HUMAN MOOD DETECTOR BY USING DIGITAL IMAGE PROCESSING

By

NOOR AZLIN BT ALI

FINAL PROJECT REPORT

Submitted to the Electrical & Electronics Engineering Programme
in Partial Fulfillment of the Requirements
for the Degree
Bachelor of Engineering (Hons)
(Electrical & Electronics Engineering)

Universiti Teknologi Petronas
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

HUMAN MOOD DETECTOR BY USING DIGITAL IMAGE PROCESSING

by

Noor Azlin bt Ali

A project dissertation submitted to the
Electrical & Electronics Engineering Programme
Universiti Teknologi PETRONAS
in partial fulfillment of the requirement for the
Bachelor of Engineering (Hons)
(Electrical & Electronics Engineering)

Approved:

Prof. P.A. Venkatachalam
Project Supervisor

Prof. Dr. P.A. Venkatachalam
Professor
Electrical & Electronic Engineering
Academic Block No. 22
Universiti Teknologi PETRONAS
Bandar Seri Iskandar
31750 Tronoh, Perak Darul Ridzuan, MALAYSIA

UNIVERSITI TEKNOLOGI PETRONAS

TRONOH, PERAK

June 2006

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

Noor Azlin bt Ali

ABSTRACT

The human face provides the most salient clue to person's emotional state. The face expressions are recognized using features such as eye, eyebrow, mouth or nose. These features play a vital role in human mood recognition. This project, which is named as 'Human Mood Detector', will be recognizing a set of emotions, which consists of angry, sad and happy, requiring vast of knowledge in Digital Image Processing. These projects are divided into two categories, software development in the first semester, and implementation in the next semester. The identification of a person's mood is based on input from digital camera. Then, the input image will be undergo digitization, noise removal, resizing, format transformation, contrast enhancement, segmentation and feature extraction. Feature extraction is a crucial stage in this system, since at this stage; the important feature information from the image will be extracted, which will lead to the desired result. A few distinct criteria that will distinguish the emotion are recognized, using segmented images of mouth, forehead or eye shape. These distinct criteria will be used to develop an algorithm that differentiates all the three emotions described before. To classify a person's mood, a similarity matching procedure will be performed from the input image whereby the input will be matched with the set of criteria that has been classified in the program. Thirty images will be used in the second semester with different set of mood for testing. If, the extracted information falls within the classified features of a certain mood, the mood of the person will be identified. Identification result will be presented at the end with the probability. The theory, concept, proposed methodology as well as software development (coding) are represented in this report.

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to the Allah the Almighty for giving the chance to complete this project successfully. I also would like to thank my project supervisor, Prof P.A. Venkatachalam for giving me this opportunity to work under his guidance and support. The chance to work under his supervision with his vast experience in Digital Image Processing is highly appreciated. Without his numerous advices and help, it would not be possible for me to complete this project. Deepest appreciation is also given to Miss Kavitha and Mr Faizan for providing advice, ideas as well as reference materials to aid in this project.

I also would like to thank all the individuals who volunteered to be the models for the test images for this system. Last but not least, I simply must thank to my family for their support, both spiritual and financial supporter throughout my five years in Universiti Teknologi Petronas. Without them, I might not be able to execute this project successfully.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

GUI	Graphical User Interface
JPEG	Joint Photographic Experts Group
SVD	Singular Value Decomposition
ROI	Region of Interest

CHAPTER 1
INTRODUCTION

1.1 BACKGROUND OF STUDY

This project is a development of a system that will identify human mood based on digital image. It is conducted basically to determine the relationship between facial expressions with the human mood. Facial expression is in terms of the change in eyes, lips, nose, eyebrows or any distinct area that will differentiate human mood in three different categories, which are angry, sad and happy faces.

The objective of this project is to detect a person's mood, even though a person can pretend to be different but yet there is still a criterion that will distinguish his mood.

Set of facial expressions with thirty images is used to extract and perform a multi-resolution analysis that yield a good result in identification of the mood. Digital image processing functions will be used to obtain the intended result. The input for this system is the image of a person's facial expression with unknown mood, which is acquired by using a camera and the output will be the detection of the person's mood. The coding will consist of many steps, such as reading the test image and preprocessing, whereby the image will be filtered to be free from noise.

Image that are artifacts which has caused by the digitization process or by other faults in the imaging set-up (for example, bad lighting) are corrected using Image enhancement techniques. Then the image is segmented and the feature extraction operations are performed to obtain useful information from the image. Most important point is to see that when performing the feature extraction is it is to be ensured that images taken are in the same environment and by using the same camera to avoid uncertainty of the results.

1.2 PROBLEM STATEMENTS

1.2.1 Problem Identification

Faces are rich in information about individual identity, mood and mental state. Nowadays, many researches have been done by expertise to relate the computing power to process images of faces for different purposes. This presents the possibility of new ways for human to interact with computers and computer controlled systems.

The aim of this project is to develop a system that would perform the task of categorizing a person's mood on three categories, namely angry, happy and sad. To identify the moods and for the purpose of analyzing the image, thirty sets of images, which are captured by digital camera are used for the training sets. However, the model used for photographing and the photographer must be static, which means the distance between both of them must be determined. Beside that, the specification of digital camera that has been used must be identified, and it is better to have the same digital camera for all models to avoid uncertainty of the desired result. The lighting used to capture the image must be the same.

1.2.2 Significance of the Project

This system can be used for a variety of purposes, such as usability studies, whereby logging facial expression during testing of new products or for retailing whereby the system developed will be recognizing customers to monitor their preferences. Besides that, this project can be implemented to record moods of students in a class during a lecture, during teacher and student interaction, and also to identify the state of emotion of persons on certain occasions.

1.3 OBJECTIVES AND SCOPE OF STUDY

1.3.1 The Relevancy of Project

This project is directly using digital image processing. It incorporates the usage of computer vision and MATLAB programming language. The entire information obtained through the computer vision and image processing as well as MATLAB software is applied towards the development of this project. This project can be applied in many areas, such as personal identification. This system may also play vital role to handle patients, disable people and senior citizens. The main beneficiaries of this work are hospital patients, nurses and doctors and also other places where human moods are to be observed.

1.3.2 Feasibility of Project Within The Scope and Time Frame

This project has been divided into two semester and both involved software development. For the first semester, the author will be involved in software development for restricted application, while for the second semester, the author will be expand the project for general implementation. A good project management technique using the time constraint efficiently will be adopted in order to complete the project successfully *(Appendix A for Gantt Chart for the first and second semester).* It is safe to say that this project is feasible to be completed within the given timeframe.

CHAPTER 2
OBJECTIVES

This project aims to achieve the following objectives:

i. To develop Human Mood Recognition System (HuMoReS) with a possible level of accuracy and reliability and a system which is robust and efficient

ii. To apply high level image processing and computer vision concepts, such as segmentation, pattern recognition, and feature extraction in the problem of human mood detection.

iii. To implement a human mood detector that will be able to identify a person's mood. The author will be developing software coding by using selected images in the database to accomplish the project.

CHAPTER 3

LITERATURE REVIEW/THEORY

In order to obtain relevant and beneficial information regarding this project, the author carried out an extensive literature review, such as by referring to books, journals, magazines, websites, etc. The information is very important for the development of the system as it provides theories, concepts and as well as techniques that be utilized throughout this project.

3.1 Face Recognition

The face recognition has become very active since 1995, when Brunelli et al. [1] used template matching for face recognition. The algorithm consists of a set of four masks representing eyes, nose, mouth and face for each registered person (Figure 3.1).

Figure 3.1: Four masks used in the template matching [1].

To identify the unknown person in the image, the algorithm first detects eyes using template matching and then normalizes position, scaling and rotation of the face in the image plane using the detected eye position. Next, for each person in the database, the algorithm places his four masks on their positions relative to eye position and computes the cross-correlation values between the four masks and the image. The unknown person in the image is classified as the person giving the greatest sum of the cross-correlation values of the four masks.

3.2 Face Emotion Recognition

Human being possesses an ability of communication through emotion in day-to-day interactions with others. Some emotions have attracted most of the interest in human computer interaction environments. The face emotion recognition includes the capturing of the real-time image using digital camera. The preprocessing, feature extraction and classification are done simultaneously and a decision on the emotion recognition is achieved. A robot which named as IFBOL [2] is able to understand a partner's feelings, and it can detect emotion, recognize conversation, go by affection of conversation partner and speak with sentiment. This IFBOL robot is able to communicate with people upon their feeling and stress.

In preprocessing, some processes such as gray scaling, edge detection and filtering [3] can be performed. These processes are necessary for a good feature extraction and also for removing the unwanted noise in the images. The Gaussian filtering and a two level filtering can be applied to the image [4]. After preprocessing, feature extraction has to be carried out. The features of eye, lips, cheek, eyebrows, mouth and skin contraction are extracted. There are several feature extraction methods available in the literature. Among the feature extraction methods, wavelet transform, eigenface, projection profile can be considered. These extracted features play a vital role in identifying the emotion of the face.

3.3 Pattern Recognition and Classification

Pattern recognition deals with the recognition of objects in images, and is applicable to any other kind of data as well. The basic approach views an instance to be recognized as a vector of measurements. A recognition system must contain some memory of the objects that it is to recognize. Recognition and learning of patterns are subjects of considerable depth and interest to cognitive psychology, pattern recognition and computer vision.

Classification is the process of grouping objects together into classes according to their perceived likeness or similarities [5]. Pattern classification involves taking the feature extracted from the image and using them to classify image objects automatically. This is done by developing classification algorithms that use the feature information. The distance or similarity measures are used for comparing different objects and their feature vectors.

3.4 Facial Expression Analysis

There is a long history of interest in the recognition of emotion from facial expressions, influenced by Darwin's pioneering work [6] and extensive studies on face perception. Face perception is very important component of human cognition as faces are rich in information about individual identity, but also about mood and mental state. Facial expression interactions are relevant in social life, teacher-student interaction, and credibility in different contexts, medicine, etc. Success in automatic recognition of emotion would lead to new evolutionary devices offering the possibility of new ways for humans to interact with computer systems.

3.5 Human Mood Detection [7]

The classification is based upon a branching flow, with an appropriate test at each node (Figure 3.2). The emotions are divided into broad categories and further tests are performed to identify each emotion individually. In this test, four emotions are divided into two categories, bad and good. Happy and surprise fall into good emotion, while sad and angry are bad moods.

The image is cropped to the mouth region, which is the primary interest. The highest-level test is TEST 1. It is performed to distinguish between pictures representing a happy or positive mood emotion from those representing a bad or negative emotion. Once a mood was detected, as either being happy or sad mood, it will then be tested again to classify it as one of the two moods in each category. The two pictures which are identified as having positive emotions are then sent to TEST 2, which distinguishes between a happy, and a surprised emotion. The other two pictures are to TEST 3, which distinguishes, between sad and angry.

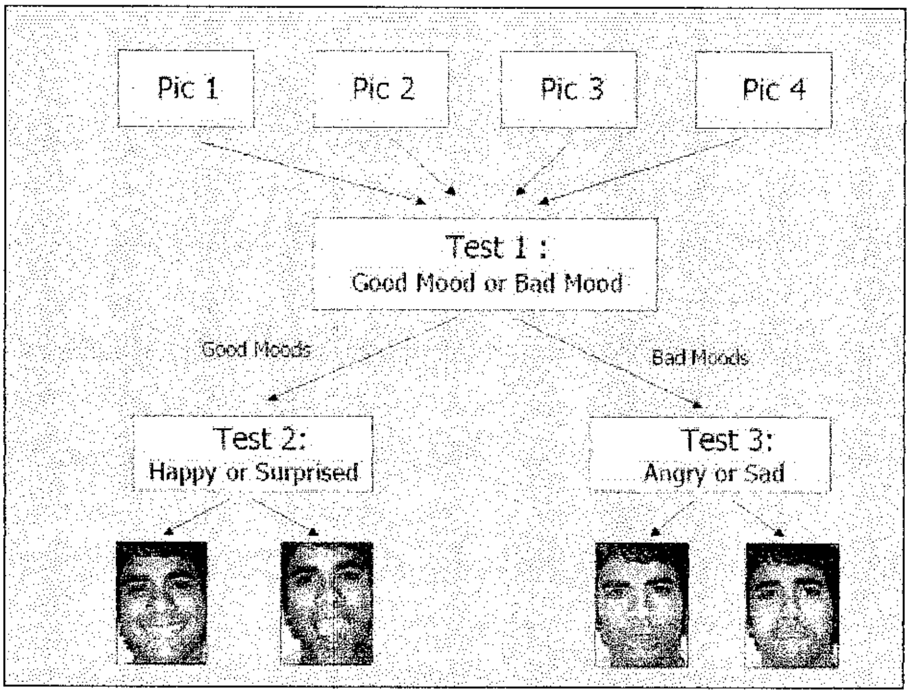

Figure 3.2: Control Flow [7]

This method was quite effective as it reduced the total number of tests to be performed to three. Having a single test to correctly detect an emotion was not feasible because it involved setting thresholds, which would not hold over a wide variety of test cases and also increase the total number of tests being performed.

GoodCrop : Image Cropping Routine

GoodCrop function is used to crop the mouth from an image. It is called from the original image and returns the part of the image containing the mouth. It is based on the sobel edge detector. Due to the image specifications, it is safe to assume that the mouth is located in the lower third of the picture. The function makes this basic assumption and does an initial crop to this size. It then performs the edge detection and counts the number of edge points along each row and column. It also does an 11 point smoothing on the row and column summations to negate the effects of the nose being part of the initial crop.

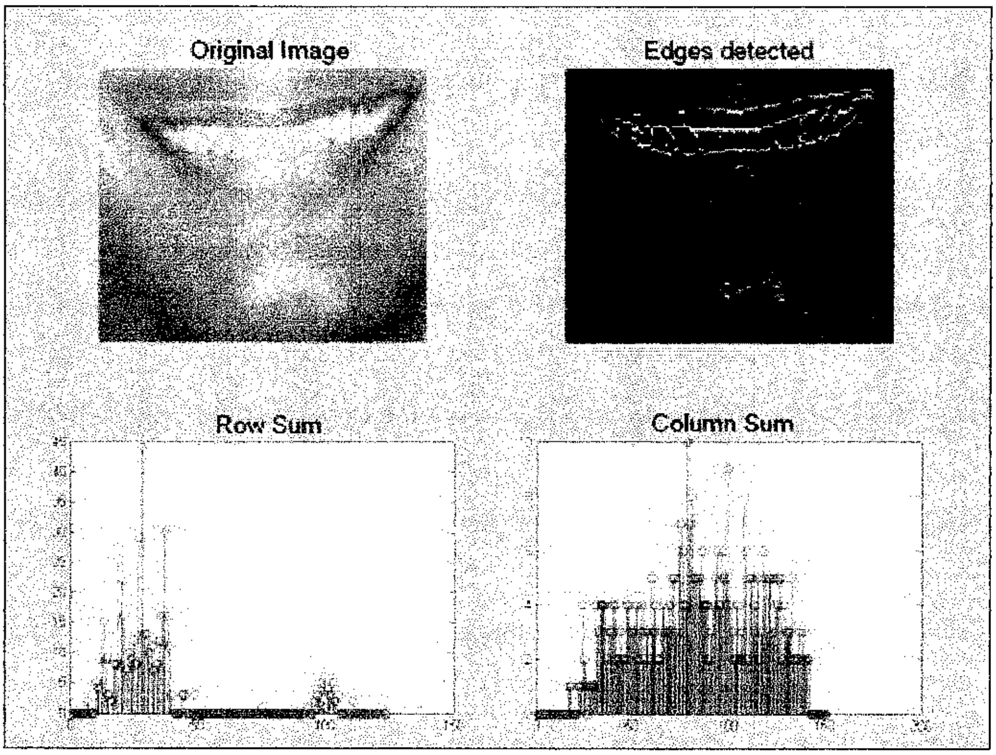

Figure 3.3: Original image and edges detected histogram [7]

It then chooses the peak corresponding to the mouth and the extent of the peak to decide on the location and extent of the mouth vertically and horizontally.

Figure 3.4: Pre and Post Cropping [7]

Test 1: Good Mood or Sad Mood

This test is used to distinguish between a good mood and a sad mood. It is based on the use of the 2D Gabor wavelet transformation. Daugman pioneered use of the 2D Gabor wavelet representation in the 1980s. A complex-valued 2D Gabor function is a plane wave restricted by a Gaussian envelope.

The Gabor wavelet representation allows description of spatial frequency structure in the image while preserving information about spatial relations. The complex amplitude of the transforms is used to test for features.

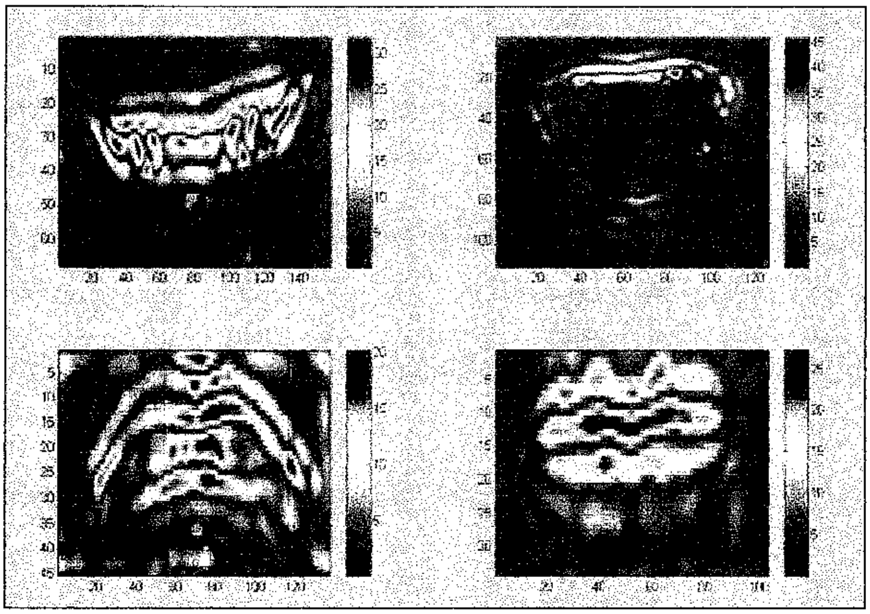

Figure 3.5: Gabor representation of the four moods for subject 1. [7]

Since the Gabar wavelet highlights and extracts features from a facial image, those images with stronger expressions have greater intensity in their transforms relative to those with weaker expressions. The happy and surprised pictures (Figure 3.5), which are on top, have relatively higher values when compared to the sad and angry pictures. After taking the 2-norm of each transform, it is able to classify the strongest two as happy or surprised and the weaker two as sad or angry.

Test 2 : Happy or Surprised

Intuitively, the distinguishing feature between a smile and a surprise is the teeth and a measure of how open the mouth is. While more teeth are visible during a smile, a surprise emotion has a more open mouth. Teeth are white and it corresponds to a high intensity, a process is applied to determine the intensity of the mouth and consequently, whether the person is smiling or not.

