AUTOMATED STUDENT'S ATTENDANCE SYSTEM

DIANE TENG SIM LEAN

ELECTRICAL & ELECTRONICS ENGINEERING
UNIVERSITI TEKNOLOGI PETRONAS

JUNE 2009

Automated Students Attendance System

by

Diane Teng Sim Lean

Dissertation submitted in partial fulfilment of
the requirements for the
Bachelor of Engineering (Hons)
(Electrical & Electronic Engineering)

JUNE 2009

Universiti Teknologi PETRONAS
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

Automated Students Attendance System

by

Diane Teng Sim Lean

A project dissertation submitted to the
Electrical & Electronic Engineering Programme
Universiti Teknologi PETRONAS
in partial fulfilment of the requirement for the
BACHELOR OF ENGINEERING (Hons)
(ELECTRICAL & ELECTRONIC ENGINEERING)

Approved by,

____
PN LILA IZNITA IZHAR

UNIVERSITI TEKNOLOGI PETRONAS

TRONOH, PERAK

June 2009

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.

____

DIANE TENG SIM LEAN

ABSTRACT

The Automated Students' Attendance System is a system that takes the attendance of students in a class automatically. The system aims to improve the current attendance system that is done manually. This work presents the computerized system of automated students' attendance system to implement genetic algorithms in a face recognition system. The extraction of face template particularly the T-zone (symmetrical between the eyes, nose and mouth) is performed based on face detection using specific HSV colour space ranges followed by template matching. Two types of templates are used; one on edge detection and another on the intensity plane in YIQ colour space. Face recognition with genetic algorithms will be performed to achieve an automated students' attendance system. With the existence of this attendance system, the occurrence of truancy could be reduced tremendously.

ACKNOWLEDGEMENTS

It has taken me one year to come this far. And throughout this whole duration, I've learnt a lot and gained priceless experiences. I would like to acknowledge the following people for their help, advice, support and encouragement throughout this duration. Without them, this report could never have been written with so much content.

I could never have come so far without the constant supervision and support of my supervisor, Pn Lila Iznita Izhar. Thank you for your constant advice and encouragement throughout this whole year. Your help is greatly appreciated.

To my family, my parents and brother, thank you for your constant care and support. You have been with me throughout, the bad times and the good times. Your unending love and support is invaluable. I will always love you all.

To my friends both from UTP and elsewhere, thanks for always being there for me. Thank you for making me laugh whenever I was down and always bringing me up again. We have shared our many of our experience with each other. Our times together will always be kept close to my heart.

Finally, to all who have helped me in one way or another, many thanks!

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND OF STUDY

Face recognition is one of the most active research issues in the field of pattern recognition and has gained a lot of attention in the past few decades. With the vast advancements in this topic, research still needs to be done to improve on the efficiency, accuracy and practicality of the many methods produced. The genetic algorithm has been chosen to be implemented into face recognition for this project.

Genetic algorithms are categorized as global search heuristics. The genetic algorithm is a search technique used in computing to find true or approximate solutions to optimize and search problems. It uses techniques inspired by evolutionary biology such as inheritance, mutation, selection, and crossover (also called recombination). The idea behind genetic algorithm is to extract optimization strategies nature uses successfully, known as Darwinian Evolution [2]. It then transforms them for application in mathematical optimization theory to find the global optimum in a defined phase space.

1.2 HISTORY OF FACE RECOGNITION

Face recognition started from the moment machine became more and more "intelligent" and had the advance of fill in, correct or help the lack of human abilities and senses.

The subject of face recognition is as old as computer vision and because of the practical importance of the topic and theoretical interest from cognitive science. Face recognition is not the only method of recognizing other people. Even humans between each other use senses in order to recognize others. Machines have a wider range for recognition purposes, which use features such as fingerprint, voice, or iris recognition. Despite the fact that these methods of identification can be more accurate, face recognition has always remains a major focus of research because of its non-invasive nature or simply because it is a human's primary method of person identification.

Since the start of this field of technology there were two main approaches. The two main approaches to face recognition are geometrical approach and pictorial approach.

The geometrical approach uses the spatial configuration of facial features. That means that the main geometrical features of the face such as the eyes, nose and mouth are first located and then faces are classified on the basis of various geometrical distances and angles between features. On the other hand, the pictorial approach uses templates of the facial features [5]. That method is using the templates of the major facial features and entire face to perform recognition on frontal views of faces. Many of the projects that where based on those two approaches have some common extensions that handle different poses backgrounds. Apart from these two techniques we have other recent template-based approaches, which form templates from the image gradient, and the principal component analysis approach, which can be read as a sub-optimal template approach [5]. Finally we have the deformable template approach that combines elements of both the pictorial and feature geometry approaches and has been applied to faces at varying pose and expression.

Since the early start of face recognition there is a strong relation and connection with the science of neural networks. Neural networks are going to be analyses in more detail later on. The most famous early example of a face recognition "system", using neural networks is the Kohonen model. That system was a simple neural network that was able to perform face recognition for aligned and normalized face images. The type of network he employed computed a face description by approximating the eigenvectors of the face image's auto-correlation matrix; these eigenvectors are now known as "eigenfaces" [9].

After that there were more other methods that where developed based on older techniques. If we want to summarize the methods that the "idea" of face recognition is based we have a geometrical approach or pictorial approach, and after that we have methods like eigenfaces, Principal Component Analysis, or other methods that process images in combination with neural networks or other expert systems [10].

