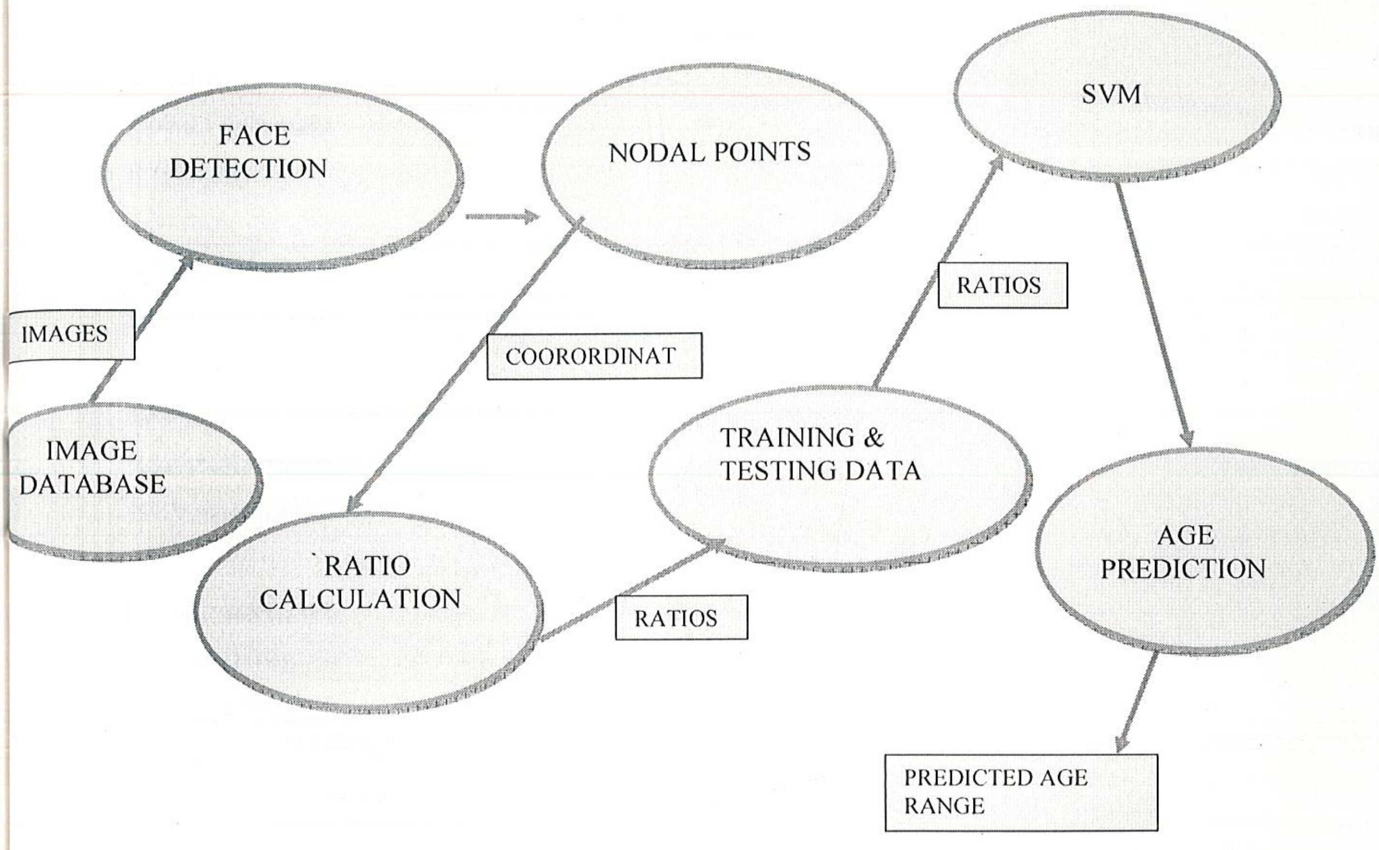


Figure 7.4(iii)

7.5 CODES

7.5.1 FACE DETECTION

```
global I;
global FileName;
I=rgb2gray(I);

BW=im2bw(I);
figure,imshow(BW)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%
%%%%% minimisation of background portion

[n1 n2]=size(BW);
r=floor(n1/10);
c=floor(n2/10);
x1=1;x2=r;
s=r*c;

for i=1:10
   y1=1;y2=c;
   for j=1:10
      if (y2<=c | y2>=9*c) | (x1==1 | x2==r*10)
         loc=find(BW(x1:x2, y1:y2)==0);
         [o p]=size(loc);
         pr=o*100/s;
         if pr<=100
            BW(x1:x2, y1:y2)=0;
            r1=x1;r2=x2;s1=y1;s2=y2;
            pr1=0;
         end
         imshow(BW);
```

```
        end
            y1=y1+c;
            y2=y2+c;
    end

 x1=x1+r;
 x2=x2+r;
end
 figure,imshow(BW)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%
%%%%%% detection of face object

L = bwlabel(BW,8);
BB  = regionprops(L, 'BoundingBox');
BB1=struct2cell(BB);
BB2=cell2mat(BB1);

[s1 s2]=size(BB2);
mx=0;
for k=3:4:s2-1
   p=BB2(1,k)*BB2(1,k+1);
   if p>mx & (BB2(1,k)/BB2(1,k+1))<1.8
      mx=p;
      j=k;
   end
end
figure,imshow(I);
hold on;
rectangle('Position',[BB2(1,j-2),BB2(1,j-1),BB2(1,j),BB2(1,j+1)],'EdgeColor','r' )
```