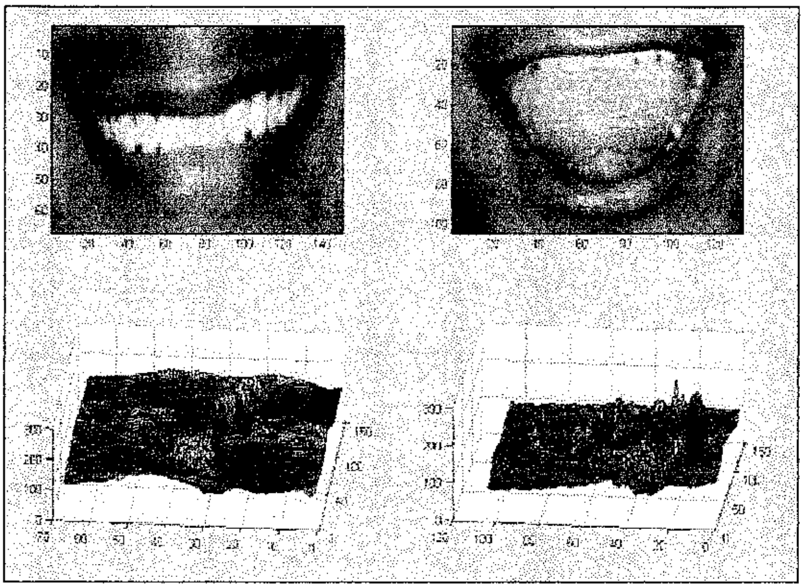

Figure 3.6: Surf plots for happy and surprised [7]

The detector first creates high-contrast versions of the original cropped pictures. It then designates a threshold value that is 95% of the maximum value, which is usually white, and determines the number of such peaks that exist. The image with the greater number is designated "happy", and the other as "surprised".

Test 3 : Sad or Angry

A difficulty happened in deciphering a sad from angry image since both emotions usually entail a closed mouth. With a closed mouth there was no way to compare intensity or even edges, as teeth create in a happy picture. CAAM Professor Dr. Mark Embree suggests Singular Value Decomposition (SVD) transform, in order to analyze images. Figure 3.7 indicates the result after applying the above technique to the mouth region.

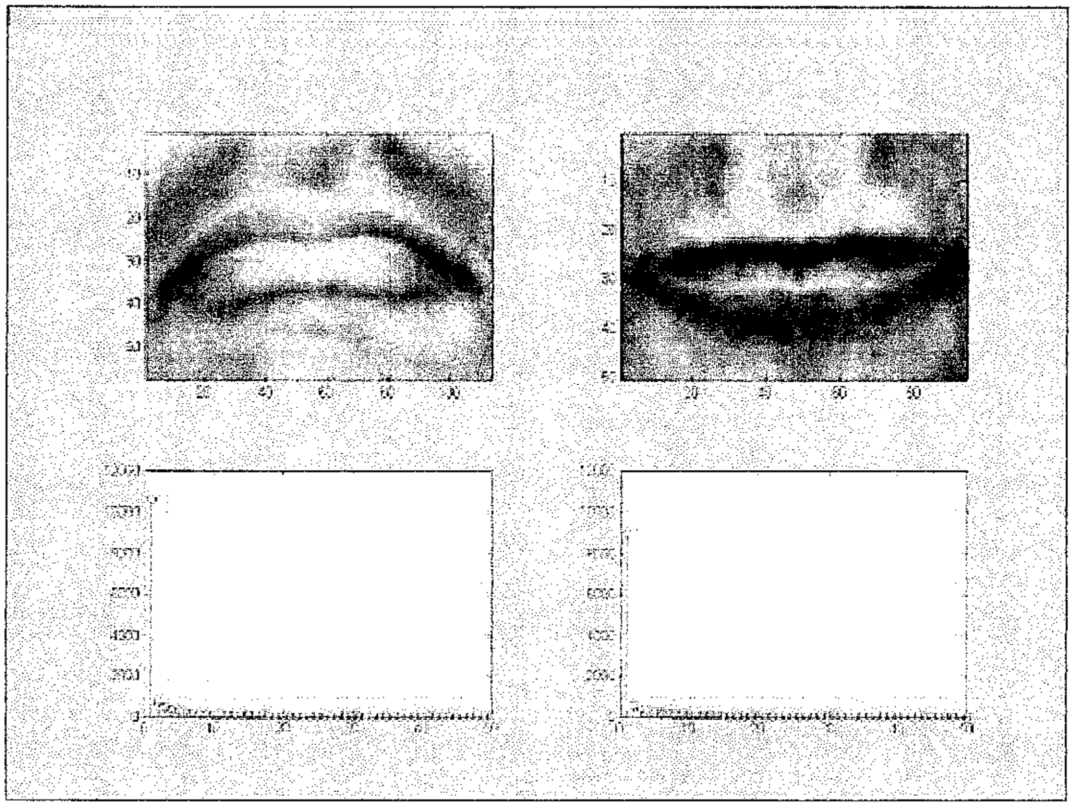

Figure 3.7: Image and corresponding svd plot for sad and angry emotions [7]

From the above svd plot, it was found that the magnitude for the sad lips is higher than the magnitude of the angry lips.

CHAPTER 4

METHODOLOGY/ PROJECT WORK

4.1 Procedure Identification and Process Flow

The human mood recognition system will be developed using a set of defined procedures. The initial stage of the project involved extensive literature research on the topic of the project and its entire related subject. This involves performing research on the subject by various means such as books, journals, magazines and websites. Digital image processing and face emotion recognition are most important area that need a firm and comprehensive knowledge from the author. Therefore, the author is required to study all the important material regarding this subject. The system development procedure is shown in Figure 4.1.

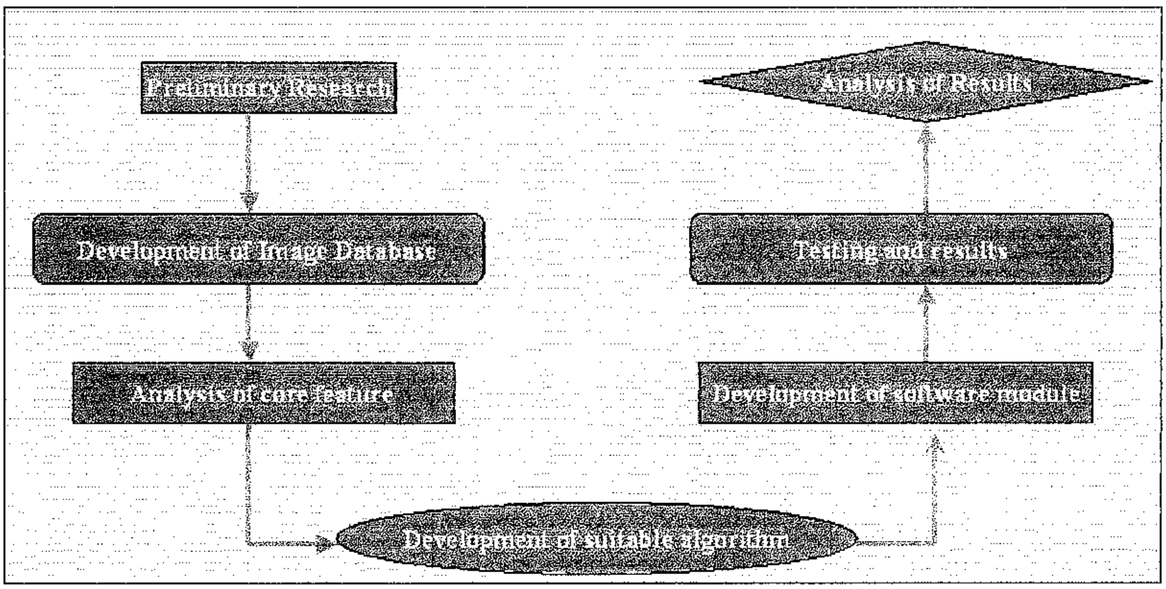

Figure 4.1: Block Diagram of system development procedure

The first step in Figure 4.1 is the preliminary findings, whereby it provides an important platform for the formative stages of the system to be based upon. For this stage, some ideas will be generated for the implementation of the next step. A table of properties has been developed in order to analyze the facial features of the angry, sad and happy mood. This is done by purely visual observation. Based on these findings, the author developed ideas on the mood recognition based on facial features using image-processing methods. Since there is not much research has been developed by the previous researchers and not published regarding the human mood recognition, the author had to formulate most of the recognition concepts herself. This preliminary finding forms an important platform for the conception of the next stage of the project.

Since this project is software based, knowledge of programming was essential in this project. Throughout this project, the algorithm that has been developed is processed by using MATLAB software. MATLAB software is a very powerful tool for mathematical calculation, visualization and programming. In addition there are several *toolboxes* available to expand the capabilities of MATLAB. The Image Processing Toolbox is one of these toolboxes. The toolbox consists of a set of functions and structures that handle image processing. This is good because it is not necessary to write code for all activation functions, training algorithms, etc. The following section will outline all the functions that have been build up so far by the author in order to create, train, test and finally visualize the emotion state of the input image.

The first step in developing algorithm is the acquisition of images. Figure 4.2 shows the specification of the camera that has been used:

Model: Konica Minolta

Resolution: 5 Mega pixel

Optical Zoom: 3.0

Size of the picture: 2048 x 1536

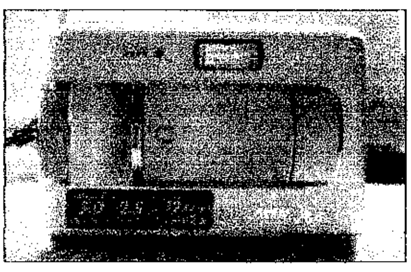

Figure 4.2: Digital Camera used and its specification

All the pictures taken are in the jpeg format. The system is developed to identify the state emotion of a person. Thus, the face images of happy, angry and sad people are needed. The face images of the people with various moods are obtained by means of an image acquisition device, which is digital camera, as shown in Figure 4.2. A sample set of 30 images; 10 for each set of emotion were acquired. These facial emotions include the full frontal facial emotion image excluding the body. The images taken are 2048 x 1536 pixel resolution in size and of jpeg format. To avoid uncertainty in the result, all the images are taken under similar lighting condition and frontal pose.

The next step after acquisition of image is the development of database. It is very important to create a useful image database before getting started with the coding because the database images will be used throughout the entire development and testing process.

Analysis of core feature is very crucial stage in the development of the system. The visible features that exhibit in the three moods are observed and identified, both the similarity and the difference. These features are essential since it will provide some idea to be used in the algorithm of the software module.

There are four algorithms that have been created for dissimilar region of interest, which consist of eye, forehead, lips region and part of the face. Selection of suitable methods is important because it entails the choice of appropriate steps to be taken to develop the program. The methods to identify the state emotion of a person is analyzed and selected. The different feature information to be extracted from the image is taken into consideration to derive the suitable algorithm. Algorithm to obtain the shape measurement, mean and coarseness were implemented in the software programming stage. This stage is significant, as the validity and the accuracy of the algorithms will determine the ultimate reliability of the system.

The next stage entails the development of the software modules. A software module is developed for each of the feature identification. For this system, four different features are used for the identification purpose, which consist of forehead, part of the face, eye and lips shape. It started with the preprocessing, which was done to crop parts of the images that is unnecessary and to classify the region of interest for processing. Apart from that, the images are filtered in order to remove noise and blur and to enhance contrast in order to produce a good result. Digital images are prone to a variety of noise. There are several ways that noise can be introduced into an image, depending on how the image is created. In this project, the noise might be produced when:

i. Since the image is acquired directly in digital format, the mechanism for gathering the data (such as CCD detector) can introduce noise
ii. Electronic transmission of image data can introduce noise.

Feature extraction method is an important stage in the development of the coding. At this stage, feature extraction is utilized to process the segmented area of the input image. The input image will be then identified to be fall under which category by classifying the each segmented region of interest. Feature extraction procedure is employed to extract shape and texture information of the input image. It was found that at closer inspection, these regions of interest would yield different result when system process different image at different emotion.

Throughout this project, the author has developed methods for the system to identify and also classify as well all the features using image processing and concepts and techniques available in MATLAB software. MATLAB software is chosen due to its versatility, popularity as well as its wide usage in the area of image processing. The software module is developed into four sections of codes to test the validity of the core algorithms needed for the recognition system. A number of test images are used in order to verify that the algorithm is generating reliable result.

The next step is the testing of the system results. When the software modules have been developed, testing is then conducted. The system is tested on its ability to identify the state emotion of an input image. The results obtained from testing by using a sample of 30 images are presented in the result section. Final analysis of the system is performed and considerations regarding some issues such as limitations as well as further improvements of the system are made.

CHAPTER 5
RESULT / DISCUSSION

5.1 Research and Analysis

Based on the analysis performed on the various moods, the author found that there is a similar facial expression characteristic that fall within the same mood. Facial expression characteristics are known as inherent features. Beside that, people will also exhibit their feelings in various ways, instead of facial expression, such as external features.

5.1.1 External Features

Typically, people will tend to express their feelings by body language or through conversation instead of facial expression only. In angry mood, people from survey done are expressing their feeling by yelling unpleasant word or terrible act, such as throwing anything nearby to someone that they are not satisfied with. Besides that, some people might show it just only by a deep sigh to show they are not satisfied towards something. In happy mood, people will exhibit their feelings in a manner of the gratitude toward another person who bring the happiness or by sharing the happiness story to other friend. Normally, for sad mood, people tend to cry or just be silent to anything as a way to escape from the surrounding and to release the sadness; also it might be the best medicine for some people.

5.1.2 Inherent Features

Inherent features relate to facial expression characteristics. Throughout this report, the author will discuss about the software module that has been developed in detail based on facial expression. For the inherent features, facial features are taken into consideration are the lip angle, roundness of the eye, coarseness of segmented part on image face and the mean of the forehead.

5.2 Proposed Human Mood Recognition System

The first stage of the work involved the acquisition and pre-processing of a dataset using a digital camera. In this project the author is using a measurement method for face classification system that identifies three basic emotions, given a face input. The emotions are happiness, sadness, and anger.

Face classification can be considered as face recognition in a loose sense and done by classifying the input image based on the result of region area. For the first semester, the author only focusing on a model, which defines as, authorized people, which a recognition system must accept to comparing without error. All the other people are unauthorized or 'aliens' and should be rejected.

While for the second semester, thirty images will be used in order to justify the previous coding and to calculate the error effectively. The face images must be taken under controlled environment parameters such as constant lighting levels; background and the model must be static. The width and height used for all images must be standard and the system proposed employs a 2048 x 1536 pixel size format. The system is only able to read jpeg file format images. Therefore, all the images must be in JPG format only. All the images that are used in the software are stored in the work folder in MATLAB software prior to running the coding in order for the software to read the image. The output will be pop-up in a window, easier for the user to understand and read the result.

Facial expression recognition deals with the classification of facial features into classes based on visual information. There are four type of software module that has been developed in the first semester. Thus, in the second semester, the author is required to analyze 30 images with different state of emotion, as well as to calculate the efficiency of the coding, which will be discussed in the next section.

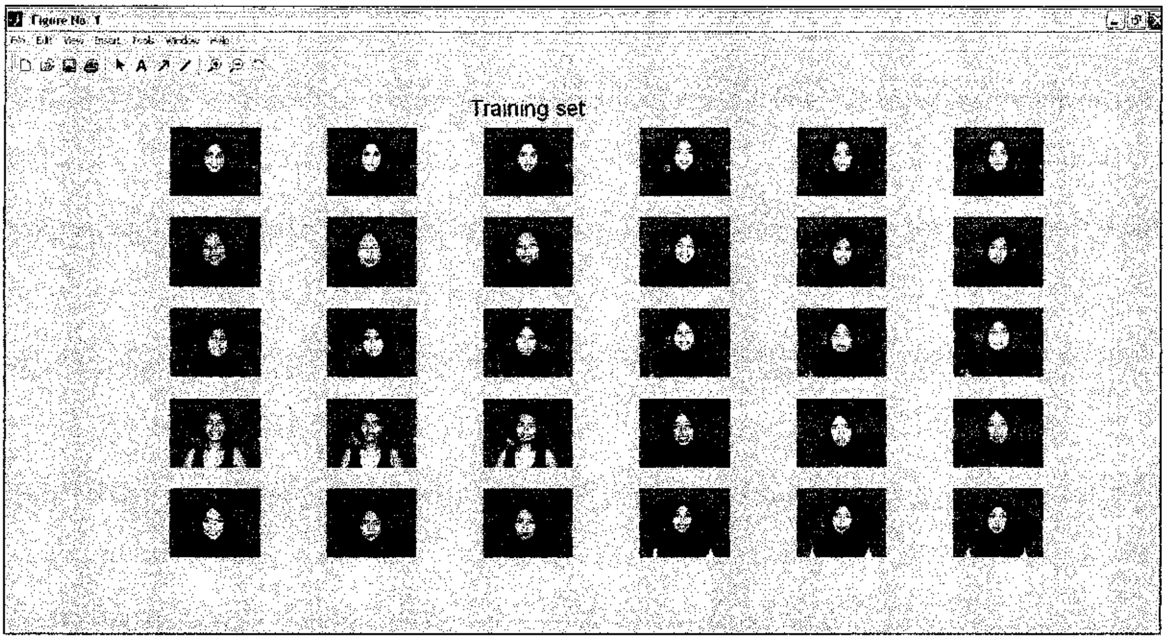

Figure 5.1: Training Set

Figure 5.1 shows the pictures of the training sets involved in analysis of this project. One model is exhibiting 3 moods, which consists of angry, sad and happy mood. Before starting the development of the software module, the criteria that are exhibit in the entire mood category is observed by the author, as stated in the table below.

Table 5.1: Inherent features

Mood / Characteristic	Angry	Sad	Happy
Frowning (forehead)	Yes	Yes	No
Size of Eyes	Biggest	Normal	Smallest
Mouth	Unvisible teeth	Unvisible teeth	Visible teeth

All the above criteria is important in order to analyze the picture and to develop the software coding, which will be used throughout this project.

5.2.1 Software Module: Identification by lips

This software module will extract the shape of the lips from the input image by using the angle calculation and perform the emotion classification and identification using this information. The flow procedure is shown as in Figure 5.2.