In the recognition stage, the input is compared against all selected model views of each person. To compare the input against a particular model view, the face is first geometrically aligned with the model view. An affine transform is applied to the input to bring the facial features automatically located by the system into correspondence with the same features on the model. A technique based on the optical flow between the transformed input and the model is used to compensate for any remaining small transformation between the two. Templates from the model are then compared with the image using normalized correlation. Both the model and input images are pre-processed with a differential operator such the ones mentioned just above. In the future, we plan to address the problem of recognizing faces when only one view of the face is available. The key to making this work will be an example-based learning system that uses multiple images of prototype faces undergoing changes in pose to learn, with the help of neural networks. The system will apply this knowledge to synthesize new virtual views of the person's face.

1.3 FACE RECOGNITION TODAY

With the rapid evolution of the technology and the commercialization of technological achievements, face recognition became more and more popular, not only for research but also for the use of security systems.

That gave the motive to many researchers, and also companies in order to develop techniques for automatically recognizing faces that would find many applications, including security and human-computer interaction. For instance, a face recognizing machine could allow automated access control for buildings or enable a computer to recognize the person sitting at the console. Most existing face recognition systems, however, work only for frontal or nearly frontal images of faces. By recognizing faces under varying pose, one makes the conditions under which face recognition systems operate less rigid.

1.4 PROBLEM STATEMENT

With the number of truancy worldwide rising incredibly in the past decade, this automated students attendance system aims to improve the condition. In Universiti Teknologi PETRONAS itself, attendance is taken manually where a sheet of attendance paper is passed around the class and students are required to put their signature next to their own names as proof of their attendance. Despite the fact that this system is able to save the lecturer from spending extra time taking class attendance thus enabling him or her to concentrate fully on the lesson, it has its flaws. Students are able to forge their friends' signatures easily without the lecturers' knowledge. With this, the accuracy of this attendance system is greatly affected. To overcome this problem, automated students' attendance system aims to give higher accuracy and efficiency. This automated system targets to use as little image for one person as possible for the database, the best case scenario being one image per person. This is taken into consideration because the database would be big as this system is targeted to be implemented in universities and schools where there are large numbers of students.

Problems faced in face recognition includes variations in pose, illumination and expression [11]. These three factors influenced the results of the work greatly and thus to avoid these problems, this system will be placed in a controlled environment where illumination will be fixed and students will be required to look straight into the camera with a neutral expression to ensure that only frontal and expressionless faces will be analysed by the system.

1.5 OBJECTIVES

The objective of this project is to recognize faces using genetic algorithms. The developed MATLAB program will be able to verify the faces of students in a class by matching the face captured by the camera to the database of faces stored. By doing so, the attendance of the students in the class will be taken. This concept will be implemented into the Automated Students Attendance System.

In order to achieve this objective, a series of research have been conducted. Different techniques used in face recognition were analysed and compared. Knowledge on the MATLAB language and its tools are very essential in this project. Cameras also need to be interfaced with the program for this system to work effectively.

1.6 SCOPE OF STUDY

The scope of study for this project is large, covering face detection, genetic algorithms, face recognition using genetic algorithms, and utilizing Matlab image processing tools.

1.7 REPORT FORMAT

This paper includes the following chapters.

Chapter 2 : Literature Review
Chapter 3 : Methodology
Chapter 4 : Results and Discussions
Chapter 5 : Conclusion and Recommendations

CHAPTER 2

LITERATURE REVIEW

2.1 FACE RECOGNITION USING GENETIC ALGORITHM BASED TEMPLATE MATCHING

This paper presents face recognition using a T-shaped template (symmetrical between eyes, nose and mouth) to match the faces in the database with the help of a genetic algorithm. The algorithm automatically tests several positions around the target and adjusts the size of the template as the matching process progresses. Two kinds of templates are used here, one based on edge detection and the other on YIQ colour information on the face [2]. This method used a maximum of only one image per person stored in the database.

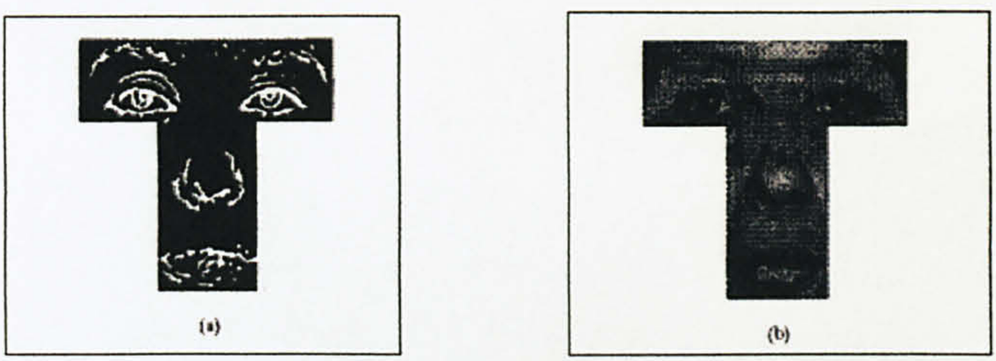


Figure 1 Templates based on (a) edge detection and (b) YIQ colour information

This system is based on relative positions of facial features from each other. For now, only frontal faces and preferably expressionless ones have been considered. Skin detection is performed to determine the face position. The position of the facial features is later extracted. Two filters are used after this. The first filter applied is the Laplacian of Gaussian to smoothen and detect the edges of the facial features. A lot of noise is produced with this filter. Therefore, a 3x3 medium filter is then used to reduce this noise. From this result, a T-shaped template is extracted.

The template is then compared to the ones in the database. Its initial position is set as the top right edge of the image. The genetic algorithm adjusts the size of the template continuously until the fitness of the genetic algorithm is above a set of threshold [2].