7.5.2 FEATURE EXTRACTION

```
global FileName;
global I;

%------------------extracting eye1--------------
I=imread(FileName);
for r=157:200
for c=79:148
e1(r-156,c-78)= I(r,c);
end;
end;
e2=mat2gray(e1);
figure, imshow(e2);
imwrite(e2,'eye1.bmp');

%---------------extracting eye2------------------
%I=imread(FileName);
for r=157:200
for c=170:245
ee1(r-156,c-169)= I(r,c);
end;
end;
ee2=mat2gray(ee1);
figure, imshow(ee2);
imwrite(ee2,'eye2.bmp');

%---------------extracting nose------------------

%I=imread(FileName);
for r=176:255
for c=133:189
n1(r-175,c-132)= I(r,c);
end;
```

```
end;
n2=mat2gray(n1);
figure, imshow(n2);
imwrite(n2,'nose.bmp');

%---------------extracting chin-----------------------------------------

%I=imread(FileName);
for r=316:353
for c=125:208
c1(r-315,c-124)= I(r,c);
end;
end;
c2=mat2gray(c1);
figure, imshow(c2);
imwrite(c2,'chin.bmp');

%------------------extracting lips--------------------------------------

%I=imread(FileName);
for r=267:308
for c=114:212
m1(r-266,c-113)=I(r,c);
end;
end;
m2=mat2gray(m1);
figure, imshow(m2);
imwrite(m2,'lips.bmp');
```

7.5.3 NODAL POINT CALCULATION

```
global FileName;
global I;

global xfine;
global yfine;
global xfine1;
global yfine1;
global xfinn;
global yfinn;
global xfinl;
global yfinl;
global xfinc;
global yfinc;
global middlex;
global middley;


P=imread('eye1.bmp');
Q=im2bw(P,0.2);
figure, imshow(Q);
imwrite(Q,'fig1.bmp');
A=imread('fig1.bmp');
B=imread('mask1.bmp');
B=~B;
imwrite(B,'mask2.bmp');
figure, imshow(B);
A=+A;
B=+B;
A1=uint8(A);
B1=uint8(B);
j=xcorr2(A1,B1);
```

```
e=mat2gray(j);
figure, imshow(e)
j1=im2bw(e);
imwrite(j1,'result1.bmp');
[i,map]=imread('result1.bmp');
figure , imshow(i)
[x,y]=size(i)
% top point  %
m = 1;
n = 1;
while (m<x)
while (n<y)
 if (i(m,n)~=0)
   upe=m
   m = x;
   n = y;
 end
 n=n+1;
end
  n = 1;
  m=m+1;
end
upe;


% bottom point %
m = x;
n = 1;
while (m>1)
while (n<y)
 if (i(m,n)~=0)
   downe=m
   m = 1;
   n = y;
```

```
  end
 n=n+1;
end
  n = 1;
  m=m-1;
end
downe;


% left point %
n = 1;
m = 1;
while (n<y)
while (m<x)
 if (i(m,n)~=0)
   lefte=n
   m = x;
   n = y;
 end
 m=m+1;
end
  m = 1;
  n=n+1;
end
lefte;
% right point %
n = y;
m = 1;
while (n>1)
while (m<x)
 if (i(m,n)~=0)
   righte=n
   m = x;
   n = 1;
```

```
 end
 m=m+1;
end
  m = 1;
  n=n-1;
end
righte;


yfine = (upe + downe)/4;
yfine = yfine + 158
xfine=(lefte + righte)/4;
xfine = xfine + 78
```

CONCLUSION AND DISCUSSION

We have outlined a computational theory for visual age classification from facial images. For now, only three age-groups were considered: babies, young adults, and senior adults. First, primary features of the face, namely the eyes, nose, mouth, chin, and virtual top of the head, are found. From these, ratios are computed that permit the distinguishing of babies from others.

Next, secondary features, namely wrinkles, will be detected and analyzed. This step permits the distinguishing of seniors from those in the two younger categories.

This work has shown that computing ratios can yield age categorization. These criteria were suggested by cranio-facial research and the observation that aging skin develops wrinkles. There are several directions that need to be further explored. The problem of varying orientation of the face needs to addressed . The work thus far has assumed mugshot viewpoints, and this makes the ratio computations easy.

Next, skin-color and eye-color restrictions will need to be loosened. Then, the computation of age in the presence of eye patches and dark glasses, and other occlusions and shadowing effects needs to be explored. Age computation must be made robust to varieties of moustaches, facial scars, and disheveled hair. Finally, an accurate estimation of the top of the skull has defied all approaches thus far. As we showed, this estimate enhance the ability to tell age and, thus, needs to be computed. In all of these endeavors, additional age-related information can be exploited: the growth of the nose and the nose-bridge, and the relative shrinking of the iris-size over time, and changes to the outline of the face.

BIBLIOGRAPHY

We have taken help of the following books and research papers.

1. Stuart Russell and Peter Norvig. Artificial Intelligence: A Modern Approach.
Pearson Education, Inc., 2003.

2. Richard O. Duda, Peter E. Hart, and David G. Stork. John Pattern Classification..
Wiley & Sons, 2001.

3. Michael Seul, Lawrence O'Gorman, and Michael J. Sammon. Cambridge. Practical
Algorithms for Image Analysis: Description, Examples, and Code.
University Press, 2000.

4. Young H. Kwon¤ and Niels da Vitoria Loboy
Age Classification from Facial Images
School of Computer Science, University of Central Florida, Orlando, Florida 32816
September 6, 1996

5. Face Detection
University at buffalo CSE666 Lecture Slides

6. CS 297 Report by Wallun Chan
Recognition And Age Prediction With Digital Images
Advisor: Dr. Chris Pollett
Department of Computer Science, San Jose State University
May 2005

Web Refrences

www.ee.ucl.ac.uk

www.vasc.ri.cmu.edu