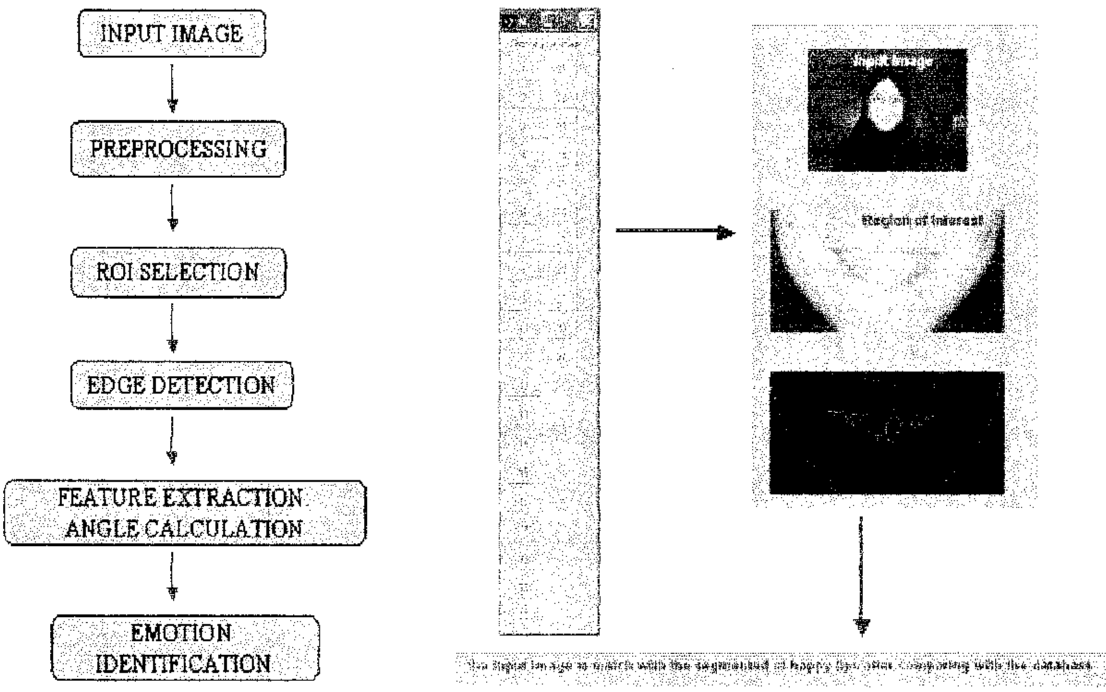

Figure 5.2: Lips Software Module Process Flow

Figure 5.2 is showing the flow process of the lips software module. The user is required to enter the desired input image, by clicking the menu table as shown in the upper left in figure above. If picture 1 is selected, the program will read the first image in the training set and cropping the input image into segmented area, which in this case is lips, and performing the next step, which is feature extraction. In the feature extraction method, the program will perform edge detection prior to the angle calculation. At the end of the program, it will classify the input image to be fallen under which category and the result will be displayed at the bottom of the output window.

However, from the result that has been obtained, it is quite varied from one model to another model, but yet the trend is still the same between each model, with the least magnitude of angle is the happy face. The result is shown as in the table 5.2.

Table 5.2: The angle value for identification by lips

Training Set No.	Emotion	Angle	Identification
1	Happy	14.5172	YES
2	Angry	17.8459	YES
3	Sad	31.7380	NO
4	Happy	16.9420	YES
5	Angry	55.8759	NO
6	Sad	33.7549	YES
7	Happy	12.5237	YES
8	Angry	43.9736	NO
9	Sad	50.8618	YES
10	Happy	15.7372	YES
11	Angry	49.4621	NO
12	Sad	22.0115	NO
13	Happy	11.3052	YES
14	Angry	14.8788	NO
15	Sad	67.8926	YES
16	Happy	15.9215	YES
17	Angry	11.4527	NO
18	Sad	29.6209	NO
19	Happy	15.1518	YES
20	Angry	23.5483	YES
21	Sad	58.6127	YES
22	Happy	13.6689	YES
23	Angry	39.5344	YES
24	Sad	61.2825	YES
25	Happy	58.6420	NO
26	Angry	20.2397	YES
27	Sad	51.9448	YES
28	Happy	14.6486	YES
29	Angry	44.5654	YES
30	Sad	71.4839	YES

(Please refer to Appendix B for the full coding)

Table 5.2 indicates the result after performing the coding. The criterion that is interested to look at is from the different in angle for different case of mood.

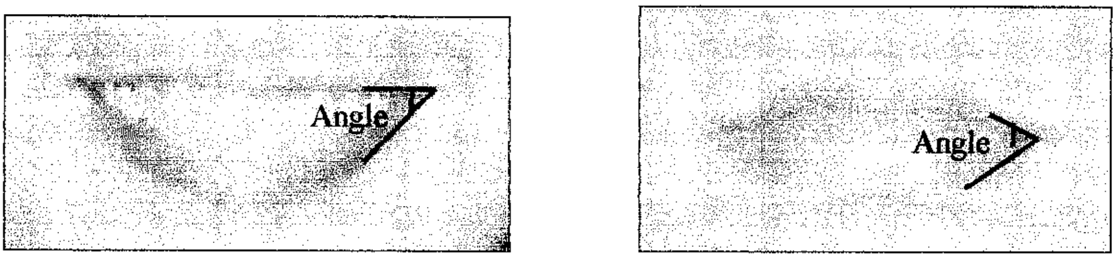

Figure 5.3: Angle increasing from happy lips to sad lips.

When happy, people lips tend to smaller and stretching, which will result in decrease in angle as shown in the Table 5.2, and it has been proved as shown in Figure 5.3. For the sad mood, some people might tend to pout their lips, resulting increase in value of angle of the lips. A low value of angle indicate happy mood, higher value indicate the angry mood, while the highest value indicate sad mood. It was found that the angle for each emotion fall within a specific range. Thus, each set of emotion has its own angle range. The specific range for each set of emotion can be identified and be programmed into the system. If the lips angle for an emotion falls into the programmed range, the system is then able to state the emotion of the input image.

For example, if the angle value falls within one of the specified range, then the range corresponding to the emotion will be decided as the emotion of the person. If the data fall into the happy mood range, the person will be identified as the happy mood.

From the Table 5.2, 9 out of 10 models for the happy lips yield the result of less than 17 for the angle of lips. Therefore, efficiency is said to be 90% for the happy mood, while for the angry and sad mood, both of them possess the efficiency of 60%.

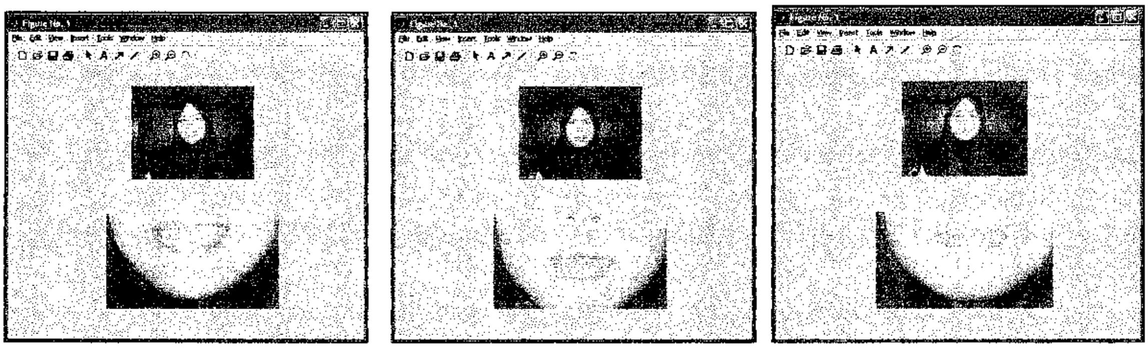

Figure 5.4: Happy, angry and sad lips region of interest, respectively

Figure 5.4 indicates the region of interest for lips. For angry and sad, the figure does not differed much, but these two categories can be differentiating by closer inspection to the region of forehead, which will be discussed in the next section.

5.2.2 Software Module: Identification by Eyes

The next software module that has been developed is the identification of eyes, whereby the region of interest is focusing on eye part. The flow process of the software module is shown in Figure 5.5.

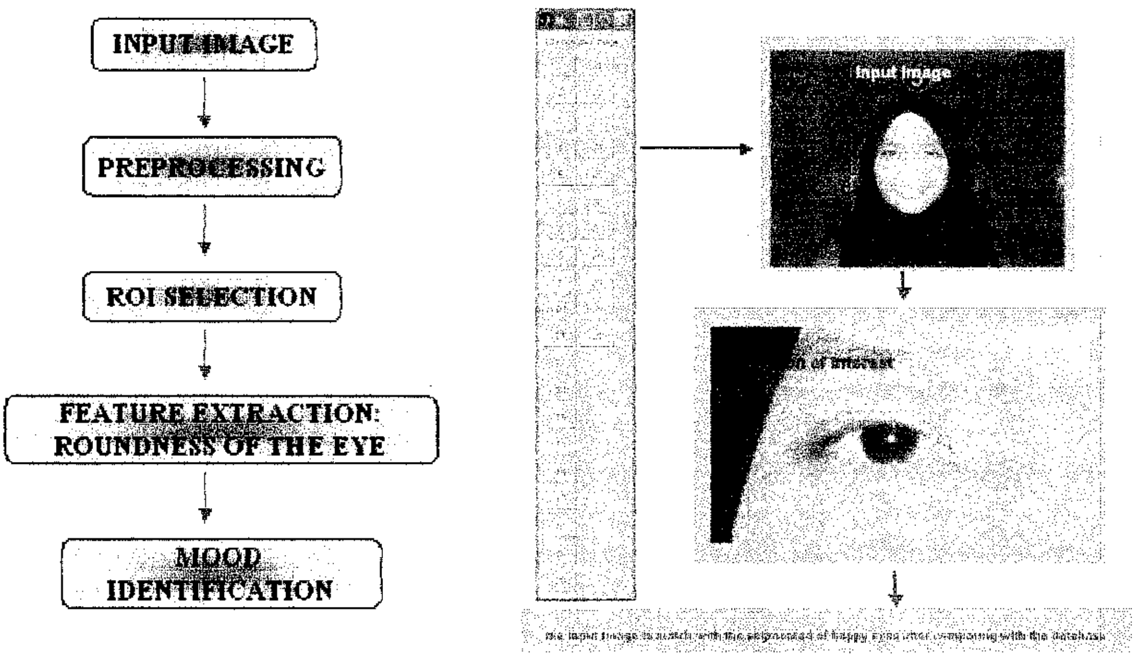

Figure 5.5: Eyes Software Module Process Flow

One of the models used is showing in the Figure 5.5. Firstly, the algorithm will read the input image, which is selected by the user, and it will perform the filtering and the selection of Region of Interest (ROI) for preprocessing. Then, it is followed by feature extraction, which will yield the result as displayed in the left side of Figure 5.5.

For the eye region, a shape analysis is done. From the analysis, it can be seen that people eyes will normally getting smaller when happy and getting bigger when angry. This analysis is taken as the core method of identification by the basis of eye shape. The roundness of the eyeball, as displayed in Figure 5.6 *(indicate by the red line)* is taken into account in order to calculate the roundness of the eye. If the value of roundness is close to 1, it is indicate that the object is approximately round.

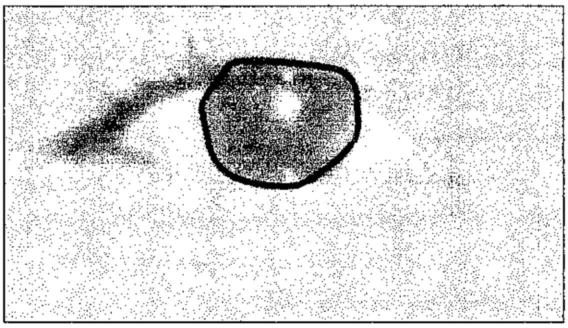

Figure 5.6: Roundness of the eyeball

Each set of emotion has a specific range of eye roundness value. If the value corresponds to the specific range of the emotion group, the state of the emotion will be identified. After performing the test and then analysis for the 30 sets of images (same as the models used for the extraction of lips region), the results are tabulated as in Table 5.3.

Table 5.3: Matric value (closer to 1 indicate the object is approximately round)

Training Set No.	Emotion	Matric	Identification
1	Happy	0.6187	YES
2	Angry	0.7425	YES
3	Sad	0.6586	YES
4	Happy	0.4123	NO
5	Angry	0.3398	NO
6	Sad	0.3648	YES
7	Happy	0.5273	YES
8	Angry	0.7343	YES
9	Sad	0.7007	YES
10	Happy	0.7746	NO
11	Angry	0.7459	YES
12	Sad	0.6735	YES
13	Happy	0.4745	NO
14	Angry	0.4547	YES
15	Sad	0.5397	NO
16	Happy	0.7088	NO
17	Angry	0.7239	YES
18	Sad	0.7349	NO
19	Happy	0.3213	YES
20	Angry	0.3234	NO
21	Sad	0.3229	YES
22	Happy	0.2940	YES
23	Angry	0.5074	YES
24	Sad	0.3204	NO
25	Happy	0.6433	YES
26	Angry	0.6922	NO
27	Sad	0.6226	NO
28	Happy	0.7359	NO
29	Angry	0.7471	YES
30	Sad	0.6795	YES

(Please refer to Appendix B for the full coding)

From the observation and after performing the analysis, it was observed that most of the eyes would exhibit a wider eye when getting angry and getting smaller when sad, and smallest when happy, as shown in Figure 5.7. Since the models used might possess a small eye (might be a mix person, such as Chinese) and some possess bigger eyes (most Malay should have), the result as tabulated in Table 5.3 is divided into two groups. But yet, the trend are still the same with angry eyes will be biggest between the other two moods.

Figure 5.7: Crop eyes for happy, angry and sad respectively.

After performing the analysis for all the three moods with the 10 models, the author note that 50% of the images are correct for happy mood, 70% are correct for angry mood while another 60% for the sad mood. Therefore, it is safe to say that, efficiency of the coding is 60%, whereby 18 out of 30 images are correctly identified.

5.2.3 Software Module: Identification by Forehead

The third software module that has been developed is the identification by forehead. This software module flow process is displayed as in Figure 5.8. As can be seen, the procedure involved is quite the same as previous software module, except for the region of interest, which in this case is forehead region.

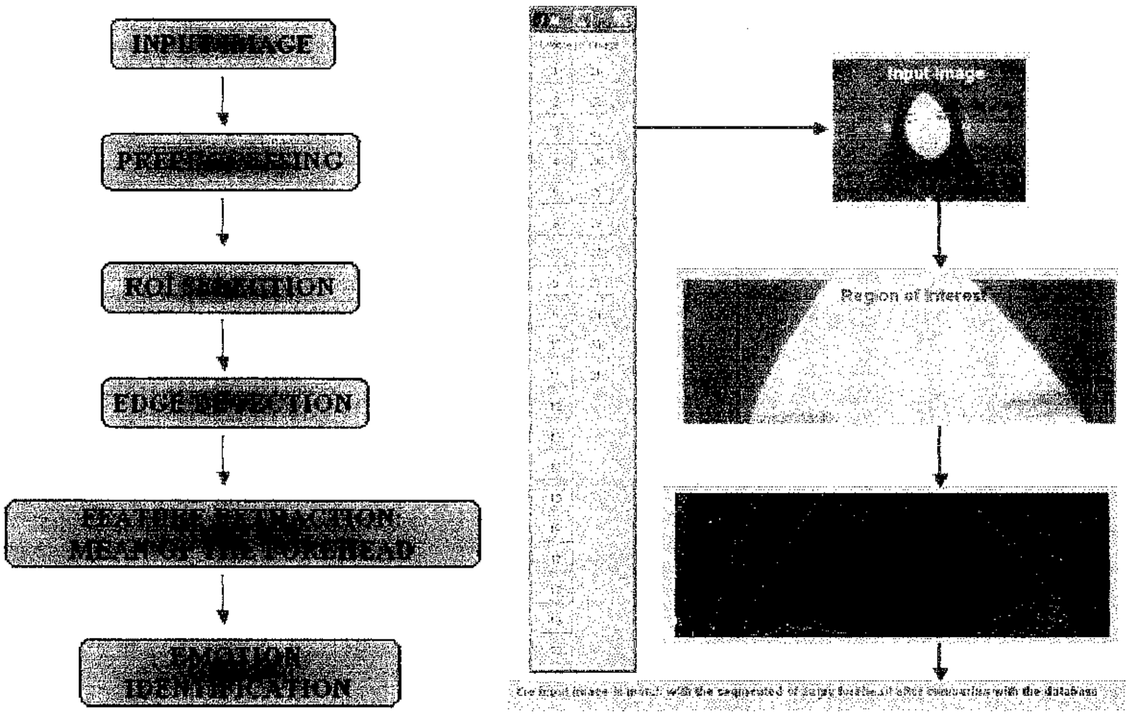

Figure 5.8: Forehead Software Module Process Flow

The input image entered by the user is segmented into the region of interest, which in this case is the forehead part. Then, the algorithm will detect the edge of the segmented area. For feature extraction method, the algorithm that has been build will calculate the mean of the image. Mean is defined as the estimation of the value around which the central clustering occurs.

When a set of values has a sufficiently strong central tendency, that is, a tendency to cluster around some particular value, then it may be useful to characterize the set by a few numbers that are related to its moments, the sums of integer powers of the values, and the best known is the mean. The result after analyzing the mean of the thirty images are tabulated as in Table 5.4.

Table 5.4: Mean Result after analysis of 30 images

Training Set No.	Emotion	Mean	Identification
1	Happy	124	NO
2	Angry	118	YES
3	Sad	130	NO
4	Happy	93	NO
5	Angry	94	YES
6	Sad	80	YES
7	Happy	116	YES
8	Angry	155	YES
9	Sad	113	YES
10	Happy	73	YES
11	Angry	118	NO
12	Sad	109	NO
13	Happy	102	YES
14	Angry	143	NO
15	Sad	187	YES
16	Happy	125	YES
17	Angry	137	YES
18	Sad	109	NO
19	Happy	86	NO
20	Angry	81	NO
21	Sad	72	NO
22	Happy	69	YES
23	Angry	104	YES
24	Sad	112	NO
25	Happy	116	YES
26	Angry	103	NO
27	Sad	141	NO
28	Happy	84	NO
29	Angry	124	YES
30	Sad	139	YES

(Please refer to Appendix B for the full coding)

Table 5.4 indicate the result after performing the analysis for the 10 models with three different moods. From the result above, a set of range has been developed to classify the input image to fall under which category, whether angry, sad or happy. The mean for the angry mood is set up to be the highest since when angry people tend to be frowning, which is increasing the coarseness of the forehead, which is directly related to the mean of the image. Happy is classified as the lowest mean, since the forehead for happy mood tend to be stretching, which decrease the value around which the central clustering occurs. The middle class range of mean for the forehead ROI is the sad mood. Some people tend to frown a little when sad mood, but not as much as when in angry mood. Therefore, it is set that the highest mean occurred when people are angry.

After analyzing the entire set of database, it has been found out that 6 out of 10 images are correct for the angry mood and happy mood, while for the sad mood, only 4 out of 10 are correct. Therefore the efficiency of the coding is 53.3%.

5.2.4 Software Module: Identification by Part of the Face

This software module will identify a person's emotion by the basis of coarseness of part of the face. Part of the face are consists of forehead, eyes, cheek and nose. The flow process of this software module is displayed as in Figure 5.9. Firstly, the input image is going for preprocessing, then segmented into the region of interest before the algorithm detects the edge of the segmented area. The feature extraction method that are applied in this software module is basically to determine the coarseness of the segmented area. Coarseness is defined as the roughness or looseness in texture or the texture is lacking in refinement.