Length of chromosomes	= 24 bits
First 6 bits	= width and height of template
Remaining 12 bits	= accuracy of matching process using edge template (6 bits) and color template (6 bits)

The database used for this paper contained 267 front view face images taken under four different illumination settings. The database was constructed by manually extracting the templates from the images after edge detection was performed. 97 persons were involved in the making of this database. The results showed 95.1% recognition accuracy in this method [2]. Speed was not considered in this work. The recommendation given in this paper was to combine genetic algorithm and distance measure method to a hybrid, also including other methods like neural network and wavelets.

2.2 FEATURE SELECTION FOR FACE RECOGNITION USING A GENETIC ALGORITHM

This paper aimed to select the most useful features for face recognition. Recognition is done using interest points extracted from the detected faces. Lowe's SIFT features were used for this strategy [1]. The matching criterion used is based on the Euclidean distances between the key points on the detected face and the key points on the image in the database [3]. Genetic algorithm is used to reduce the features until the most useful features are left. So far, no experiments have been conducted to show that the selected features are the best for face recognition yet.

2.3 THE DESIGN OF A COMPOSITE WAVELET MATCHED FILTER FOR FACE RECOGNITION USING BREEDER GENETIC ALGORITHM

This paper used the breeder genetic algorithm to design a composite wavelet matched filter for face recognition. The breeder genetic algorithm is different from the genetic algorithm in that it uses artificial selection, where only the top T% of the fittest individuals are selected as parents to produce the next generation [4]. The filter is used in an optical correlator in the VanderLugt architecture to separate two classes of input face images. To design this filter, the Mexican Hat and Morlet wavelets were used. The performance of this filter has been evaluated in the presence of additive white Gaussian noise. The filter designed using a Mexican hat wavelet was found to perform better than the filter designed using a Morlet wavelet because it focuses on the lower spatial frequency components of the target images [4]. The noise performance was also studied and the performance of the designed filter was found to perform better for high values of input signal to noise ratio.

2.4 3D FACE RECOGNITION BY CONSTRUCTING DEFORMATION INVARIANT IMAGE

In this paper, a proposition of a novel deformation invariant image for robust 3D face recognition is done. Firstly, the depth image and the intensity image is obtained from the original 3-dimensional facial data. Geodesic level curves are then generated by constructing radial geodesic distance image from the depth image. Finally, a deformation invariant image is constructed by evenly sampling points from the selected geodesic level curves in the intensity image. The experiments done in this work were based on the 3D CASIA Face Database which includes 123 individuals with complex expressions. The experimental results obtained showed that the proposed method substantially improves the recognition performance under various facial expressions.

2.5 THE EFFECT OF IMAGE ENHANCEMENT ON BIOMEDICAL PATTERN RECOGNITION

This paper examined the quantitative effect of image enhancement on biomedical pattern recognition. Wavelet-based image enhancement techniques developed earlier [12] were applied to a well known biomedical pattern recognition problem; chromosome classification. In this work, experiments were conducted on a test set of chromosome images before and after the enhancement, using the same feature measurement and classifier methods. Test results showed that their image enhancement method substantially reduced the error rate of chromosome classification from 15.6% to 8.5%.

2.6 SALT-AND-PEPPER NOISE DETECTION AND REDUCTION USING FUZZY SWITCHING MEDIAN FILTER

This paper presents a new fuzzy switching median filter. This filter employs fuzzy techniques in image processing. It is able to remove salt-and-pepper noise in digital images while preserving image details and textures very well. This recursive fuzzy switching median filter is an extension to the classical switching median filter. To do this, it employs fuzzy inference mechanism. Unlike some filtering mechanisms which require iterations, and thus needs lengthy processing time, this filter only needs to be applied once and has a very efficient computational time.

2.7 CONTRAST ENHANCEMENT WITH HISTOGRAM-ADAPTIVE IMAGE SEGMENTATION

This approach aimed at the improvement of the visual quality of underexposed or low-contrast images. In this paper, a new contrast-enhancement algorithm was developed based on the segmentation of the image area with relatively high density of dark elements. The brightness intervals of the selected segments are changed followed by equalization of the corresponding parts of the histogram. This method was proven to be effective with experiments done on many test images. The advantages of this new method include:

- the low computational complexity and the small number of parameters, which values are set as a result of the image histogram analysis;
- the universal use, for example – for underexposed and low-contrast images;
- for successful contours extraction (reduction of the extracted contours).

CHAPTER 3

METHODOLOGY

There are three main stages involved in performing the face recognition; face region detection, template extraction and face recognition using genetic algorithm. Face region detection can be divided into four sub-stages, which are image acquisition, face detection, straightening of face and cropping of face. The second stage, template extraction can be divided into two sub-stages, eyes and mouth detection and template extraction respectively. These stages are summarized in the flowchart in Figure 2.