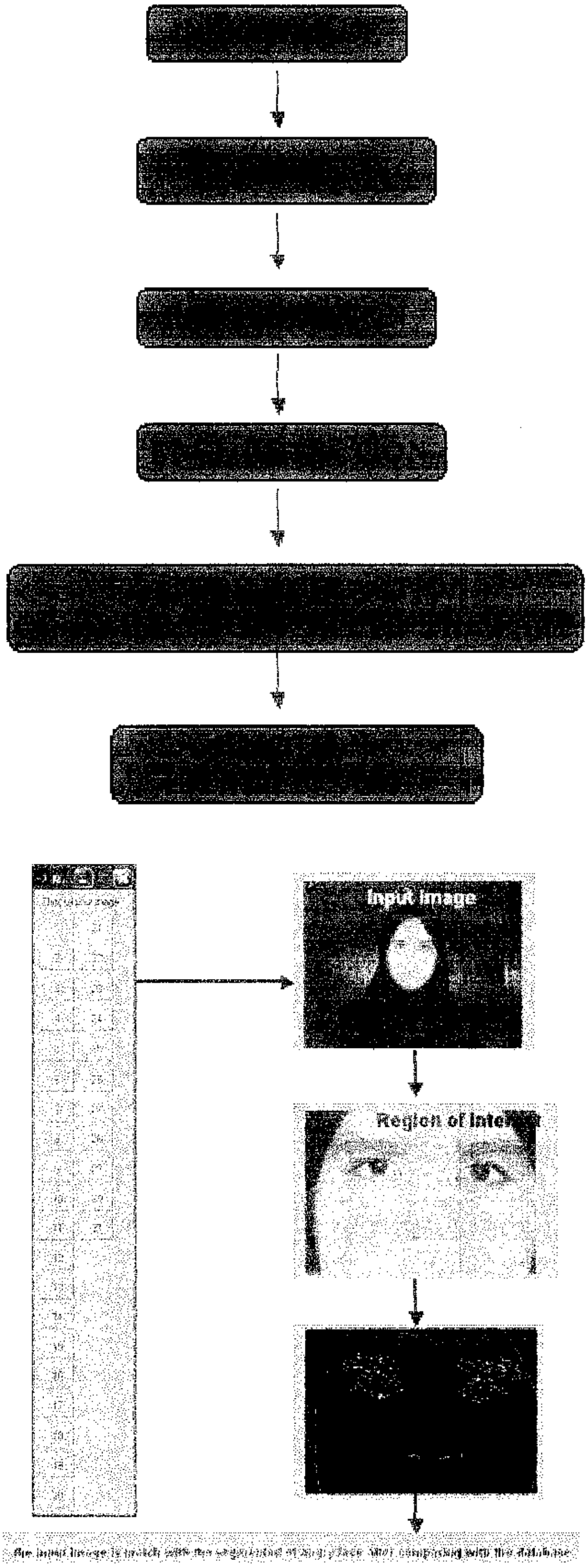

Figure 5.9: Part of the Face Software Module Process Flow

After analyzing the thirty sets of images, the result are tabulated as in Table 5.5.

Table 5.5: Coarseness Result after analysis of 30 images

Training Set No.	Emotion	Coarseness	Identification
1	Happy	0.99971347	YES
2	Angry	0.99974094	NO
3	Sad	0.99972545	NO
4	Happy	0.99979055	YES
5	Angry	0.99973853	NO
6	Sad	0.99977654	NO
7	Happy	0.99958602	NO
8	Angry	0.9995873	YES
9	Sad	0.99959994	NO
10	Happy	0.99975324	YES
11	Angry	0.99963093	YES
12	Sad	0.99969939	YES
13	Happy	0.99973118	YES
14	Angry	0.99975282	YES
15	Sad	0.99971193	YES
16	Happy	0.99971684	YES
17	Angry	0.99967469	NO
18	Sad	0.99965921	YES
19	Happy	0.99959641	NO
20	Angry	0.99962313	YES
21	Sad	0.9996801	YES
22	Happy	0.99967518	NO
23	Angry	0.99962843	YES
24	Sad	0.99961281	NO
25	Happy	0.99968477	NO
26	Angry	0.99956737	YES
27	Sad	0.99967298	YES
28	Happy	0.99978695	YES
29	Angry	0.99976926	NO
30	Sad	0.99977971	NO

(Please refer to Appendix B for the full coding)

Table 5.5 indicate the result after performing the analysis for the 10 models with three different moods. From the result above, a set of range has been developed to classify the input image to fall under which category, whether angry, sad or happy. The coarseness for the angry mood is set up to be the highest since when angry people tend to be frowning, which is increasing the coarseness of the forehead, which is directly related to the mean of the image. Happy is classified as the lowest coarseness, since when happy the forehead tend to be stretching as well as the cheek, while for the middle class of coarseness is the sad mood. Some people tend to frown a little when sad mood, but not as much as when in angry mood.

After analyzing the entire set of database, it has been found out that 6 out of 10 images are correct for the angry mood, while for the sad and happy mood, only 5 out of 10 are correct (as shown in Table 5.5). Therefore the total efficiency of the coding is 53.3%, since 16 out of 30 are correctly identified.

5.3 Discussion

Testing performed on the system was able to give the author an understanding of the system performance. The system is divided into four software modules and each of the software's performance was analyzed by the author. In an overall view, the system was able to give reliable output.

For the first software module, that is identification by lips region, an efficiency of 70% was obtained. Although there were several face emotion images that could not be identified correctly, the wrong percentage are yet still small, whereby only 9 out of 30 are incorrectly identified compared to 21 out of 30 are correctly identified.

The same goes to identification by forehead, eyes and part of the face. Eye module yield an efficiency of 60% while both the forehead module and part of the face module had an efficiency of 53.3%.

One of the factors that cause incorrect output is due to the handshaking of the photographer, since the photographer used is not the expert one. Besides that, the models used also sometimes not static. Thus, some images appear to be closer or far away from the photographer. The exact distance criteria that should be followed are disobeyed sometimes when taking the pictures. Another reason is that due to the lighting. Actually, two types of lighting used when capturing the thirty sets of images. Thus, some images appear to be brighter while some darker. The inconsistency in lighting may have some discrepancies in the results. This is because these software modules depend heavily on color information.

Generally, incorrect identifications might also be caused due to the fact that sometimes both the moods exhibits the same criteria, such as when angry and sad mood. Some people might pout when angry, and some might pout when sad. The same things happen when some people tend to be frowning in both angry and sad mood. Thus, this causes some discrepancies and inconsistencies in the results.

Overall, the most reliable software module is the identification by lips region, followed by eye region. This is because most of the people tend to smile when happy, and it provides clear distinction between happy, angry and sad mood. Thus, this software is very reliable. The eye region is reliable to differentiate the three categories of emotions since the roundness of eyes for happy, angry and sad mood are quite different and this features can be easily employed in programming the software.

The identification by forehead and part of the face is a bit lower in performance when compared to the other two-software module. This is because, although the coarseness and the mean can be visibly see through naked eyes, especially forehead part, but this observation is relatively difficult to implement in computational basis.

Problems were faced in obtaining images with similar lighting conditions and within exact distance between the photographer and models, as there was not enough equipment available to ensure this condition, such as no tripod to be used to hold the camera and most important things is that it is almost not possible to make the models in exact position all the time.

5.4 Limitations

Currently, the proposed human mood recognition system still has a number of limitations that has to overcome. Admittedly, the recognition system would not perform very desirably in the real world system because the efficiency rate would be much lower that the efficiency by using the test images. This is because the test images that have been chosen basically possess typical features for each type of emotion group. Limitations of the system are as below:

i. The system is able to recognize faces of single view frontal pose only. The inability of the system to recognize other poses such as side view or profile view is due to the fact that the author has yet to develop the algorithms perform such operation.

ii. The system is unable to handle face images of different sizes and formats. Currently the user has to manually ensure that the face images are of similar size and of the bitmap format.

iii. System is unable to handle variable parameters of the camera and the surrounding environment. Currently, the test images must be taken under very similar lighting conditions and similar background with no extra details.

CHAPTER 6
SYSTEM IMPLEMENTATION

This entire project is a software-based endeavor. Therefore, the author is required to implement a system that are easier for the user to understand and work with. Simple Graphical User Interface (GUI) is developed in this project. GUIs provide a means through which individuals can communicate with the computer without programming commands. The components in the MATLAB have become quite standardized and developed into a user friendly and intuitive set of tools. These tools can be used to increase the productivity of a user or to provide a window to the sophistication and power of a MATLAB application for people with little or no MATLAB programming experience [8].

This section is intended to familiarize the reader with the software that has been developed in this project. The user is required to choose the image by clicking the figure menu, as shown in Figure 6.1.

If the user are trying to analyze their own image, instead of the training set, the user are required to save the image in the Matlab work folder. The input face image must be of jpeg file format, and the image size must be in 2048 x 1536. The result might be differed when using different size of image. Then, the user are required to click the number 31 if the intend to analyze their own set of image, since number 1 until 30 is reserved for the training set. The number 31 is reserved for new image, instead of the training set. The result displayed below is the result after running the image that is not in the training set.

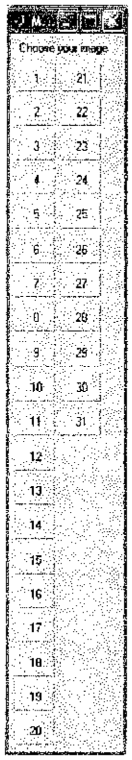

Figure 6.1: Human Mood Recognition System

The user is able to select any of the input images in the training set by clicking the corresponding number. When the selection is made, the screen will automatically go to the required input image.

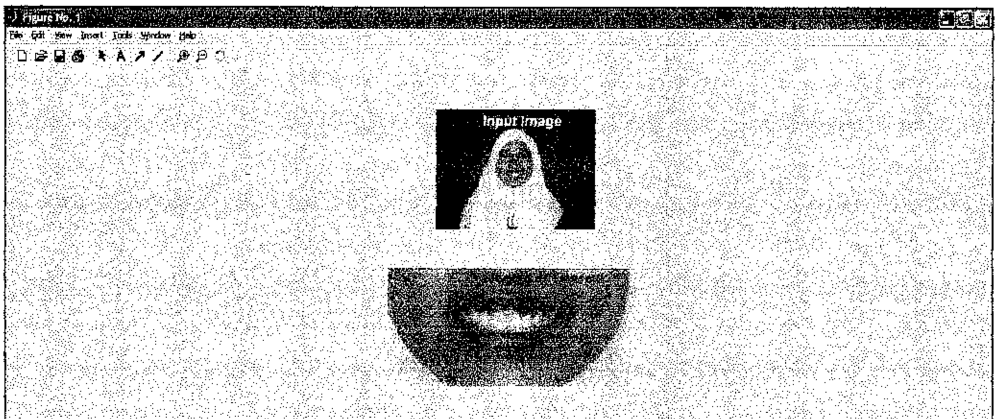

Figure 6.2: Lips Region Software Module

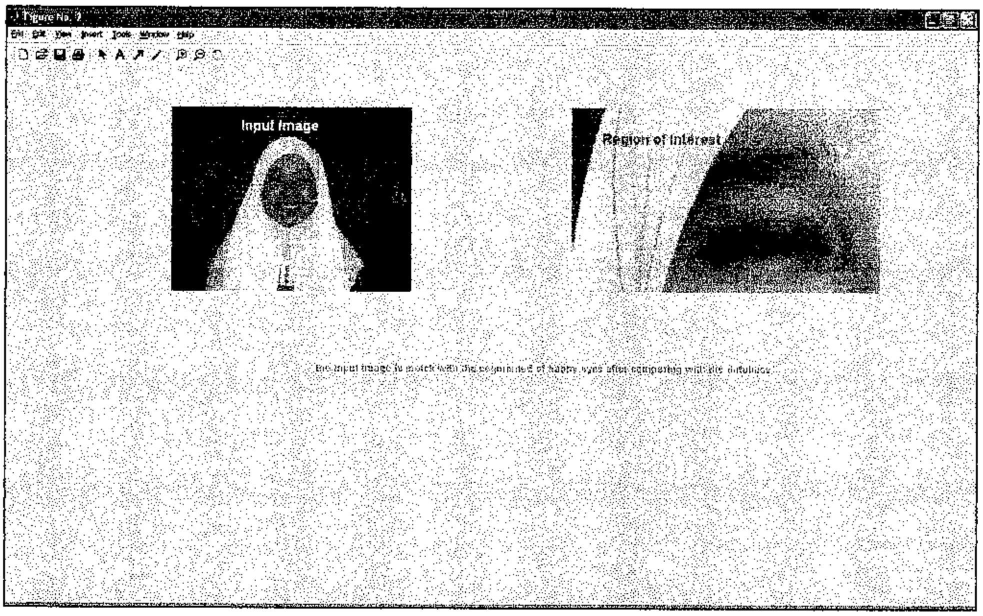

Figure 6.3: Eye Region Software Module

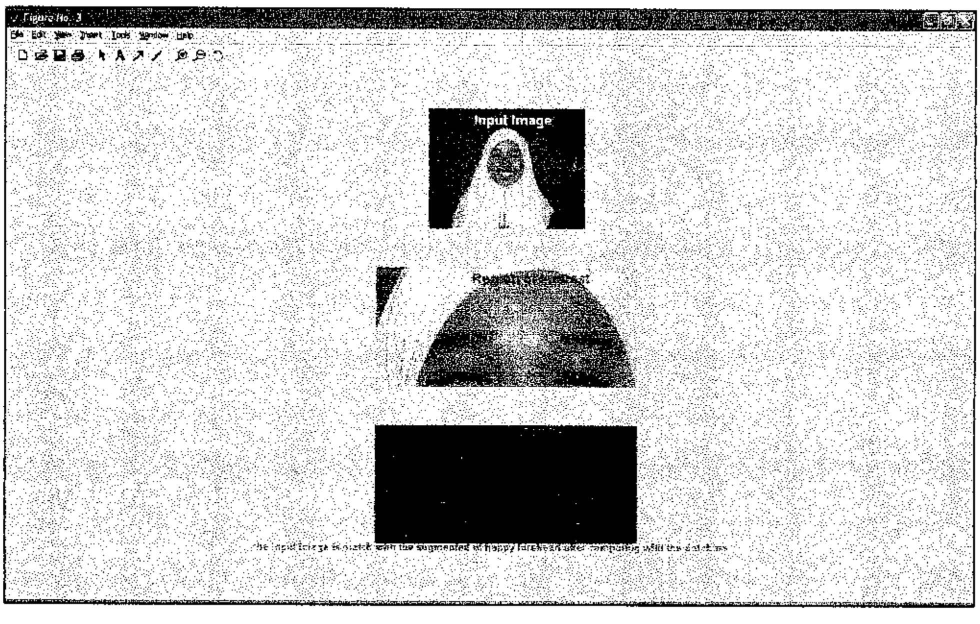

Figure 6.4: Forehead Region Software Module

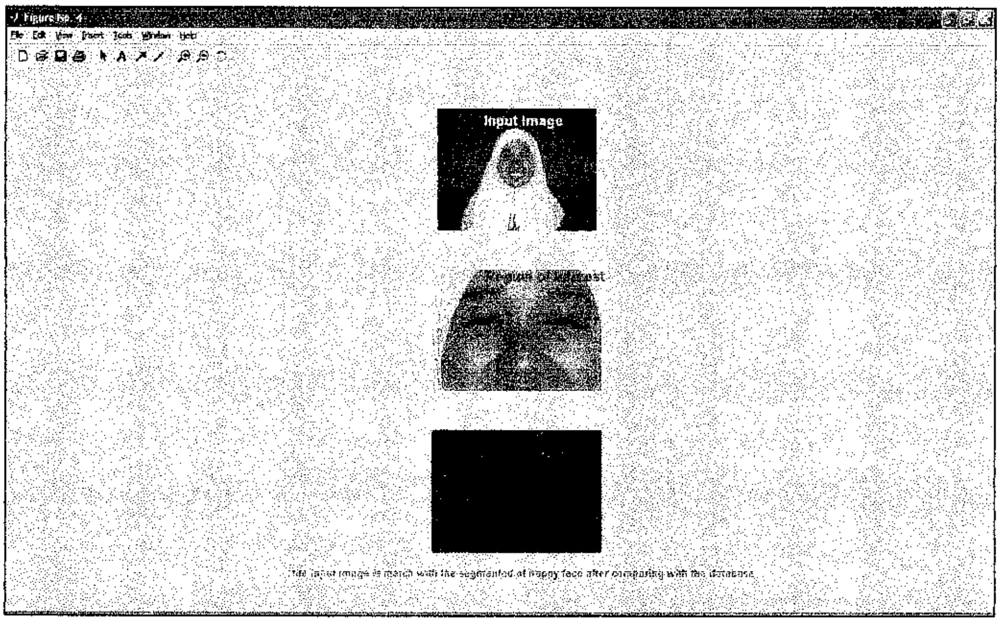

Figure 6.5: Part of the Face Region Software Module

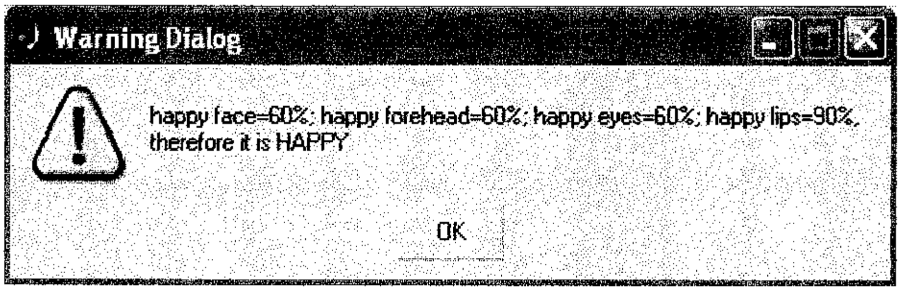

Figure 6.6: Identification Result

Figure 6.6 is the total classification of the input image. Since the entire four software modules indicate the happy mood, therefore the input image is classified as happy mood.

CHAPTER 7

CONCLUSION / RECOMMENDATION

7.1 Conclusion

The author has successfully developed a human mood recognition system that would be able to identify, in general, the state emotion of happy, angry and sad mood. It is conducted basically to determine the relationship between facial expressions with the human mood. It can be used in various purposes, such as logging facial expression during testing of new products or for retailing whereby the system developed will be recognizing customers to monitor their preferences. This project requires lots of knowledge in Digital Image Processing and MATLAB software. This system employs feature extraction technique to obtain information on a few salient facial features, which are the lips changes, roundness of the eyes, mean of the forehead and the coarseness of the segmented part of the face. It has been observed that these facial features provide discriminatory information about the differences in facial features for different moods. This system is developed in several stages. Up to now, software that will be able to recognize state emotion by the basis of inherent features has been developed. A set of 30 images has been used for analysis and testing purposes. From the testing, it can be concluded that moderate but consistent recognition rates have been achieved in this project. The system has a possible level of accuracy.

7.2 Recommendations

The accuracy of the system can be improved by using high-level image processing techniques and better image recognition technique. This is important to the system because image processing and image recognition module does the brainwork of the system.

As for now, the system is only able to recognize the frontal pose and within some distance between the photographer and model. Hence, the recognition system can be improved by capturing the model with side view in order to identify the mood. Besides that, the system that has been developed so far can only recognized person by capturing the image by using digital camera. For the future, this project can be improved by using the real-time technique, which is by using web cam. Furthermore, this project can also be enhanced by adding more interesting and user friendly Graphical User Interface to attract the audience to use this system.

If the future student can implement all the above recommendation, this project will become more interesting and usefully human mood recognition project.

REFERENCES

[1] R.Brunelli, T.Poggio, Face Recognition: Features versus Templates, IEEE Trans. Pattern Anal. Mach. Intell. 15 (10) (1993) 1042-1052.

[2] Kenji Tamural, Shohei Kat, Takanori Yamakita, Kenji Kimuradand Hidenori Itoh, A System of Dialogical Mental Health Care with Sensibility Technology Communication Robot, International Symposium on Micromechatronics and Human Science, 2003, pp 67-71.

[3] Rafael C. Gonzales and Richard E. Woods. Digital Image Processing: Prentice Hall Inc, USA, 2002, pp.672-674.

[4] De Silva L.C and Suen Chun Hui. Real Time Facial Feature Extraction and Emotion. Proceedings of the 2003 Joint Conference of the Fourth International Recognition. Information, Communication and Signal Processing, 2003 and the Fourth Pacific Rim Conference on Multimedia, vol.3, 15-18 Dec 2003, pp 1310-1314.

[5] Anzai Y.(1992), Pattern Recognition and Machine Learning, Academic Press

[6] B. Fasel, Juergen Luettin, *Automatic Facial Expression Analysis: A Survey,* Journal Of The Pattern Recognition Society, accepted 15 February 2002.

[7] Mood Detection, http://cnx.rice.edu/content

[8] Patrick Marchand, Graphics and GUIs with MATLAB, CRC Press, 1996, pp 207

APPENDICES

[1] APPENDIX A: Gantt Chart for FYP I and FYP II

[2] APPENDIX B: Full Coding

APPENDIX A

Appendix A: Gantt Chart for Final Year Project 1

No.	Detail / Week	1	2	3	4	5	6	7	8	9	10	11	12	13
1	Selection of Project Topics	■	■											
2	Preliminary Research Work			■	■									
3	Submission of Preliminary Report				■									
4	Brainstorming and Planning					■	■							
5	Research, Literature and Program Learning					■	■	■	■	■	■	■	■	■
6	Submission of Progress Report							■						
7	Training and Testing the System							■	■	■	■	■	■	■
8	Submission of Final Draft											■		
9	Submission of Interim Report												■	
10	Oral Presentation													■

Appendix A: Gantt Chart for Final Year Project II

No	Detail/Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	Project Work	■	■	■	■	■	■	■	■	■	■	■	■								
	- Practical Work																				
2	Submission of Progress Report 1			■																	
3	Submission of Progress Report 2							■													
4	Submission of Draft Report												■								
5	Submission of Final Report (soft cover)														■						
6	Oral Presentation																		■		
7	Submission of Technical Report																			■	
8	Submission of Final Report (hard cover)																				■

APPENDIX B

```
all

all

g off MATLAB:polyfit:RepeatedPointsOrRescale
g off MATLAB:singularMatrix

elect = menu ('Choose your image', '1','2','3','4','5',...
','8','9','10',...
12','13','14','15',...
17','18','19','20',...
22','23','24','25',...
27','28','29','30','31');

r_select == 1
= imread ('1.jpg');
uth_start_row = 755;
uth_start_col = 770;
opA = A(mouth_start_row:1000, mouth_start_col:1250, :);
ycrop=cropA;
gure,subplot (3,1,1);imshow(A);text(600,150,'Input Image','Color','y',...
 'FontSize',14,'FontWeight','bold');
oplot (3,1,2);imshow (cropA);text(190,20,'Region of Interest','Color','b',...
 'FontSize',14,'FontWeight','bold');
 user_select==2
imread('2.jpg');
uth_start_row = 755;
uth_start_col = 770;
opB = B(mouth_start_row:1000, mouth_start_col:1250, :);
  trycrop=cropB;
  figure,subplot (3,1,1);imshow(B);text(600,150,'Input Image','Color','y',...
 'FontSize',14,'FontWeight','bold');
oplot (3,1,2);imshow (cropB);text(190,20,'Region of Interest','Color','b',...
 'FontSize',14,'FontWeight','bold');
 user_select==3
mread('3.jpg');
uth_start_row = 755;
uth_start_col = 770;
opC = C(mouth_start_row:1000, mouth_start_col:1250, :);
  trycrop=cropC;
  figure,subplot (3,1,1);imshow(C);text(600,150,'Input Image','Color','y',...
 'FontSize',14,'FontWeight','bold');
oplot (3,1,2);imshow (cropC);text(190,20,'Region of Interest','Color','b',...
 'FontSize',14,'FontWeight','bold');
 user_select==4
mread('4.jpg');
uth_start_row = 725;
uth_start_col = 775;
opD = D(mouth_start_row:900, mouth_start_col:1250, :);
  trycrop=cropD;
  figure,subplot (3,1,1);imshow(D);text(600,150,'Input Image','Color','y',...
 'FontSize',14,'FontWeight','bold');
oplot (3,1,2);imshow (cropD);text(190,20,'Region of Interest','Color','b',...
 'FontSize',14,'FontWeight','bold');
 user_select==5
mread('5.jpg');
```

```
ıth_start_row = 725;
ıth_start_col = 775;
ɔpE = E(mouth_start_row:900, mouth_start_col:1250, :);
  trycrop=cropE;
  figure,subplot (3,1,1);imshow(E);text(600,150,'Input Image','Color','y',...
 'FontSize',14,'FontWeight','bold');
ɔplot (3,1,2);imshow (cropE);text(190,20,'Region of Interest','Color','b',...
 'FontSize',14,'FontWeight','bold');
 user_select==6
imread('6.jpg');
ıth_start_row = 725;
ıth_start_col = 775;
ɔpF = F(mouth_start_row:900, mouth_start_col:1250, :);
  trycrop=cropF;
  figure,subplot (3,1,1);imshow(F);text(600,150,'Input Image','Color','y',...
 'FontSize',14,'FontWeight','bold');
ɔplot (3,1,2);imshow (cropF);text(190,20,'Region of Interest','Color','b',...
 'FontSize',14,'FontWeight','bold');
 user_select==7
.mread('7.jpg');
  mouth_start_row = 820;
ıth_start_col = 710;
ɔpG = G(mouth_start_row:1110, mouth_start_col:1250, :);
  trycrop=cropG;
  figure,subplot (3,1,1);imshow(G);text(600,150,'Input Image','Color','y',...
 'FontSize',14,'FontWeight','bold');
ɔplot (3,1,2);imshow (cropG);text(190,20,'Region of Interest','Color','b',...
 'FontSize',14,'FontWeight','bold');
 user_select==8
.mread('8.jpg');
  mouth_start_row = 820;
ıth_start_col = 710;
ɔpH = H(mouth_start_row:1110, mouth_start_col:1250, :);
  trycrop=cropH;
  figure,subplot (3,1,1);imshow(H);text(600,150,'Input Image','Color','y',...
 'FontSize',14,'FontWeight','bold');
ɔplot (3,1,2);imshow (cropH);text(190,20,'Region of Interest','Color','b',...
 'FontSize',14,'FontWeight','bold');
 user_select==9
.mread('9.jpg');
  mouth_start_row = 820;
ıth_start_col = 710;
ɔpI = I(mouth_start_row:1110, mouth_start_col:1250, :);
  trycrop=cropI;
  figure,subplot (3,1,1);imshow(I);text(600,150,'Input Image','Color','y',...
 'FontSize',14,'FontWeight','bold');
ɔplot (3,1,2);imshow (cropI);text(190,20,'Region of Interest','Color','b',...
 'FontSize',14,'FontWeight','bold');
 user_select==10
.mread('10.jpg');
  mouth_start_row = 790;
ıth_start_col = 774;
ɔpJ = J(mouth_start_row:1000, mouth_start_col:1220, :);
  trycrop=cropJ;
  figure,subplot (3,1,1);imshow(J);text(600,150,'Input Image','Color','y',...
 'FontSize',14,'FontWeight','bold');
ɔplot (3,1,2);imshow (cropJ);text(190,20,'Region of Interest','Color','b',...
```

```
'FontSize',14,'FontWeight','bold');
user_select==11
imread('11.jpg');
  mouth_start_row = 790;
ıth_start_col = 774;
ɔpK = K(mouth_start_row:1000, mouth_start_col:1220, :);
  trycrop=cropK;
  figure,subplot (3,1,1);imshow(K);text(600,150,'Input Image','Color','y',...
'FontSize',14,'FontWeight','bold');
ɔplot (3,1,2);imshow (cropK);text(190,20,'Region of Interest','Color','b',...
'FontSize',14,'FontWeight','bold');
user_select==12
imread('12.jpg');
      mouth_start_row = 790;
ıth_start_col = 774;
ɔpL = L(mouth_start_row:1000, mouth_start_col:1220, :);
  trycrop=cropL;
  figure,subplot (3,1,1);imshow(L);text(600,150,'Input Image','Color','y',...
'FontSize',14,'FontWeight','bold');
ɔplot (3,1,2);imshow (cropL);text(190,20,'Region of Interest','Color','b',...
'FontSize',14,'FontWeight','bold');
user_select==13
imread('13.jpg');
  mouth_start_row = 850;
ıth_start_col = 830;
ɔpM = M(mouth_start_row:1120, mouth_start_col:1280, :);
  trycrop=cropM;
  figure,subplot (3,1,1);imshow(M);text(600,150,'Input Image','Color','y',...
'FontSize',14,'FontWeight','bold');
ɔplot (3,1,2);imshow (cropM);text(190,20,'Region of Interest','Color','b',...
'FontSize',14,'FontWeight','bold');
user_select==14
imread('14.jpg');
  mouth_start_row = 850;
ıth_start_col = 830;
ɔpN = N(mouth_start_row:1120, mouth_start_col:1280, :);
  trycrop=cropN;
  figure,subplot (3,1,1);imshow(N);text(600,150,'Input Image','Color','y',...
'FontSize',14,'FontWeight','bold');
ɔplot (3,1,2);imshow (cropN);text(190,20,'Region of Interest','Color','b',...
'FontSize',14,'FontWeight','bold');
user_select==15
imread('15.jpg');
  mouth_start_row = 850;
ıth_start_col = 830;
ɔpO = O(mouth_start_row:1120, mouth_start_col:1280, :);
  trycrop=cropO;
  figure,subplot (3,1,1);imshow(O);text(600,150,'Input Image','Color','y',...
'FontSize',14,'FontWeight','bold');
ɔplot (3,1,2);imshow (cropO);text(190,20,'Region of Interest','Color','b',...
'FontSize',14,'FontWeight','bold');
user_select==16
imread('16.jpg');
  mouth_start_row = 735;
ıth_start_col = 770;
ɔpP = P(mouth_start_row:990, mouth_start_col:1250, :);
  trycrop=cropP;
```

```
  figure,subplot (3,1,1);imshow(P);text(600,150,'Input Image','Color','y',...
 'FontSize',14,'FontWeight','bold');
bplot (3,1,2);imshow (cropP);text(190,20,'Region of Interest','Color','b',...
 'FontSize',14,'FontWeight','bold');
 user_select==17
imread('17.jpg');
  mouth_start_row = 735;
uth_start_col = 770;
ɔpQ = Q(mouth_start_row:990, mouth_start_col:1250, :);
  trycrop=cropQ;
  figure,subplot (3,1,1);imshow(Q);text(600,150,'Input Image','Color','y',...
 'FontSize',14,'FontWeight','bold');
ɔplot (3,1,2);imshow (cropQ);text(190,20,'Region of Interest','Color','b',...
 'FontSize',14,'FontWeight','bold');
 user_select==18
imread('18.jpg');
  mouth_start_row = 735;
ɹth_start_col = 770;
ɔpR = R(mouth_start_row:990, mouth_start_col:1250, :);
  trycrop=cropR;
  figure,subplot (3,1,1);imshow(R);text(600,150,'Input Image','Color','y',...
 'FontSize',14,'FontWeight','bold');
ɔplot (3,1,2);imshow (cropR);text(190,20,'Region of Interest','Color','b',...
 'FontSize',14,'FontWeight','bold');
 user_select==19
imread('19.jpg');
ɹth_start_row = 700;
ɹth_start_col = 770;
ɔpS= S(mouth_start_row:980, mouth_start_col:1250, :);
  trycrop=cropS;
  figure,subplot (3,1,1);imshow(S);text(600,150,'Input Image','Color','y',...
 'FontSize',14,'FontWeight','bold');
ɔplot (3,1,2);imshow (cropS);text(190,20,'Region of Interest','Color','b',...
 'FontSize',14,'FontWeight','bold');
 user_select==20
imread('20.jpg');
  mouth_start_row = 700;
ɹth_start_col = 770;
ɔpT= T(mouth_start_row:980, mouth_start_col:1250, :);
  trycrop=cropT;
  figure,subplot (3,1,1);imshow(T);text(600,150,'Input Image','Color','y',...
 'FontSize',14,'FontWeight','bold');
ɔplot (3,1,2);imshow (cropT);text(190,20,'Region of Interest','Color','b',...
 'FontSize',14,'FontWeight','bold');
 user_select == 21
= imread ('21.jpg');
      mouth_start_row = 700;
ɹth_start_col = 770;
ɔpU= U(mouth_start_row:980, mouth_start_col:1250, :);
  trycrop=cropU;
  figure,subplot (3,1,1);imshow(U);text(600,150,'Input Image','Color','y',...
 'FontSize',14,'FontWeight','bold');
ɔplot (3,1,2);imshow (cropU);text(190,20,'Region of Interest','Color','b',...
 'FontSize',14,'FontWeight','bold');