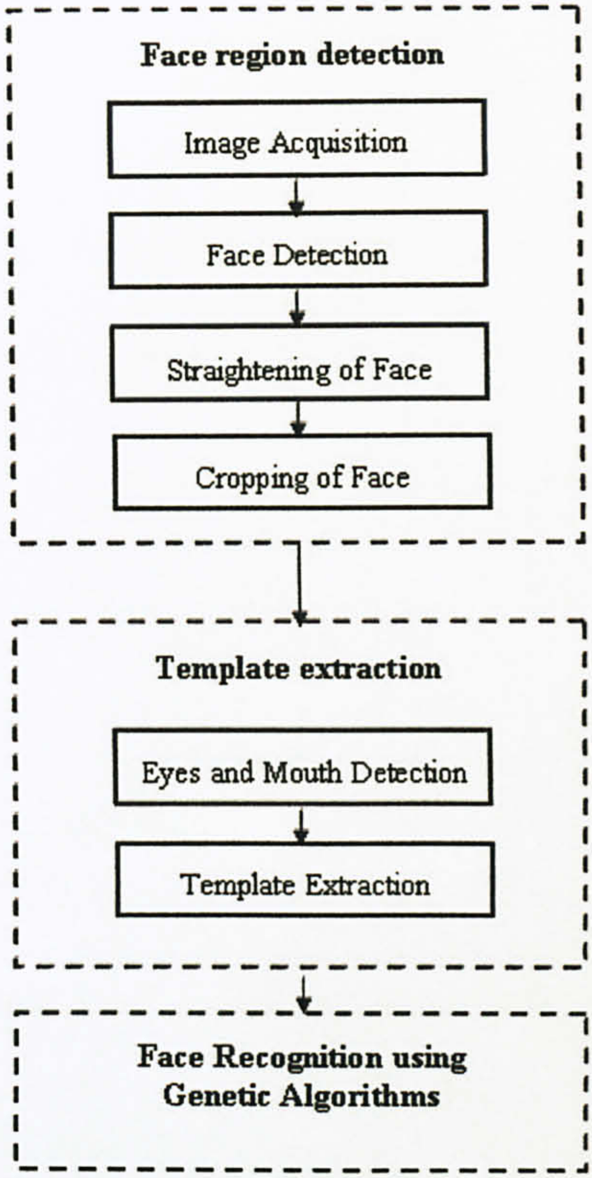


Figure 2 Flowchart of Project Methodology

3.1 FACE REGION DETECTION

Image is obtained by interfacing a web camera with Matlab using the built-in function "videoinput". The input image is of eight bit depth of size 320 X 240 and is of RGB type.

The face region is then found using specified HSV color space ranges. The ranges for H are specified to be less than 0.10 and more than 0.90 while the ranges for S are more than 0.20 and less than 0.99.

The detected face is first pre-processed by removing unwanted noise and enhancing the contrast. Noise is removed by passing the image through a 3x3 median filter. The median filter replaces the middle pixel of the mask with the median value of all the other pixels in the mask. The median filter is used because it is an effective neighborhood averaging method. The median filter is an effective method that can suppress isolated noise without blurring sharp edges. Specifically, the median filter replaces a pixel by the median of all pixels in the neighborhood:

$$y[m,n] = median\{x[i,j], (i,j) \in \omega\}$$

where ω represents a neighborhood centered around location in the image.

The Contrast Limited Adaptative Histogram Equalization (CLAHE) method is used to enhance its contrast. CLAHE is a generalization of adaptive histogram equalization and ordinary histogram equalization. The reason CLAHE was chosen as the contrast enhancement technique is because it operates on small regions in the image, called tiles, rather than the entire image. Each tile's contrast is enhanced separately. The neighboring tiles are then combined using bilinear interpolation to eliminate artificially induced boundaries. The contrast, especially in homogeneous areas, can be limited to avoid amplifying any noise that might be present in the image.

Thresholding is done on the pre-processed image to obtain a binary image. Otsu's method for thresholding is used. This method chooses the threshold to minimize the intra-class variance of the black and white pixels. This is defined as a weighted sum of variances of the two classes:

$$\sigma_w^2(t) = q_1(t)\sigma_1^2(t) + q_2(t)\sigma_2^2(t)$$

Weights ωi are the probabilities of the two classes separated by a threshold t and variances of these classes.

Morphological operations are performed on the regions. Erosion is performed to disconnect regions such as the spectacles and the eyes or the eyes and the nose. Dilation is done to connect separated regions like the whites of the eyes and the pupils of the eyes. By removing all regions with sizes of less than 230 and more than 500 pixels, most of the unwanted regions are removed. Regions which are non-horizontal, that is, more than 30 or less than -30 degrees are eliminated, leaving only the eyes and mouth regions.

The center of mass of the remaining regions, that is, the eyes and mouth regions are then calculated. The results of this calculation are called centroids. The first element of centroid is the horizontal coordinate (or x-coordinate) of the center of mass, and the second element is the vertical coordinate (or y-coordinate).

By rotating the image based on the centroids on both of the eyes at an angle so that both centroids on a straight horizontal line, a straight-face image is obtained. To perform this operation, the formula to obtain the angle of rotation in radian is as shown below.

$$angle = \tan^{-1}\left(\frac{y_1 = y_0}{x_1 - x_0}\right)$$

where left eye = (x_0, y_0)

and right eye = (x_1, y_1)

The final step of this section is to crop the face once again so that the eyes and the mouth form the boundaries of the image.

3.2 TEMPLATE EXTRACTION

From the centroids on both eyes and mouth, a skeleton-T shape is formed by joining the centroids in a two straight lines. Bounding boxes are then drawn around the detected eyes and mouth regions using the "regionprops" built-in MATLAB function. The Skeleton-T shape is then expanded according to the height of the bounding boxes surrounding the eyes and the width of the bounding box surrounding the mouth. These parameters are computed automatically from the bounding boxes obtained. This forms an expanded-T shape which covers the eyes and the mouth accurately.

The expanded-T is complemented and added to the face image to extract T-templates. Two T-templates are extracted from the face image, one with edge detection and one with the intensity plane. The reason edge detection is done is because the process significantly reduces the amount of data in the image, while preserving the most important structural features of that image. The Canny's detection method is chosen because of its relatively low error rate compared to the other edge detection methods [18]. The intensity plane is an important part of the image and will be useful in the template matching process later on.

3.3 FACE RECOGNITION USING GENETIC ALGORITHM

This part is to be conducted using genetic algorithm. Genetic algorithm will be used to match the T-Templates to perform face recognition. Research is currently being done on how Genetic Algorithms can be implemented in image correlation to match the two templates with templates of images in the database.

3.4 SOFTWARE AND HARDWARE

Matlab will be used to program and simulate the face recognition system. To capture the image of the subject, a web camera will be used.

CHAPTER 4

RESULTS AND DISCUSSIONS

4.1 RESULTS

4.1.1 Face Region Detection

Figure 3 shows the 320 X 240 RGB image obtained through a web camera.

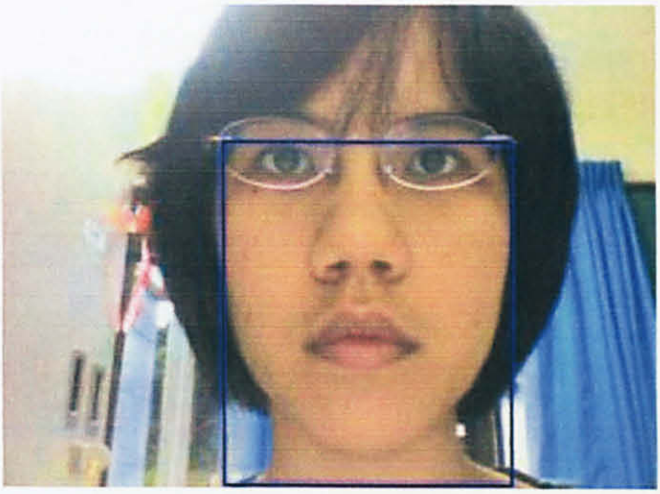


Figure 3 Image Taken by Interfacing MATLAB with Web Camera

The detected face is shown in Figure 4 below.

Figure 4 Detected Face

The pre-processed images are shown in Figures 5 and 6. Figure 5 shows the image passed through a 3 X 3 median filter. Figure 6 shows an image contrast enhanced using the Contrast Limited Adaptative Histogram Equalization (CLAHE) method.

Figure 5 Face Image Median Filtered

Figure 6 Face Image Contrast Enhanced

In Figure 7 is the binary face image after it has gone through the Otsu's thresholding process.

Figure 7 Face Image Thresholded

Figure 8 shows the image with the regions eroded using a disk structural element with a diameter of one pixel. Figure 9 shows the regions labeled with random colours after erosion.

Figure 8 Image with Regions Eroded

Figure 9 Image with Eroded Regions Labeled

Regions which are not horizontal are removed. Other regions with sizes less than 180 pixels and more than 500 pixels are also removed, leaving the eyes and mouth regions as shown in Figure 10. Figure 11 shows the three regions labeled with different colours.

Figure 10 Eyes and Mouth Regions

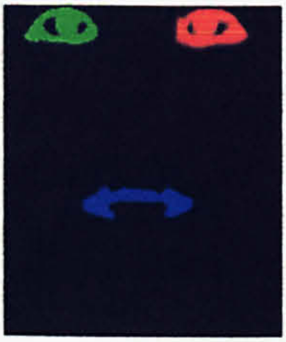


Figure 11 Eyes and Mouth Regions Labeled

Centroids which represent the center of mass of each region are placed on the eyes and mouth as shown in Figure 12.

Figure 12 Centroids Placed on Eyes and Mouth

By performing rotation so that both centroids on the eyes are on a horizontal line, both face images, binary and RGB, are straightened as shown in Figure 13 and 14 respectively.

Figure 13 Binary image of Face Straightened

Figure 14 RGB Face Image Straightened

The face image is once again cropped based on the binary image in Figure 13 so that the eyes and the mouth form the boundaries of the image. The resulting image is shown in Figure 15.

Figure 15 Face Image Cropped with Eyes and Mouth as Boundaries

4.1.2 Template Extraction

By joining the centroids on both eyes and mouth, skeleton-T shape is formed as shown in Figure 16.

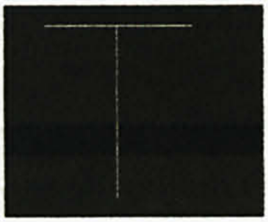


Figure 16 Skeleton-T Shape

Figure 17 shows the bounding boxes drawn around the detected eyes and mouth regions.

Figure 17 Bounding Boxes Around Eyes and Mouth Regions

Figure 18 shows the skeleton-T shape transformed into an expanded-T shape.

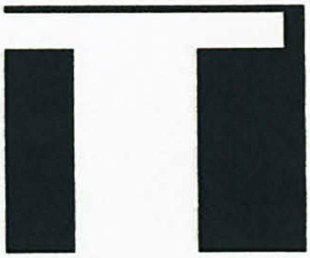


Figure 18 Expanded-T

In Figure 19 are the two templates extracted from the face image. Figure 19(a) shows the template extracted based on edge detection while Figure 19(b) shows the template extracted based on the intensity plane.

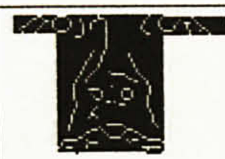
Figure 19 (a) Edge Detection T-Template (b) Intensity Plane T-Template

4.1.3 Performance Analysis

Experiment was done to measure the accuracy in extracting T-templates from a set of test images. Test images consist of frontal and expressionless faces. The table below shows the results obtained in this experiment. From the observation made, the system was found to be accurate in extracting T-templates.

Table 1 Results of Experiment to Measure Accuracy in Extracting T-Templates

No.	Input Image	Detected Face	T-Templates	
			Edge Detection	Intensity Plane
1				
2				
3				
4				

5				
6				
7				
8				
9				
10				

11				
12				
13				
14				
14				

15			nil	nil

Based on the performance analysis in the table above, it is estimated that the system is 93.33% accurate.

4.2 DISCUSSIONS

There was initially some minor problems with acquiring the image using a web camera. The image taken was later found to be in YUV colour space. Therefore, converting the image from YUV to RGB solved the problem.

Getting the average range of HSV colour space from a collection of skin images, the range was determined and pre-defined. The average ratio of the dimensions of the face was also pre-defined. The face region was then determined with reference to the pre-defined parameters.