 user_select==22
imread('22.jpg');
ɹth_start_row = 730;
```

```
uth_start_col = 750;
opV = V(mouth_start_row:1110, mouth_start_col:1260, :);
  trycrop=cropV;
  figure,subplot (3,1,1);imshow(V);text(600,150,'Input Image','Color','y',...
 'FontSize',14,'FontWeight','bold');
oplot (3,1,2);imshow (cropV);text(190,20,'Region of Interest','Color','b',...
 'FontSize',14,'FontWeight','bold');
 user_select==23
imread('23.jpg');
  mouth_start_row = 730;
uth_start_col = 750;
opW = W(mouth_start_row:1110, mouth_start_col:1260, :);
  trycrop=cropW;
  figure,subplot (3,1,1);imshow(W);text(600,150,'Input Image','Color','y',...
 'FontSize',14,'FontWeight','bold');
bplot (3,1,2);imshow (cropW);text(190,20,'Region of Interest','Color','b',...
 'FontSize',14,'FontWeight','bold');
 user_select==24
imread('24.jpg');
  mouth_start_row = 730;
uth_start_col = 750;
opX = X(mouth_start_row:1110, mouth_start_col:1260, :);
  trycrop=cropX;
  figure,subplot (3,1,1);imshow(X);text(600,150,'Input Image','Color','y',...
 'FontSize',14,'FontWeight','bold');
bplot (3,1,2);imshow (cropX);text(190,20,'Region of Interest','Color','b',...
 'FontSize',14,'FontWeight','bold');
 user_select==25
imread('25.jpg');
uth_start_row = 910;
uth_start_col = 770;
opY = Y(mouth_start_row:1190, mouth_start_col:1250, :);
  trycrop=cropY;
  figure,subplot (3,1,1);imshow(Y);text(600,150,'Input Image','Color','y',...
 'FontSize',14,'FontWeight','bold');
bplot (3,1,2);imshow (cropY);text(190,20,'Region of Interest','Color','b',...
 'FontSize',14,'FontWeight','bold');
 user_select==26
imread('26.jpg');
  mouth_start_row = 910;
uth_start_col = 770;
opZ = Z(mouth_start_row:1190, mouth_start_col:1250, :);
  trycrop=cropZ;
  figure,subplot (3,1,1);imshow(Z);text(600,150,'Input Image','Color','y',...
 'FontSize',14,'FontWeight','bold');
bplot (3,1,2);imshow (cropZ);text(190,20,'Region of Interest','Color','b',...
 'FontSize',14,'FontWeight','bold');
 user_select==27
=imread('27.jpg');
  mouth_start_row = 910;
uth_start_col = 770;
opAA = AA(mouth_start_row:1190, mouth_start_col:1250, :);
  trycrop=cropAA;
  figure,subplot (3,1,1);imshow(AA);text(600,150,'Input Image','Color','y',...
 'FontSize',14,'FontWeight','bold');
bplot (3,1,2);imshow (cropAA);text(190,20,'Region of Interest','Color','b',...
 'FontSize',14,'FontWeight','bold');
```

```
 user_select==28
=imread('28.jpg');
uth_start_row = 770;
uth_start_col = 750;
ɔpAB = AB(mouth_start_row:940, mouth_start_col:1250, :);
  trycrop=cropAB;
  figure,subplot (3,1,1);imshow(AB);text(600,150,'Input Image','Color','y',...
 'FontSize',14,'FontWeight','bold');
ɔplot (3,1,2);imshow (cropAB);text(190,20,'Region of Interest','Color','b',...
 'FontSize',14,'FontWeight','bold');
 user_select==29
=imread('29.jpg');
  mouth_start_row = 770;
uth_start_col = 750;
ɔpAC = AC(mouth_start_row:940, mouth_start_col:1250, :);
  trycrop=cropAC;
  figure,subplot (3,1,1);imshow(AC);text(600,150,'Input Image','Color','y',...
 'FontSize',14,'FontWeight','bold');
ɔplot (3,1,2);imshow (cropAC);text(190,20,'Region of Interest','Color','b',...
 'FontSize',14,'FontWeight','bold');
 user_select==30
=imread('30.jpg');
  mouth_start_row = 770;
uth_start_col = 750;
ɔpAD = AD(mouth_start_row:940, mouth_start_col:1250, :);
  trycrop=cropAD;
  figure,subplot (3,1,1);imshow(AD);text(600,150,'Input Image','Color','y',...
 'FontSize',14,'FontWeight','bold');
ɔplot (3,1,2);imshow (cropAD);text(190,20,'Region of Interest','Color','b',...
 'FontSize',14,'FontWeight','bold');
 user_select==31
=imread('31.jpg');
uth_start_row = 730;
uth_start_col = 780;
ɔpAE = AE(mouth_start_row:960, mouth_start_col:1275, :);
/crop=cropAE;
jure,subplot (3,1,1);imshow(AE);text(600,150,'Input Image','Color','y',...
 'FontSize',14,'FontWeight','bold');
ɔplot (3,1,2);imshow (cropAE);text(190,20,'Region of Interest','Color','b',...
 'FontSize',14,'FontWeight','bold');
```

```
く = mouth_start_col-1;
ʏ = mouth_start_row-1;

ɔ2gray(trycrop);
ɔld = graythresh(I);
ɔld = graythresh(I);

]=size(img);
color and display
l
lt=edge(img,'prewitt');
 z==3
g1=rgb2ycbcr(img);
l=edge(img1(:,:,1),'prewitt');
l=(dx1*511);
g2(:,:,1)=dx1;
g2(:,:,2)=img1(:,:,2);
g2(:,:,3)=img1(:,:,3);
lt=ycbcr2rgb(uint8(img2));


;
:(3,1,3);imshow (R);text(190,20,'Edge Color','Color','b',...
 'FontSize',14,'FontWeight','bold');


r
color(I,128,255);
cial('unsharp');
ilt2(h,I,BW);
n2bw(I,threshold);
BW;

size(BW);
le mouth
 4;
 min(find(BW(:,col1)));
ı angled (between upper lips and below lips)
 12;
 min(find(BW(row2,:)));
```

```
ry1 = bwtraceboundary(BW, [row1, col1], 'N', 8, 70);
the search direction to counterclockwise, in order to trace downward.
ry2 = bwtraceboundary(BW, [row2, col2], 'E', 8,90,'counter');

y offsets in order to draw in the original image
polyfit(boundary1(:,2), boundary1(:,1), 1);
polyfit(boundary2(:,2), boundary2(:,1), 1);
= [1 ab1(1)]; % create a vector based on the line equation
= [1 ab2(1)];
ot(vect1, vect2);

ute vector lengths
1 = sqrt(sum(vect1.^2));
2 = sqrt(sum(vect2.^2));

in the larger angle of intersection in degrees
= 180-acos(dp/(length1*length2))*180/pi;
in=angle;
le_in<=75
gle_in=angle;

gle_in>75 && angle_in<=90)
gle_in=angle_in - 45;
 (angle_in>90)
gle_in=angle_in-90




ection = [1 ,-ab1(1); 1, -ab2(1)] \ [ab1(2); ab2(2)];

y offsets in order to compute the location in the original,
 not cropped, image.

ection = intersection + [offsetY; offsetX];
x = intersection(2);
y = intersection(1);


ay result
gle_in<=17)
-300,300,'the input image is match with the segmented of happy lips after comparing
the database','Color','b',...
 'FontSize',10,'FontWeight','bold');
 (angle_in>17 && angle_in<=44)
-300,300,'the input image is match with the segmented of angry lips after comparing
the database','Color','b',...
 'FontSize',10,'FontWeight','bold');
 (angle_in>44)
ext (-300,300,'the input image is match with the segmented of sad lips after compar
th the database','Color','b',...
 'FontSize',10,'FontWeight','bold');
d
```

```
r_select == 1
= imread ('1.jpg');
er=A;
= rgb2gray(user);
eboth_start_row = 555;
eboth_start_col = 700;
opM = M(eyeboth_start_row:730, eyeboth_start_col:1000);
ycrop=cropM;
 user_select == 2
= imread ('2.jpg');
er=B;
= rgb2gray(user);
eboth_start_row = 555;
eboth_start_col = 700;
opM = M(eyeboth_start_row:730, eyeboth_start_col:1000);
ycrop=cropM;
 user_select == 3
= imread ('3.jpg');
er=C;
= rgb2gray(user);
eboth_start_row = 555;
eboth_start_col = 700;
opM = M(eyeboth_start_row:730, eyeboth_start_col:1000);
ycrop=cropM;
 user_select == 4
= imread ('4.jpg');
er=D;
= rgb2gray(user);
eboth_start_row = 455;
eboth_start_col = 700;
opM = M(eyeboth_start_row:630, eyeboth_start_col:1000);
ycrop=cropM;
 user_select ==5
= imread ('5.jpg');
er=E;
= rgb2gray(user);
eboth_start_row = 455;
eboth_start_col = 700;
opM = M(eyeboth_start_row:630, eyeboth_start_col:1000);
ycrop=cropM;
 user_select ==6
= imread ('6.jpg');
er=F;
= rgb2gray(user);
eboth_start_row = 455;
eboth_start_col = 700;
opM = M(eyeboth_start_row:630, eyeboth_start_col:1000);
ycrop=cropM;
 user_select == 7
= imread ('7.jpg');
er=G;
= rgb2gray(user);
eboth_start_row = 560;
eboth_start_col = 700;
opM = M(eyeboth_start_row:735, eyeboth_start_col:1000),
ycrop=cropM;
```

```
 user_select == 8
= imread ('8.jpg');
er=H;
= rgb2gray(user);
eboth_start_row = 560;
eboth_start_col = 700;
opM = M(eyeboth_start_row:735, eyeboth_start_col:1000),
ycrop=cropM;
 user_select == 9
= imread ('9.jpg');
er=I;
= rgb2gray(user);
eboth_start_row = 560;
eboth_start_col = 700;
opM = M(eyeboth_start_row:735, eyeboth_start_col:1000),
ycrop=cropM;
 user_select == 10
= imread ('10.jpg');
er=J;
= rgb2gray(user);
eboth_start_row = 560;
eboth_start_col = 700;
opM = M(eyeboth_start_row:735, eyeboth_start_col:1000),
ycrop=cropM;
 user_select == 11
= imread ('11.jpg');
er=K;
= rgb2gray(user);
eboth_start_row = 560;
eboth_start_col = 700;
opM = M(eyeboth_start_row:735, eyeboth_start_col:1000),
ycrop=cropM;
 user_select == 12
= imread ('12.jpg');
er=L;
= rgb2gray(user);
eboth_start_row = 560;
eboth_start_col = 700;
opM = M(eyeboth_start_row:735, eyeboth_start_col:1000),
ycrop=cropM;
 user_select == 13
= imread ('13.jpg');
er=M;
= rgb2gray(user);
eboth_start_row = 580;
eboth_start_col = 780;
opM = M(eyeboth_start_row:755, eyeboth_start_col:1080),
ycrop=cropM;
 user_select == 14
= imread ('14.jpg');
er=N;
= rgb2gray(user);
eboth_start_row = 580;
eboth_start_col = 780;
opM = M(eyeboth_start_row:755, eyeboth_start_col:1080),
ycrop=cropM;
 user_select == 15
```

```
= imread ('15.jpg');
er=O;
= rgb2gray(user);
eboth_start_row = 580;
eboth_start_col = 780;
opM = M(eyeboth_start_row:755, eyeboth_start_col:1080),
ycrop=cropM;
 user_select == 16
= imread ('16.jpg');
er=P;
= rgb2gray(user);
eboth_start_row = 460;
eboth_start_col = 750;
opM = M(eyeboth_start_row:635, eyeboth_start_col:1050),
ycrop=cropM;
 user_select == 17
= imread ('17.jpg');
er=Q;
= rgb2gray(user);
eboth_start_row = 460;
eboth_start_col = 750;
opM = M(eyeboth_start_row:635, eyeboth_start_col:1050),
ycrop=cropM;
 user_select == 18
= imread ('18.jpg');
er=R;
= rgb2gray(user);
eboth_start_row = 460;
eboth_start_col = 750;
opM = M(eyeboth_start_row:635, eyeboth_start_col:1050),
ycrop=cropM;
 user_select == 19
= imread ('19.jpg');
er=S;
= rgb2gray(user);
eboth_start_row = 460;
eboth_start_col = 730;
opM = M(eyeboth_start_row:635, eyeboth_start_col:1030),
ycrop=cropM;
 user_select == 20
= imread ('20.jpg');
er=T;
= rgb2gray(user);
eboth_start_row = 460;
eboth_start_col = 730;
opM = M(eyeboth_start_row:635, eyeboth_start_col:1030),
ycrop=cropM;
 user_select == 21
= imread ('21.jpg');
er=U;
= rgb2gray(user);
eboth_start_row = 460;
eboth_start_col = 730;
opM = M(eyeboth_start_row:635, eyeboth_start_col:1030),
ycrop=cropM;
 user_select == 22
= imread ('22.jpg');
```

```
er=V;
= rgb2gray(user);
eboth_start_row = 520;
eboth_start_col = 710;
opM = M(eyeboth_start_row:695, eyeboth_start_col:1010),
ycrop=cropM;
 user_select == 23
= imread ('23.jpg');
er=W;
= rgb2gray(user);
eboth_start_row = 520;
eboth_start_col = 710;
opM = M(eyeboth_start_row:695, eyeboth_start_col:1010),
ycrop=cropM;
 user_select == 24
= imread ('24.jpg');
er=X;
= rgb2gray(user);
eboth_start_row = 520;
eboth_start_col = 710;
opM = M(eyeboth_start_row:695, eyeboth_start_col:1010),
ycrop=cropM;
 user_select == 25
= imread ('25.jpg');
er=Y;
= rgb2gray(user);
eboth_start_row = 630;
eboth_start_col = 710;
opM = M(eyeboth_start_row:805, eyeboth_start_col:1010),
ycrop=cropM;
 user_select == 26
= imread ('26.jpg');
er=Z;
= rgb2gray(user);
eboth_start_row = 630;
eboth_start_col = 710;
opM = M(eyeboth_start_row:805, eyeboth_start_col:1010),
ycrop=cropM;
 user_select == 27
 = imread ('27.jpg');
er=AA;
= rgb2gray(user);
eboth_start_row = 630;
eboth_start_col = 710;
opM = M(eyeboth_start_row:805, eyeboth_start_col:1010),
ycrop=cropM;
 user_select == 28
 = imread ('28.jpg');
er=AB;
= rgb2gray(user);
eboth_start_row = 542;
eboth_start_col = 710;
opM = M(eyeboth_start_row:717, eyeboth_start_col:1010),
ycrop=cropM;
 user_select == 29
 = imread ('29.jpg');
er=AC;
```

```
= rgb2gray(user);
eboth_start_row = 542;
eboth_start_col = 710;
opM = M(eyeboth_start_row:717, eyeboth_start_col:1010),
ycrop=cropM;
 user_select == 30
 = imread ('30.jpg');
er=AD;
= rgb2gray(user);
eboth_start_row = 542;
eboth_start_col = 710;
opM = M(eyeboth_start_row:717, eyeboth_start_col:1010),
ycrop=cropM;
 user_select == 31
 = imread ('31.jpg');
er=AE;
= rgb2gray(user);
eboth_start_row = 475;
eboth_start_col = 700;
opM = M(eyeboth_start_row:650, eyeboth_start_col:1000),
ycrop=cropM;
```

is used to measure the roundness of eyes, by assuming that both eyes dentical, therefore reduce the cropped image earline into two sided, hoose only one side as shown below

```
,subplot (2,2,1);imshow(user);text(600,150,'Input Image','Color','y',...
 'FontSize',14,'FontWeight','bold');
c (2,2,2);imshow (cropM);text(30,30,'Region of Interest','Color','b',...
 'FontSize',14,'FontWeight','bold');

M;

r out the noise
old = graythresh(I);
color(I,128,255);
cial('unsharp');
ilt2(h,I,BW);

ve all object containing fewer than 40 pixels
areaopen(BW,180);

 a gap in the pen's cap
rel('disk',2);
nclose(BW,se);

 any holes, so that regionprops can be used to estimate
area enclosed by each of the boundaries
nfill(BW,'holes');

= bwboundaries(BW,'noholes');

lay the label matrix and draw each boundary
v(label2rgb(L, @jet, [.5 .5 .5]))
1
= 1:length(B)
dary = B{k};


= regionprops(L,'Area','Centroid');

old = 0.94;

 over the boundaries
= 1:length(B)

tain (X,Y) boundary coordinates corresponding to label 'k'
dary = B{k};

mpute a simple estimate of the object's perimeter
a_sq = diff(boundary).^2;
meter = sum(sqrt(sum(delta_sq,2)));

tain the area calculation corresponding to label 'k'
 = stats(k).Area;

mpute the roundness metric
ic = 4*pi*area/perimeter^2;

splay the results
ic_string = sprintf('%2.2f',metric);

rk objects above the threshold with a black circle
```

```
etric > threshold
ntroid = stats(k).Centroid;
ot(centroid(1),centroid(2),'ko');



tric>0.2500 && metric< 0.3220 || metric>=0.5201 && metric<0.6434)
-250,250,'the input image is match with the segmented of happy eyes after comparing
the database','Color','b',...
 'FontSize',10,'FontWeight','bold');
yes=metric;
 (metric>0.42 && metric<0.5200 || metric>0.7239 && metric < 0.7771)
-250,250,'the input image is match with the segmented of angry eyes after comparing
the database','Color','b',...
 'FontSize',10,'FontWeight','bold');
yes=metric;
 (metric>0.3221 && metric<0.41 || metric>0.6435 && metric < 0.7238)
ext (-250,250,'the input image is match with the segmented of sad eyes after compar
th the database','Color','b',...
 'FontSize',10,'FontWeight','bold');
s=metric;
d




r_select == 1
= imread ('1.jpg');
er=A;
art_row = 410;
art_col = 720;
opA = A(start_row:580, start_col:1235, :);
ycrop=cropA;
 user_select==2
imread('2.jpg');
er=B;
art_row = 410;
art_col = 720;
opB = B(start_row:580, start_col:1235, :);
ycrop=cropB;
 user_select==3
imread('3.jpg');
er=C;
art_row = 410;
art_col = 720;
opC = C(start_row:580, start_col:1235, :);
ycrop=cropC;
 user_select == 4
= imread ('4.jpg');
er=D;
art_row = 390;
art_col = 720;
opD = D(start_row:560, start_col:1235, :);
ycrop=cropD;
 user_select == 5
= imread ('5.jpg');
```

```
er=E;
art_row = 390;
art_col = 720;
opE = E(start_row:560, start_col:1235, :);
ycrop=cropE;
 user_select == 6
= imread ('6.jpg');
er=F;
art_row = 390;
art_col = 720;
opF = F(start_row:560, start_col:1235, :);
ycrop=cropF;
 user_select == 7
= imread ('7.jpg');
er=G;
art_row = 430;
art_col = 720;
opG = G(start_row:600, start_col:1235, :);
ycrop=cropG;
 user_select == 8
= imread ('8.jpg');
er=H;
tart_row = 430;
art_col = 720;
opH =H(start_row:600, start_col:1235, :);
ycrop=cropH;
 user_select == 9
= imread ('9.jpg');
er=I;
art_row = 430;
art_col = 720;
opI = I(start_row:600, start_col:1235, :);
ycrop=cropI;
 user_select == 10
= imread ('10.jpg');
er=J;
art_row = 420;
art_col = 720;
opJ = J(start_row:650, start_col:1235, :);
ycrop=cropJ;
 user_select == 11
= imread ('11.jpg');
er=K;
art_row = 420;
art_col = 720;
opK = K(start_row:650, start_col:1235, :);
ycrop=cropK;
 user_select == 12
= imread ('12.jpg');
er=L;
art_row = 420;
art_col = 720;
opL = L(start_row:650, start_col:1235, :);
ycrop=cropL;

 user_select == 13
= imread ('13.jpg');
```

```
er=M;
art_row = 480;
art_col = 770;
opM = M(start_row:650, start_col:1285, :);
ycrop=cropM;
 user_select == 14
= imread ('14.jpg');
er=N;
art_row = 480;
art_col = 770;
opN = N(start_row:650, start_col:1285, :);
ycrop=cropN;
 user_select == 15
= imread ('15.jpg');
er=O;
art_row = 480;
art_col = 770;
opO = O(start_row:650, start_col:1285, :);
ycrop=cropO;

 user_select == 16
= imread ('16.jpg');
er=P;
art_row = 390;
art_col = 720;
opP = P(start_row:560, start_col:1235, :);
ycrop=cropP;
 user_select == 17
= imread ('17.jpg');
er=Q;
art_row = 390;
art_col = 720;
opQ= Q(start_row:560, start_col:1235, :);
ycrop=cropQ;
 user_select == 18
= imread ('18.jpg');
er=R;
art_row = 390;
art_col = 720;
opR = R(start_row:560, start_col:1235, :);
ycrop=cropR;

 user_select == 19
= imread ('19.jpg');
er=S;
art_row = 370;
art_col = 720;
opS = S(start_row:540, start_col:1235, :);
ycrop=cropS;
 user_select == 20
= imread ('20.jpg');
er=T;
art_row = 370;
art_col = 720;
opT = T(start_row:540, start_col:1235, :);
ycrop=cropT;
 user_select == 21
```

```
= imread ('21.jpg');
ɘr=U;
ɪrt_row = 370;
ɪrt_col = 720;
ɔpU = U(start_row:540, start_col:1235, :);
ʏcrop=cropU;

 user_select == 22
= imread ('22.jpg');
ɘr=V;
ɪrt_row = 420;
ɪrt_col = 720;
ɔpV = V(start_row:590, start_col:1235, :);
ʏcrop=cropV;
 user_select == 23
= imread ('23.jpg');
ɘr=W;
ɪrt_row = 420;
ɪrt_col = 720;
ɔpW = W(start_row:590, start_col:1235, :);
ʏcrop=cropW;
 user_select == 24
= imread ('24.jpg');
ɘr=X;
ɪrt_row = 420;
ɪrt_col = 720;
ɔpX = X(start_row:590, start_col:1235, :);
ʏcrop=cropX;

 user_select == 25
= imread ('25.jpg');
ɘr=Y;
ɪrt_row = 550;
ɪrt_col = 720;
ɔpY = Y(start_row:720, start_col:1235, :);
ʏcrop=cropY;
 user_select == 26
= imread ('26.jpg');
ɘr=Z;
ɪrt_row = 550;
ɪrt_col = 720;
ɔpZ = Z(start_row:720, start_col:1235, :);
ʏcrop=cropZ;
 user_select == 27
 = imread ('27.jpg');
ɘr=AA;
ɪrt_row = 550;
ɪrt_col = 720;
ɔpAA = AA(start_row:720, start_col:1235, :);
ʏcrop=cropAA;

 user_select == 28
 = imread ('28.jpg');
ɘr=AB;
ɪrt_row = 410;
ɪrt_col = 720;
ɔpAB=AB(start_row:580, start_col:1235, :);
```

```
/crop=cropAB;
 user_select == 29
 = imread ('29.jpg');
er=AC;
art_row = 410;
art_col = 720;
opAC = AC(start_row:580, start_col:1235, :);
/crop=cropAC;
 user_select == 30
 = imread ('30.jpg');
er=AD;
art_row = 410;
art_col = 720;
opAD = AD(start_row:580, start_col:1235, :);
/crop=cropAD;
 user_select == 31
 = imread ('31.jpg');
er=AE;
art_row = 390;
art_col = 720;
opAE = AE(start_row:620, start_col:1235, :);
/crop=cropAE;
```

```
,subplot (3,1,1);imshow(user);text(600,150,'Input Image','Color','y',...
```

```
 'FontSize',14,'FontWeight','bold');
: (3,1,2);imshow (trycrop);text(190,20,'Region of Interest','Color','b',...
 'FontSize',14,'FontWeight','bold');
 1];
 1];
=impixel(user,c,r);
l; 1];
av(pixels);
ranspose(pixels);
ls*B;


o2gray(trycrop);
old = graythresh(I);

]=size(img);

ring out the noise
color(I,128,255);
cial('unsharp');
ilt2(h,I,BW);
m2bw(I,threshold);

color and display
l
lt=edge(img,'prewitt');
 z==3
g1=rgb2ycbcr(img);
l=edge(img1(:,:,1),'prewitt');
l=(dx1*511);
g2(:,:,1)=dx1;
g2(:,:,2)=img1(:,:,2);
g2(:,:,3)=img1(:,:,3);
lt=ycbcr2rgb(uint8(img2));


;
:(3,1,3);imshow (R);text(190,20,'Edge Color','Color','b',...
 'FontSize',14,'FontWeight','bold');

calculation
(K);
ard deviation calculation
rd=std2(trycrop);
nce calculation
tandard*standard);
r;
eness calculation

-b;

ay result
=108)
-180,180,'the input image is match with the segmented of happy forehead after compa
ith the database','Color','b',...
 'FontSize',10,'FontWeight','bold');
orehead=T;
```

```
 (T>=109 && T<=117)
-180,180,'the input image is match with the segmented of sad forehead after compari
ı the database','Color','b',...
 'FontSize',10,'FontWeight','bold');
ehead=T;
 (T>=118 && T<=187)
ext (-180,200,'the input image is match with the segmented of angry forehead after
ing with the database','Color','b',...
 'FontSize',10,'FontWeight','bold');
orehead=T;
d




r_select == 1
= imread ('1.jpg');
er=A;
art_row = 480;
art_col = 760;
opA = A(start_row:820, start_col:1240, :);
ycrop=cropA;
 user_select==2
= imread ('2.jpg');
  user=B;
art_row = 480;
art_col = 760;
opB = B(start_row:820, start_col:1240, :);
ycrop=cropB;
 user_select==3
= imread ('3.jpg');
  user=C;
art_row = 480;
art_col = 760;
opC = C(start_row:820, start_col:1240, :);
ycrop=cropC;
 user_select==4
= imread ('4.jpg');
  user=D;
art_row = 410;
art_col = 750;
opD = D(start_row:750, start_col:1230, :);
ycrop=cropD;
 user_select==5
= imread ('5.jpg');
  user=E;
art_row = 410;
art_col = 750;
opE = E(start_row:750, start_col:1230, :);
ycrop=cropE;
 user_select==6
  user=F;
= imread ('6.jpg');
art_row = 410;
art_col = 750;
opF = F(start_row:750, start_col:1230, :);
```

```
/crop=cropF;
 user_select==7
= imread ('7.jpg');
  user=G;
art_row = 500;
art_col = 730;
>pG = G(start_row:840, start_col:1210, :);
/crop=cropG;
 user_select==8
= imread ('8.jpg');
  user=H;
art_row = 500;
art_col = 730;
>pH = H(start_row:840, start_col:1210, :);
/crop=cropH;
 user_select==9
= imread ('9.jpg');
  user=I;
art_row = 500;
art_col = 730;
>pI = I(start_row:840, start_col:1210, :);
/crop=cropI;
 user_select==10
= imread ('10.jpg');
  user=J;
art_row = 520;
art_col = 780;
>pJ = J(start_row:860, start_col:1235, :);
/crop=cropJ;
 user_select==11
= imread ('11.jpg');
  user=K;
art_row = 520;
art_col = 780;
>pK = K(start_row:860, start_col:1235, :);
/crop=cropK;
 user_select==12
= imread ('12.jpg');
  user=L;
art_row = 520;
art_col = 780;
>pL = L(start_row:860, start_col:1235, :);
/crop=cropL;
 user_select==13
= imread ('13.jpg');
  user=M;
art_row = 550;
art_col = 775;
>pM = M(start_row:890, start_col:1255, :);
/crop=cropM;
 user_select==14
= imread ('14.jpg');
  user=N;
art_row = 550;
art_col = 775;
>pN = N(start_row:890, start_col:1255, :);
/crop=cropN;
```

```
 user_select==15
= imread ('15.jpg');
  user=O;
art_row = 550;
art_col = 775;
pO = O(start_row:890, start_col:1255, :);
ycrop=cropO;
 user_select==16
= imread ('16.jpg');
  user=P;
art_row = 430;
art_col = 770;
pP = P(start_row:770, start_col:1250, :);
ycrop=cropP;
 user_select==17
= imread ('17.jpg');
  user=Q;
art_row = 430;
art_col = 770;
pQ = Q(start_row:770, start_col:1250, :);
ycrop=cropQ;
 user_select==18
= imread ('18.jpg');
  user=R;
art_row = 430;
art_col = 770;
pR= R(start_row:770, start_col:1250, :);
ycrop=cropR;
 user_select==19
= imread ('19.jpg');
  user=S;
art_row = 390;
art_col = 760;
pS = S(start_row:730, start_col:1240, :);
ycrop=cropS;
 user_select==20
= imread ('20.jpg');
  user=T;
art_row = 390;
art_col = 760;
pT = T(start_row:730, start_col:1240, :);
ycrop=cropT;
 user_select==21
= imread ('21.jpg');
  user=U;
art_row = 390;
art_col = 760;
pU = U(start_row:730, start_col:1240, :);
ycrop=cropU;
 user_select==22
= imread ('22.jpg');
  user=V;
art_row = 440;
art_col = 760;
pV = V(start_row:780, start_col:1240, :);
ycrop=cropV;
 user_select==23
```

```
= imread ('23.jpg');
  user=W;
art_row = 440;
art_col = 760;
opW = W(start_row:780, start_col:1240, :);
ycrop=cropW;
 user_select==24
= imread ('24.jpg');
  user=X;
art_row = 440;
art_col = 760;
opX = X(start_row:780, start_col:1240, :);
ycrop=cropX;
 user_select==25
= imread ('25.jpg');
  user=Y;
art_row = 610;
art_col = 750;
opY = Y(start_row:950, start_col:1230, :);
ycrop=cropY;
 user_select==26
= imread ('26.jpg');
  user=Z;
art_row = 610;
art_col = 750;
opZ = Z(start_row:950, start_col:1230, :);
crop=cropZ;
 user_select==27
 = imread ('27.jpg');
  user=AA;
rt_row = 610;
rt_col = 750;
pAA = AA(start_row:950, start_col:1230, :);
crop=cropAA;
 user_select==28
 = imread ('28.jpg');
  user=AB;
rt_row = 460;
rt_col = 750;
pAB = AB(start_row:800, start_col:1230, :);
crop=cropAB;
user_select==29
= imread ('29.jpg');
 user=AC;
rt_row = 460;
rt_col = 750;
pAC = AC(start_row:800, start_col:1230, :);
crop=cropAC;
user_select==30
= imread ('30.jpg');
 user=AD;
rt_row = 460;
rt_col = 750;
pAD = AD(start_row:800, start_col:1230, :);
crop=cropAD;

user_select==31
```

```
; = imread ('31.jpg');
er=AE;
art_row = 460;
art_col = 750;
opAE = AE(start_row:800, start_col:1230, :);
ycrop=cropAE;
```

```
,subplot (3,1,1);imshow(user);text(600,150,'Input Image','Color','y',...
 'FontSize',14,'FontWeight','bold');
t (3,1,2);imshow (trycrop);text(150,20,'Region of Interest','Color','b',...
 'FontSize',14,'FontWeight','bold');


b2gray(trycrop);
old = graythresh(I);

]=size(img);


color and display
1
lt=edge(img,'prewitt');
 z==3
g1=rgb2ycbcr(img);
l=edge(img1(:,:,1),'prewitt');
```

```
1=(dx1*511);
g2(:,:,1)=dx1;
g2(:,:,2)=img1(:,:,2);
g2(:,:,3)=img1(:,:,3);
lt=ycbcr2rgb(uint8(img2));



;
t(3,1,3);imshow (R);text(190,20,'Edge Color','Color','b',...
 'FontSize',14,'FontWeight','bold');

ring out the noise
color(I,128,255);
cial('unsharp');
ilt2(h,I,BW);
m2bw(I,threshold);
] = size(trycrop);

late mean
2(trycrop);
late standard deviation
rd=std2(trycrop);
late Variance
tandard*standard);
r;
lculate coarseness of the picture

-b;

ar<=0.99962843)
-400,400,'the input image is match with the segmented of happy face after comparing
the database','Color','b',...
 'FontSize',10,'FontWeight','bold');

 (coar>0.99971)
-400,400,'the input image is match with the segmented of angry face after comparing
the database','Color','b',...
 'FontSize',10,'FontWeight','bold');
ace=1;
 (coar>=0.99963 && coar<0.99971)
ext (-400,400,'the input image is match with the segmented of sad face after compar
th the database','Color','b',...
 'FontSize',10,'FontWeight','bold');
e=1;
d




r_select==20

warn_fig=msgbox('happy face=60%; happy forehead=60%; sad eyes=60%; angry lips=60%,
ore it is HAPPY');
push=findobj(h_warn_fig,'style','pushbutton');
t(h_push,'units','pixels');
sh_pos=get(h_push,'position')
```

```
g_pos=get(h_warn_fig,'position');
int_loc=fig_pos(1:2)+push_pos(1:2)+push_pos(3:4)/2;
t(h_warn_fig,'userdata',point_loc)
t(0,'pointerlocation',point_loc,'userdata',h_warn_fig);

ile any(h_warn_fig==get(0,'children'))
  set (0,'pointerlocation',get(h_warn_fig,'userdata'));
  drawnow
d


ar<=0.99962843 && T<=108 && (metric>0.2500 && metric< 0.3220 || metric>=0.5201 && n
0.6434) && angle_in<=17)

warn_fig=msgbox('happy face=60%; happy forehead=60%; happy eyes=60%; happy lips=90
efore it is HAPPY');
push=findobj(h_warn_fig,'style','pushbutton');
t(h_push,'units','pixels');
sh_pos=get(h_push,'position')
g_pos=get(h_warn_fig,'position');
int_loc=fig_pos(1:2)+push_pos(1:2)+push_pos(3:4)/2;
t(h_warn_fig,'userdata',point_loc)
t(0,'pointerlocation',point_loc,'userdata',h_warn_fig);

ile any(h_warn_fig==get(0,'children'))
  set (0,'pointerlocation',get(h_warn_fig,'userdata'));
  drawnow
d

 (coar<=0.99962843 && T<=108 && (metric>0.2500 && metric< 0.3220 || metric>=0.5201
ric<0.6434) && angle_in>17 && angle_in<=44)

warn_fig=warndlg('happy face=60%; happy forehead=60%; happy eyes=60%; angry lips=60
refore it is HAPPY');
push=findobj(h_warn_fig,'style','pushbutton');
t(h_push,'units','pixels');
sh_pos=get(h_push,'position')
g_pos=get(h_warn_fig,'position');
int_loc=fig_pos(1:2)+push_pos(1:2)+push_pos(3:4)/2;
t(h_warn_fig,'userdata',point_loc)
t(0,'pointerlocation',point_loc,'userdata',h_warn_fig);

ile any(h_warn_fig==get(0,'children'))
  set (0,'pointerlocation',get(h_warn_fig,'userdata'));
  drawnow
d

 (coar<=0.99962843 && T<=108 && (metric>0.2500 && metric< 0.3220 || metric>=0.5201
ric<0.6434) && angle_in>44)

warn_fig=warndlg('happy face=60%; happy forehead=60%; happy eyes=60%; sad lips=60%,
fore it is HAPPY');
push=findobj(h_warn_fig,'style','pushbutton');
t(h_push,'units','pixels');
sh_pos=get(h_push,'position')
g_pos=get(h_warn_fig,'position');
int_loc=fig_pos(1:2)+push_pos(1:2)+push_pos(3:4)/2;
```

```
t(h_warn_fig,'userdata',point_loc)
t(0,'pointerlocation',point_loc,'userdata',h_warn_fig);

ile any(h_warn_fig==get(0,'children'))
  set (0,'pointerlocation',get(h_warn_fig,'userdata'));
  drawnow
d

 (coar<=0.99962843 && T<=108 && (metric>0.42 && metric<0.5200 || metric>0.7239 && n
< 0.7771) && angle_in<=17)
warn_fig=warndlg('happy face=60%; happy forehead=60%; angry eyes=70%; happy lips=9(
refore it is HAPPY');
push=findobj(h_warn_fig,'style','pushbutton');
t(h_push,'units','pixels');
sh_pos=get(h_push,'position')
g_pos=get(h_warn_fig,'position');
int_loc=fig_pos(1:2)+push_pos(1:2)+push_pos(3:4)/2;
t(h_warn_fig,'userdata',point_loc)
t(0,'pointerlocation',point_loc,'userdata',h_warn_fig);

ile any(h_warn_fig==get(0,'children'))
  set (0,'pointerlocation',get(h_warn_fig,'userdata'));
  drawnow
d


 (coar<=0.99962843 && T<=108 && (metric>0.42 && metric<0.5200 || metric>0.7239 && n
< 0.7771) && angle_in<=17)

warn_fig=warndlg('happy face=60%; happy forehead=60%; angry eyes=70%; happy lips=9(
refore it is HAPPY');
push=findobj(h_warn_fig,'style','pushbutton');
t(h_push,'units','pixels');
sh_pos=get(h_push,'position')
g_pos=get(h_warn_fig,'position');
int_loc=fig_pos(1:2)+push_pos(1:2)+push_pos(3:4)/2;
t(h_warn_fig,'userdata',point_loc)
t(0,'pointerlocation',point_loc,'userdata',h_warn_fig);

ile any(h_warn_fig==get(0,'children'))
  set (0,'pointerlocation',get(h_warn_fig,'userdata'));
  drawnow
d

 (coar<=0.99962843 && T<=108 && (metric>0.3221 && metric<0.41 || metric>0.6435 && n
< 0.7238) && angle_in<=17)

warn_fig=warndlg('happy face=60%; happy forehead=60%; sad eyes=60%; happy lips=90%,
fore it is HAPPY');
push=findobj(h_warn_fig,'style','pushbutton');
t(h_push,'units','pixels');
sh_pos=get(h_push,'position')
g_pos=get(h_warn_fig,'position');
int_loc=fig_pos(1:2)+push_pos(1:2)+push_pos(3:4)/2;
t(h_warn_fig,'userdata',point_loc)
t(0,'pointerlocation',point_loc,'userdata',h_warn_fig);
```

```
ile any(h_warn_fig==get(0,'children'))
  set (0,'pointerlocation',get(h_warn_fig,'userdata'));
  drawnow
d