It is essential to pre-process images before performing any other steps on them ensure better results. The importance of this process is more evident in noisy images where the results of the face detection may be affected. The procedures which play a big role in this process have been identified to be filtering and contrast enhancing. Filtering would remove noise in an image while contrast enhancing, as the name suggests, enhances the contrast in the image to reduce the difficulty of detecting the face areas.

Some experimentation was done in this area. Low pass filters tend to blur the images. Thus, this filter was eliminated from the list. High pass filters sharpened the images, including the noise. This was not suitable and so the high pass filter was not useful in our case.

The median filter was found to do a better job compared to the low pass filter and the high pass filter in preserving useful details in the images because it completely eliminates the noise. The median filter considers each pixel in the image in turn and looks at its nearby neighbors to decide whether or not it is representative of its surroundings. However, instead of simply replacing the pixel value with the *mean* of neighboring pixel values, it replaces it with the *median* of those values. When taking the median, only the color value of one or two healthy pixels was kept. Besides eliminating unwanted noise, the median filter also maintains the sharpness of the image. Thus, the median filter was used for the purpose of noise removal.

Contrast enhancement increases the total contrast of an image by making light colours lighter and dark colours darker at the same time. Two methods were compared here, i.e. automatic contrast stretching and contrast-limited adaptive histogram equalization (CLAHE). The result of CLAHE method was found to be more satisfying as the contrast was more apparent. This is because the CLAHE method analyses regions at a time instead of the whole image at once.

Regions are eroded to separate joined regions like the eyebrows and eyes, hair and eyebrows, spectacles and eyes. Structuring elements used in this process are disk and rectangle. The disk was found to separate regions from different angles while the rectangle when used horizontally, separates regions that are joined vertically.

Regions are labeled by the 8-connectivity method and the area of each region is computed. This is done as a preface to the region removal stage.

The big regions which contain of more than 400 pixels and very small regions which contain less than 20 pixels are first removed. The image is then dilated to join some disjointed regions before further removing regions that are less than 200 pixels in size. After this, regions that are not horizontal, i.e. more than 32 degrees or less than -32 degrees are removed. The reason for this step is because the eyes and mouth are more or less horizontal.

Dilation is done to join disjointed objects and to close up holes in regions. This is especially essential when the whites of the eyes are disconnected by the pupils.

The face region with the eyes and mouth as the border is cropped out based on the binary image obtained from the removal of the unnecessary regions earlier.

The cropped face region is once again thresholded, filtered and contrast enhanced. After obtaining wanted features by eliminating other regions, the regions of the features are grouped up by dilating the image with a line structuring element having a length of 12 pixels. Dilation is done to join up regions of eyes, nose and mouth. Other regions are further eliminated, leaving the grouped up regions. The features are marked with bounding boxes.

T-templates are extracted from the face image in a manner that the outline of the template fits the bounding boxes accurately. This is done with the calculation of the height of the bounding boxes of the eyes and the width of the bounding box of the mouth. The skeleton-T is dilated with a rectangular structural element which has the dimensions obtained earlier.

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 CONCLUSION

Up to this point, work has been done until T-template extraction and will be continued for face recognition using Genetic Algorithms. A 320 X 240 image was obtained through the web camera and pre-processed by passing it though a 3 X 3 median filter followed by contrast enhancing. Thresholding was then performed on the image to obtain a binary image. Unwanted regions were removed, leaving only the eyes and mouth regions. The face is then straightened based on the centroids on both the eyes. The face region with the eyes and mouth as its boundaries was cropped automatically. Based on the bounding boxes around eyes and mouth regions, an expanded-T shape was formed. Two T-templates, one based on edge detection and another on the intensity plane were extracted from the image.

5.2 RECOMMENDATIONS

Genetic algorithm will be implemented for the matching process where the T-templates extracted from the face image will be compared with the T-templates of the images in the database. Research is being done on implementing genetic algorithm into image correlation to perform face recognition.

REFERENCES

[1] David G. Lowe, Distinctive image fearutes from scale-invariant keypoints, *International Journal of Computer Vision,* 60, 2 (2004), pp. 91-110

[2] Stephen Karungaru, Minoru Fukumi, Norio Akamatsu, Face Recognition using Genetic Algorithm based Template Matching, *International Symposium on Communications and Information Technologies* (2004), pp.1252-1257

[3] Derya Ozkan, Feature Selection For Face Recognition Using A Genetic Algorithm, *Bilkent University, Department of Computer Engineering, Turkey, Ankara*

[4] Aloka Sinha, Kehar Singh, The Design of a Composite Wavelet Matched Filter for Face Recognition Using Breeder Genetic Algorithm, *Department of Physics, Indian Institute of Technology Delhi, New Delhi-110016, India*, 2005

[5] R. Brunelli and T. Poggio, "Face Recognition: Features versus Templates", IEEE Trans. on PAMI, 1993.

[6] Heseltine, T.; Pears, N. and Austin, J.. "Three-dimensional face recognition using combinations of surface feature map subspace components". 2008

[7] Kakadiaris, I. A.; Passalis, G. and Toderici, G. and Murtuza, N. and Karampatziakis, N. and Theoharis, T.. "3D face recognition in the presence of facial expressions: an annotated deformable model approach". 2007

[8] Queirolo, C. C.; Silva, L.; Bellon, O. R. and Segundo, M. P. "3D Face Recognition using Simulated Annealing and the Surface Interpenetration Measure".2009.