 (coar<=0.99962843 && T>=118 && T<=187 && (metric>0.2500 && metric< 0.3220 || metri
201 && metric<0.6434) && angle_in<=17)

warn_fig=warndlg('happy face=60%; angry forehead=60%; happy eyes=60%; happy lips=9(
refore it is HAPPY');
push=findobj(h_warn_fig,'style','pushbutton');
t(h_push,'units','pixels');
sh_pos=get(h_push,'position')
g_pos=get(h_warn_fig,'position');
int_loc=fig_pos(1:2)+push_pos(1:2)+push_pos(3:4)/2;
t(h_warn_fig,'userdata',point_loc)
t(0,'pointerlocation',point_loc,'userdata',h_warn_fig);

ile any(h_warn_fig==get(0,'children'))
  set (0,'pointerlocation',get(h_warn_fig,'userdata'));
  drawnow
d

 (coar<=0.99962843 && T>=109 && T<=117 && (metric>0.2500 && metric< 0.3220 || metri
201 && metric<0.6434) && angle_in<=17)

warn_fig=warndlg('happy face=60%; sad forehead=40%; happy eyes=60%; happy lips=90%,
fore it is HAPPY');
push=findobj(h_warn_fig,'style','pushbutton');
t(h_push,'units','pixels');
sh_pos=get(h_push,'position')
g_pos=get(h_warn_fig,'position');
int_loc=fig_pos(1:2)+push_pos(1:2)+push_pos(3:4)/2;
t(h_warn_fig,'userdata',point_loc)
t(0,'pointerlocation',point_loc,'userdata',h_warn_fig);

ile any(h_warn_fig==get(0,'children'))
  set (0,'pointerlocation',get(h_warn_fig,'userdata'));
  drawnow
d

 (coar>0.99971 && T<=108 && metric>0.2500 && (metric< 0.3220 || metric>=0.5201 && m
0.6434) && angle_in<=17)
sp ('happy9')
 (coar>=0.99963 && coar<0.99971 && T<=108 && (metric>0.2500 && metric< 0.3220 || me
0.5201 && metric<0.6434) && angle_in<=17)
sp ('happy10')



 (coar>0.99971 && T>=118 && T<=187 && (metric>0.42 && metric<0.5200 || metric>0.723
etric < 0.7771) && angle_in>17 && angle_in<=44)

warn_fig=warndlg('angry face=50%; angry forehead=60%; angry eyes=70%; angry lips=6C
refore it is ANGRY');
push=findobj(h_warn_fig,'style','pushbutton');
t(h_push,'units','pixels');
```

```
.sh_pos=get(h_push,'position')
g_pos=get(h_warn_fig,'position');
int_loc=fig_pos(1:2)+push_pos(1:2)+push_pos(3:4)/2;
t(h_warn_fig,'userdata',point_loc)
t(0,'pointerlocation',point_loc,'userdata',h_warn_fig);

ile any(h_warn_fig==get(0,'children'))
   set (0,'pointerlocation',get(h_warn_fig,'userdata'));
   drawnow
d

 (coar>0.99971 && T>=118 && T<=187 && (metric>0.42 && metric<0.5200 || metric>0.72
etric < 0.7771) && angle_in>44)

warn_fig=warndlg('angry face=50%; angry forehead=60%; angry eyes=70%; sad lips=60%
fore it is ANGRY');
push=findobj(h_warn_fig,'style','pushbutton');
t(h_push,'units','pixels');
sh_pos=get(h_push,'position')
g_pos=get(h_warn_fig,'position');
int_loc=fig_pos(1:2)+push_pos(1:2)+push_pos(3:4)/2;
t(h_warn_fig,'userdata',point_loc)
t(0,'pointerlocation',point_loc,'userdata',h_warn_fig);

ile any(h_warn_fig==get(0,'children'))
   set (0,'pointerlocation',get(h_warn_fig,'userdata'));
   drawnow
d

 (coar>0.99971 && T>=118 && T<=187 && (metric>0.42 && metric<0.5200 || metric>0.72
etric < 0.7771) && angle_in<=17)

warn_fig=warndlg('angry face=50%; angry forehead=60%; angry eyes=70%; happy lips=9
erefore it is ANGRY');
push=findobj(h_warn_fig,'style','pushbutton');
et(h_push,'units','pixels');
sh_pos=get(h_push,'position')
g_pos=get(h_warn_fig,'position');
int_loc=fig_pos(1:2)+push_pos(1:2)+push_pos(3:4)/2;
et(h_warn_fig,'userdata',point_loc)
et(0,'pointerlocation',point_loc,'userdata',h_warn_fig);

ile any(h_warn_fig==get(0,'children'))
   set (0,'pointerlocation',get(h_warn_fig,'userdata'));
   drawnow
nd

f (coar>0.99971 && T>=118 && T<=187 && (metric>0.3221 && metric<0.41 || metric>0.64
metric < 0.7238) && angle_in>17 && angle_in<=44 )

_warn_fig=warndlg('angry face=50%, angry forehead=60%, sad eyes=60%, angry lips=60%
efore it is ANGRY');
_push=findobj(h_warn_fig,'style','pushbutton');
et(h_push,'units','pixels');
ush_pos=get(h_push,'position')
ig_pos=get(h_warn_fig,'position');
oint_loc=fig_pos(1:2)+push_pos(1:2)+push_pos(3:4)/2;
```

```
:t(h_warn_fig,'userdata',point_loc)
:t(0,'pointerlocation',point_loc,'userdata',h_warn_fig);

ıile any(h_warn_fig==get(0,'children'))
   set (0,'pointerlocation',get(h_warn_fig,'userdata'));
   drawnow
ıd

: (coar>0.99971 && T>=118 && T<=187 && (metric>0.2500 && metric< 0.3220 || metric>=(
 && metric<0.6434) && angle_in>17 && angle_in<=44)
.sp ('angry15')
: (coar>0.99971 && T>=109 && T<=117 && (metric>0.42 && metric<0.5200 || metric>0.72:
ıetric < 0.7771) && angle_in>17 && angle_in<=44)
.sp ('angry16')
: (coar>0.99971 && T<=108 && (metric>0.42 && metric<0.5200 || metric>0.7239 && metr:
7771) && angle_in>17 && angle_in<=44)
.sp ('happy17')
 (coar>=0.99963 && coar<0.99971 && T>=118 && T<=187 && (metric>0.42 && metric<0.52(
ıetric>0.7239 && metric < 0.7771) && angle_in>17 && angle_in<=44)

 warn_fig=warndlg('sad face=50%; angry forehead=60%; angry eyes=70%; angry lips=60%,
:fore it is ANGRY');
 push=findobj(h_warn_fig,'style','pushbutton');
:t(h_push,'units','pixels');
ısh_pos=get(h_push,'position')
 g_pos=get(h_warn_fig,'position');
ıint_loc=fig_pos(1:2)+push_pos(1:2)+push_pos(3:4)/2;
:t(h_warn_fig,'userdata',point_loc)
:t(0,'pointerlocation',point_loc,'userdata',h_warn_fig);

.ile any(h_warn_fig==get(0,'children'))
   set (0,'pointerlocation',get(h_warn_fig,'userdata'));
   drawnow
.d

 (coar<=0.99962843 && T>=118 && T<=187 && (metric>0.42 && metric<0.5200 || metric>(
 && metric < 0.7771) && angle_in>17 && angle_in<=44)

 warn_fig=warndlg('happy face=60%; angry forehead=60%; angry eyes=70%; angry lips=6(
:refore it is ANGRY');
 push=findobj(h_warn_fig,'style','pushbutton');
 t(h_push,'units','pixels');
 sh_pos=get(h_push,'position')
 g_pos=get(h_warn_fig,'position');
 int_loc=fig_pos(1:2)+push_pos(1:2)+push_pos(3:4)/2;
 t(h_warn_fig,'userdata',point_loc)
 t(0,'pointerlocation',point_loc,'userdata',h_warn_fig);

 ile any(h_warn_fig==get(0,'children'))
   set (0,'pointerlocation',get(h_warn_fig,'userdata'));
   drawnow
 d

  (coar>=0.99963 && coar<0.99971 && T>=109 && T<=117 && (metric>0.3221 && metric<0.4
.etric>0.6435 && metric < 0.7238) && angle_in>44)
 sp ('sad20')
  (coar>=0.99963 && coar<0.99971 && T>=109 && T<=117 && (metric>0.3221 && metric<0.4
```

```
ietric>0.6435 && metric < 0.7238) && angle_in>17 && angle_in<=44)
 sp ('sad21')
  (coar>=0.99963 && coar<0.99971 && T>=109 && T<=117 && (metric>0.3221 && metric<0.4
ietric>0.6435 && metric < 0.7238) && angle_in<=17)
 sp ('sad22')
  (coar>=0.99963 && coar<0.99971 && T>=109 && T<=117 && (metric>0.42 && metric<0.520
ietric>0.7239 && metric < 0.7771) && angle_in>44)
 sp ('sad23')
  (coar>=0.99963 && coar<0.99971 && T>=109 && T<=117 && (metric>0.2500 && metric< 0.
 | metric>=0.5201 && metric<0.6434) && angle_in>44)
 sp ('sad24')
  (coar>=0.99963 && coar<0.99971 && T>=118 && T<=187 && (metric>0.3221 && metric<0.4
ietric>0.6435 && metric < 0.7238) && angle_in>44)

   h_warn_fig=warndlg('sad face=50%; angry forehead=60%; sad eyes=60%; sad lips=60%,
:fore it is SAD');
 push=findobj(h_warn_fig,'style','pushbutton');
:t(h_push,'units','pixels');
.sh_pos=get(h_push,'position')
.g_pos=get(h_warn_fig,'position');
int_loc=fig_pos(1:2)+push_pos(1:2)+push_pos(3:4)/2;
:t(h_warn_fig,'userdata',point_loc)
:t(0,'pointerlocation',point_loc,'userdata',h_warn_fig);

iile any(h_warn_fig==get(0,'children'))
   set (0,'pointerlocation',get(h_warn_fig,'userdata'));
   drawnow
id

  (coar>=0.99963 && coar<0.99971 && T<=108 && (metric>0.3221 && metric<0.41 || metr
35 && metric < 0.7238) && angle_in>44)
 sp ('sad26')

  (coar>0.99971 && T>=109 && T<=117 && (metric>0.3221 && metric<0.41 || metric>0.64
ietric < 0.7238) && angle_in>44)

 warn_fig=warndlg('angry face=50%; sad forehead=40%; sad eyes=60%; sad lips=60%; the
: it is SAD');
 push=findobj(h_warn_fig,'style','pushbutton');
:t(h_push,'units','pixels');
ish_pos=get(h_push,'position')
.g_pos=get(h_warn_fig,'position');
int_loc=fig_pos(1:2)+push_pos(1:2)+push_pos(3:4)/2;
:t(h_warn_fig,'userdata',point_loc)
:t(0,'pointerlocation',point_loc,'userdata',h_warn_fig);

iile any(h_warn_fig==get(0,'children'))
   set (0,'pointerlocation',get(h_warn_fig,'userdata'));
   drawnow
id

  (coar<=0.99962843 && T>=109 && T<=117 && (metric>0.3221 && metric<0.41 || metric>
 && metric < 0.7238) && angle_in>44)
 warn_fig=warndlg('happy face=60%; sad forehead=40%; sad eyes=60%; sad lips=60%, the
: it is SAD');
 push=findobj(h_warn_fig,'style','pushbutton');
:t(h_push,'units','pixels');
```

```
sh_pos=get(h_push,'position')
g_pos=get(h_warn_fig,'position');
int_loc=fig_pos(1:2)+push_pos(1:2)+push_pos(3:4)/2;
t(h_warn_fig,'userdata',point_loc)
t(0,'pointerlocation',point_loc,'userdata',h_warn_fig);

ile any(h_warn_fig==get(0,'children'))
  set (0,'pointerlocation',get(h_warn_fig,'userdata'));
  drawnow
d


 (coar>0.99971 && T>=118 && T<=187 && (metric>0.3221 && metric<0.41 || metric>0.643
etric < 0.7238) && angle_in>44)

warn_fig=warndlg('angry face=60%; angry forehead=60%; sad eyes=60%; sad lips=60%, t
re it is ANGRY');
push=findobj(h_warn_fig,'style','pushbutton');
t(h_push,'units','pixels');
sh_pos=get(h_push,'position')
g_pos=get(h_warn_fig,'position');
int_loc=fig_pos(1:2)+push_pos(1:2)+push_pos(3:4)/2;
t(h_warn_fig,'userdata',point_loc)
t(0,'pointerlocation',point_loc,'userdata',h_warn_fig);

ile any(h_warn_fig==get(0,'children'))
  set (0,'pointerlocation',get(h_warn_fig,'userdata'));
  drawnow
d

 (coar>0.99971 && T>=109 && T<=117 && (metric>0.3221 && metric<0.41 || metric>0.643
etric < 0.7238) && angle_in>17 && angle_in<=44)

warn_fig=warndlg('angry face=50%; sad forehead=40%; sad eyes=60%; angry lips=60%, t
re it is ANGRY');
push=findobj(h_warn_fig,'style','pushbutton');
t(h_push,'units','pixels');
sh_pos=get(h_push,'position')
g_pos=get(h_warn_fig,'position');
int_loc=fig_pos(1:2)+push_pos(1:2)+push_pos(3:4)/2;
t(h_warn_fig,'userdata',point_loc)
t(0,'pointerlocation',point_loc,'userdata',h_warn_fig);

ile any(h_warn_fig==get(0,'children'))
  set (0,'pointerlocation',get(h_warn_fig,'userdata'));
  drawnow
d

 (coar>0.99971 && T<=108 && (metric>0.42 && metric<0.5200 || metric>0.7239 && metri
7771) && angle_in<=17)

warn_fig=warndlg('angry face=50%; happy forehead=60%; angry eyes=70%; happy lips=9(
refore it is HAPPY');
push=findobj(h_warn_fig,'style','pushbutton');
t(h_push,'units','pixels');
sh_pos=get(h_push,'position')
g_pos=get(h_warn_fig,'position');
```

```
›int_loc=fig_pos(1:2)+push_pos(1:2)+push_pos(3:4)/2;
:t(h_warn_fig,'userdata',point_loc)
:t(0,'pointerlocation',point_loc,'userdata',h_warn_fig);

ıile any(h_warn_fig==get(0,'children'))
   set (0,'pointerlocation',get(h_warn_fig,'userdata'));
   drawnow
ıd


 (coar>0.99971 && T<=108 && (metric>0.42 && metric<0.5200 || metric>0.7239 && metr:
7771) && angle_in>44)

warn_fig=warndlg('angry face=50%; happy forehead=60%; angry eyes=70%; sad lips=60%,
:fore it is ANGRY');
push=findobj(h_warn_fig,'style','pushbutton');
:t(h_push,'units','pixels');
ısh_pos=get(h_push,'position')
.g_pos=get(h_warn_fig,'position');
›int_loc=fig_pos(1:2)+push_pos(1:2)+push_pos(3:4)/2;
:t(h_warn_fig,'userdata',point_loc)
:t(0,'pointerlocation',point_loc,'userdata',h_warn_fig);

ıile any(h_warn_fig==get(0,'children'))
   set (0,'pointerlocation',get(h_warn_fig,'userdata'));
   drawnow
ıd



 (coar<=0.99962843 && T<=108 && (metric>0.3221 && metric<0.41 || metric>0.6435 && ı
< 0.7238) && angle_in>17 && angle_in<=44)
sp ('happy33')
 (coar<=0.99962843 && T>=118 && T<=187 && (metric>0.2500 && metric< 0.3220 || metr:
·201 && metric<0.6434) && angle_in>44)

warn_fig=warndlg('happy face=60%; angry forehead=60%; happy eyes=60%; sad lips=60%,
:fore it is HAPPY');
push=findobj(h_warn_fig,'style','pushbutton');
:t(h_push,'units','pixels');
sh_pos=get(h_push,'position')
g_pos=get(h_warn_fig,'position');
›int_loc=fig_pos(1:2)+push_pos(1:2)+push_pos(3:4)/2;
:t(h_warn_fig,'userdata',point_loc)
:t(0,'pointerlocation',point_loc,'userdata',h_warn_fig);

.ile any(h_warn_fig==get(0,'children'))
   set (0,'pointerlocation',get(h_warn_fig,'userdata'));
   drawnow
.d

 (coar<=0.99962843 && T>=109 && T<=117 && (metric>0.42 && metric<0.5200 || metric>(
&& metric < 0.7771) && angle_in>17 && angle_in<=44)

warn_fig=warndlg('happy face=60%; sad forehead=40%; angry eyes=70%; angry lips=60%,
:fore it is ANGRY');
push=findobj(h_warn_fig,'style','pushbutton');
```

```
t(h_push,'units','pixels');
sh_pos=get(h_push,'position')
g_pos=get(h_warn_fig,'position');
int_loc=fig_pos(1:2)+push_pos(1:2)+push_pos(3:4)/2;
t(h_warn_fig,'userdata',point_loc)
t(0,'pointerlocation',point_loc,'userdata',h_warn_fig);

ile any(h_warn_fig==get(0,'children'))
  set (0,'pointerlocation',get(h_warn_fig,'userdata'));
  drawnow
d



 (coar>=0.99963 && coar<0.99971 && T<=108 && (metric>0.3221 && metric<0.41 || metri
35 && metric < 0.7238) && angle_in>17 && angle_in<=44)

warn_fig=warndlg('sad face=50%; happy forehead=60%; sad eyes=60%; angry lips=60%, t
re it is SAD');
push=findobj(h_warn_fig,'style','pushbutton');
t(h_push,'units','pixels');
sh_pos=get(h_push,'position')
g_pos=get(h_warn_fig,'position');
int_loc=fig_pos(1:2)+push_pos(1:2)+push_pos(3:4)/2;
t(h_warn_fig,'userdata',point_loc)
t(0,'pointerlocation',point_loc,'userdata',h_warn_fig);

ile any(h_warn_fig==get(0,'children'))
  set (0,'pointerlocation',get(h_warn_fig,'userdata'));
  drawnow
d

 (coar>=0.99963 && coar<0.99971 && T<=108 && (metric>0.3221 && metric<0.41 || metri
35 && metric < 0.7238) && angle_in<=17)

warn_fig=warndlg('sad face=50%; happy forehead=60%; sad eyes=60%; happy lips=90%; t
re it is HAPPY');
push=findobj(h_warn_fig,'style','pushbutton');
t(h_push,'units','pixels');
sh_pos=get(h_push,'position')
g_pos=get(h_warn_fig,'position');
int_loc=fig_pos(1:2)+push_pos(1:2)+push_pos(3:4)/2;
t(h_warn_fig,'userdata',point_loc)
t(0,'pointerlocation',point_loc,'userdata',h_warn_fig);

ile any(h_warn_fig==get(0,'children'))
  set (0,'pointerlocation',get(h_warn_fig,'userdata'));
  drawnow
d

 (coar>=0.99963 && coar<0.99971 && T<=108 && (metric>0.42 && metric<0.5200 || metri
39 && metric < 0.7771) && angle_in<=17)

warn_fig=warndlg('sad face=50%; happy forehead=60%; angry eyes=70%; happy lips=90%,
fore it is HAPPY');
push=findobj(h_warn_fig,'style','pushbutton');
t(h_push,'units','pixels');
```

```
ısh_pos=get(h_push,'position')
.g_pos=get(h_warn_fig,'position');
ıint_loc=fig_pos(1:2)+push_pos(1:2)+push_pos(3:4)/2;
:t(h_warn_fig,'userdata',point_loc)
:t(0,'pointerlocation',point_loc,'userdata',h_warn_fig);

ıile any(h_warn_fig==get(0,'children'))
   set (0,'pointerlocation',get(h_warn_fig,'userdata'));
   drawnow
ıd


: (cqar>0.99971 && T>=109 && T<=117 && (metric>0.42 && metric<0.5200 || metric>0.72:
ıetric < 0.7771) && angle_in<=17)

 warn_fig=warndlg('angry face=50%; sad forehead=40%; angry eyes=70%; happy lips=60%,
:fore it is ANGRY');
 push=findobj(h_warn_fig,'style','pushbutton');
:t(h_push,'units','pixels');
ısh_pos=get(h_push,'position')
.g_pos=get(h_warn_fig,'position');
ıint_loc=fig_pos(1:2)+push_pos(1:2)+push_pos(3:4)/2;
:t(h_warn_fig,'userdata',point_loc)
:t(0,'pointerlocation',point_loc,'userdata',h_warn_fig);

ıile any(h_warn_fig==get(0,'children'))
   set (0,'pointerlocation',get(h_warn_fig,'userdata'));
   drawnow
ıd

: (coar>0.99971 && T>=118 && T<=187 && (metric>0.2500 && metric< 0.3220 || metric>=
 && metric<0.6434) && angle_in<=17)

 warn_fig=warndlg('angry face=50%; angry forehead=60%; happy eyes=60%; happy lips=9
:refore it is HAPPY');
 push=findobj(h_warn_fig,'style','pushbutton');
:t(h_push,'units','pixels');
ısh_pos=get(h_push,'position')
.g_pos=get(h_warn_fig,'position');
ıint_loc=fig_pos(1:2)+push_pos(1:2)+push_pos(3:4)/2;
:t(h_warn_fig,'userdata',point_loc)
:t(0,'pointerlocation',point_loc,'userdata',h_warn_fig);

ıile any(h_warn_fig==get(0,'children'))
   set (0,'pointerlocation',get(h_warn_fig,'userdata'));
   drawnow
ıd

: (coar>0.99971 && T>=109 && T<=117 && (metric>0.2500 && metric< 0.3220 || metric>=
 && metric<0.6434) && angle_in>44)

 warn_fig=warndlg('angry face=50%; sad forehead=40%; happy eyes=60%; sad lips=60%;
ıre it is SAD');
 push=findobj(h_warn_fig,'style','pushbutton');
:t(h_push,'units','pixels');
ısh_pos=get(h_push,'position')
.g_pos=get(h_warn_fig,'position');
```

```
›int_loc=fig_pos(1:2)+push_pos(1:2)+push_pos(3:4)/2;
:t(h_warn_fig,'userdata',point_loc)
:t(0,'pointerlocation',point_loc,'userdata',h_warn_fig);

ile any(h_warn_fig==get(0,'children'))
  set (0,'pointerlocation',get(h_warn_fig,'userdata'));
  drawnow
ıd

: (coar>=0.99963 && coar<0.99971 && T<=108 &&  (metric>0.2500 && metric< 0.3220 || r
•=0.5201 && metric<0.6434) && angle_in>44)

_warn_fig=warndlg('sad face=50%; happy forehead=60%; happy eyes=60%; sad lips=60%; t
›re it is HAPPY');
_push=findobj(h_warn_fig,'style','pushbutton');
:t(h_push,'units','pixels');
ısh_pos=get(h_push,'position')
.g_pos=get(h_warn_fig,'position');
›int_loc=fig_pos(1:2)+push_pos(1:2)+push_pos(3:4)/2;
:t(h_warn_fig,'userdata',point_loc)
:t(0,'pointerlocation',point_loc,'userdata',h_warn_fig);

ile any(h_warn_fig==get(0,'children'))
  set (0,'pointerlocation',get(h_warn_fig,'userdata'));
  drawnow
ıd

: (coar<=0.99962843 && T<=108 &&  (metric>0.42 && metric<0.5200 || metric>0.7239 &&
: < 0.7771) && angle_in>44)

_warn_fig=warndlg('happy face=60%; happy forehead=60%; angry eyes=60%; sad lips=60%
efore it is HAPPY');
_push=findobj(h_warn_fig,'style','pushbutton');
:t(h_push,'units','pixels');
ısh_pos=get(h_push,'position')
ig_pos=get(h_warn_fig,'position');
›int_loc=fig_pos(1:2)+push_pos(1:2)+push_pos(3:4)/2;
:t(h_warn_fig,'userdata',point_loc)
:t(0,'pointerlocation',point_loc,'userdata',h_warn_fig);

ile any(h_warn_fig==get(0,'children'))
  set (0,'pointerlocation',get(h_warn_fig,'userdata'));
  drawnow
ıd
```