[9] P. Belhumeur, J. Hespanha, and D. Kriegman. "Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection". 1997

[10] M. Turk and A. Pentland. "Face recognition using eigenfaces". Vision and Modeling Group, The Media Laboratory Massechusetts Institute of Technology. 1991

[11] Kailash J. Karande, Sanjay N.Talbar. "Face Recognition under Variation of Pose and Illumination using Independent Component Analysis". Department of Information Technology, Sinhgad Institute of Technology, Lonavala, Pune, Maharashtra State, India. 2008.

[12] Qiang Wu; Yu-Ping Wang; Zhongmin Liu. Tiehan Chen; Castleman, K.R.; "The effect of image enhancement on biomedical pattern recognition" [Engineering in Medicine and Biology, 2002. 24th Annual Conference and the Annual Fall Meeting of the Biomedical Engineering Society] EMBS/BMES Conference, 2002. Proceedings of the Second Joint. Volume 2, 23-26 Oct. 2002 Page(s):1067 - 1069 vol.2

[13] Jinshan Tang; Qingling Sun; Agyepong, K.. "An Image Enhancement Algorithm Based on a Contrast Measure in the Wavelet Domain for Screening Mammograms". Image Processing, 2007. ICIP 2007. IEEE International Conference on Volume 5, Sept. 16 2007-Oct. 19 2007 Page(s):V - 29 - V - 32

[14] Y. Wang, Q. Wu, K. R. Castleman, Z. Xiong, "Image enhancement using multiscale differential operators". Proc. ICASSP 2001. Salt Lake City. 2001.

[15] Toh, K.K.V.; Ibrahim, H.; Mahyuddin, M.N. "Salt-and-pepper noise detection and reduction using fuzzy switching median filter". Volume 54, Issue 4, November 2008 Page(s):1956 - 1961

[16] Rubin, S.H.; Kountchev, R.; Todorov, V.; Kountcheva, R.. "Contrast Enhancement with Histogram-Adaptive Image Segmentation". Information Reuse and Integration, 2006 IEEE International Conference on 16-18 Sept. 2006 Page(s):602 - 607

[17] Polesel, A.; Ramponi, G.; Mathews, V.J.. "Adaptive unsharp masking for contrast enhancement". Image Processing, 1997. Proceedings., International Conference on Volume 1, 26-29 Oct. 1997 Page(s):267 - 270 vol.1 Digital Object Identifier 10.1109/ICIP.1997.647756

[18] Ikhlas Abdel-Qader; Osama Abudayyeh; Michael E. Kelly. "Analysis of Edge Detection Techniques for Crack Identification in Bridges". J. Comp. in Civ. Engrg.. 2003. Volume 17, Issue 4, pages 255-263

APPENDICES

APPENDIX A : MATLAB CODES FOR FACE DETECTION SYSTEM

```
%%==========================================================================
%              Automated Students' Attendance System
%                          written by
%                     Diane Teng Sim Lean
%%==========================================================================
%%
% 1. Obtain database image if none yet
button = questdlg( 'Do you already have an image in the database?' );
if strcmp( button, 'No' ) == true
    [image_face] = capture_image();
end

%%
% 2. Obtain face image through web camera
[image_face] = webcam_getface();

%%
% 3. Obtain face area
[image_dilate2, image_face] = obtain_face_area(image_face);

%%
% 4. Straighten face area
[straightened_face, straightened_bw] = straightened_face_area(image_face,
image_dilate2);

%%
% 5. Crop out face area
[image_cropped, bw_cropped] = crop_face(straightened_face, straightened_bw);

%%
% 6. Mark features on face and obtain T-template
[t_template, image_cropped] = obtain_t_template(image_cropped, bw_cropped);

%%
% 7. Extract T-template of edge detected image and intensity plane of image
[T_edge, T_intensity] = extract_t_template(t_template, image_cropped);
```

```
%==========================================================================
%                     Saves images into face database
%==========================================================================
%%
function [image_face] = capture_image();
%%
button = 'No';

while strcmp( button, 'No' ) == true
% 1. Obtain face image through web camera
[image_face] = webcam_getface();

button = questdlg( 'Do you really want to save this image?' );
end

    i = 1;
    present = 0;
    if (((exist( 'Face_images', 'dir') == 7 ) ~= 1) )
        mkdir( 'Face_images' );
    end

    filename = input( 'Save image as:','s' );


        str = strcat( filename, '.jpg' );
        imwrite(  image_face, ['Face_images/' str] );

    msgbox( strcat( 'Image has been written as...', filename, '.jpg...in
Face_images folder' ), 'Saved image', 'help' )
    close all;
```

```
%=========================================================================
%    Captures image through web camera and performs face detection
%=========================================================================

function [image_face] = webcam_getface()

%% -----------------------------------------------------------------------
% Defining parameters (face)
%===========================

% Skin color (hue and saturation values)
skin_color_H_min = 0.10;
skin_color_H_max = 0.90;
skin_color_S_min = 0.20;
skin_color_S_max = 0.99;

% Color (white) for marking skin colored areas
skin_color = [0 0 1];

% Maximum size of noise removed
noise_face_max = 500;

% Face dimensions relative to the scene dimensions (approximated)
face_height_min = 0.15;
face_height_max = 0.75;
face_width_min = 0.10;
face_width_max = 0.75;

% Scene dimensions
scene_width = 320;
scene_height = 240;

% Convert face dimensions relative to absolute dimensions
face_height_min = face_height_min*scene_height;
face_height_max = face_height_max*scene_height;
face_width_min = face_width_min*scene_width;
face_width_max = face_width_max*scene_width;

%% -----------------------------------------------------------------------
% Connecting webcam to Matlab
%==============================

% Initializing webcam
obj = videoinput( 'winvideo', 1 );

%Change captured image's color space from YUV to RGB
obj.ReturnedColorSpace = 'rgb';

% Set parameters for webcam
set( obj, 'Tag', 'motion detection setup' );
set( obj, 'TriggerRepeat', Inf );
obj.FrameGrabInterval = 30;

%% -----------------------------------------------------------------------
% Detecting face region
%=======================

    a = 0;
       while( a<10 )
         image = getsnapshot(imaqgate('privateGetField', obj, 'uddobject'));

         % Convert image color space from RGB to HSV
            RGB = im2double( image );
```

```
        HSV = rgb2hsv( RGB );
        H = HSV( :, :, 1 );
        S = HSV( :, :, 2 );
        V = HSV( :, :, 3 );

        % Detect skin color in HSV-space
        image_skin_H = (H < skin_color_H_min) | (H > skin_color_H_max);
        image_skin_S = (S > skin_color_S_min) & (S < skin_color_S_max);
        image_skin = image_skin_H & image_skin_S;

        % Clean up skin image
        image_clean = bwareaopen( image_skin , noise_face_max );

        % Search all bounding boxes
        image_labeled = bwlabel( image_clean );
        region_props = regionprops( image_labeled, 'BoundingBox' );
        region_num = length( region_props );

        % Show image captured by webcam
        imshow( image ), title( 'Please face the camera approximately 30cm
from camera lense' );

        % Process each possible face in image
        for b = 1:region_num

            % Retrieve label parameters
            region_params = region_props(b).BoundingBox;
            box_x = floor(region_params(1))+1;
            box_y = floor(region_params(2))+1;
            box_width = floor(region_params(3))-3;
            box_height = floor(region_params(4))-3;

            % Matching feature properties for face detection
            face_height = (box_height < face_height_max) & (box_height >
face_height_min);
            face_width = (box_width < face_width_max) & (box_width >
face_width_min);
            face_ratio_land = (box_width > box_height) & (box_width <
2*box_height);
            face_ratio_port = (box_width < box_height) & (2*box_width >
box_height);

            % Processing regions that correspond to faces
            if (face_ratio_land || face_ratio_port) && (face_width &&
face_height)


              % Draw rectangle
              line( [box_x box_x+box_width], [box_y box_y], 'Color',
skin_color );
              line( [box_x box_x+box_width], [box_y+box_height
box_y+box_height], 'Color', skin_color );
              line( [box_x box_x], [box_y box_y+box_height], 'Color',
skin_color );
              line( [box_x+box_width box_x+box_width], [box_y
box_y+box_height], 'Color', skin_color );

                  image_face = image( box_y:box_y+box_height,
box_x:box_x+box_width, : );

          end
      end
      a = a+1;
```

```
end
imaqfind
figure, imshow( image_face ), title( 'Detected Face' );
```

```
%==========================================================================
%  Obtain face area with eyes and mouth as boundaries from detected face
%==========================================================================

function [image_dilate2, image_face] = obtain_face_area(image_face);

%% ------------------------------------------------------------------------
% Cropping face region with eyes and mouth as border
%==================================================

% Convert to grayscale, enhance, filter and threshold
image_grayscale = rgb2gray( image_face );
image_median = medfilt2( image_grayscale, [2 2] );
image_clahe = adapthisteq( image_median );
thresh_level = graythresh( image_clahe );
image_threshold = im2bw( image_clahe, thresh_level );
%image_threshold = im2bw( image_clahe, 0.45 );
image_threshold = imcomplement( image_threshold );
figure, imshow( image_median ), title( 'filtered' );
figure, imshow( image_clahe ), title( 'contrast enhanced' );
figure, imshow( image_threshold ), title( 'thresholded' );

%%
% Erode image to disconnect regions
se = strel('disk',1);
image_eroded = imerode(image_threshold,se);
%sel = strel( 'rectangle', [3 1] );
%image_eroded = imerode( image_eroded, sel );
figure, imshow( image_eroded ), title( 'eroded' );
%%

% Label regions
image_labeled = bwlabel( image_eroded );
color_regions= label2rgb(image_labeled, 'hsv', 'black', 'shuffle');
region_props = regionprops( image_labeled, 'BoundingBox' );
region_num = length( region_props )
figure, imshow( color_regions ), title( 'coloured regions' );

% Select very large and very small regions with pixels less than 20 and more
than 400
Stats_area = regionprops( image_labeled, 'area' );
All_area = [Stats_area.Area];
Maximum_size = max( max( All_area ) )
Minimum_size = min( min( All_area ) )
Search = find ( ( [Stats_area.Area] > 200 ) & ( [Stats_area.Area] < 540 ) );
image_area = ismember( image_labeled, Search );
figure, imshow( image_area ), title( 'selected regions (big & small removed)' );

%%
% Dilate image to connect disconnected regions
se2 = strel( 'disk', 1 );
image_dilate = imdilate( image_area, se2 );
figure, imshow( image_dilate ), title( 'dilated' );
image_label2 = bwlabel( image_dilate );
image_region = label2rgb(image_label2, 'hsv', 'black', 'shuffle');
figure, imshow( image_region ), title( 'color2' );

%%
% Remove small regions less than 200 pixels

Stats_area = regionprops( image_label2, 'area' );
All_area = [Stats_area.Area];
Search2 = find ( [Stats_area.Area] > 230 );
```

```
image_area2 = ismember( image_label2, Search2 );
figure, imshow( image_area2 ), title( 'selection2' );

%%
% Remove regions which are not horizontal
image_area2 = bwlabel( image_area2 );
Stats_angle = regionprops( image_area2, 'Orientation' );
All_angle = [Stats_angle.Orientation];
Search = find( ([Stats_angle.Orientation] < 30) & ([Stats_angle.Orientation] > -
30) );
regions_horizontal = ismember( image_area2,Search );
figure, imshow( regions_horizontal ), title( 'remaining horizontal regions' );
image_dilate2 = regions_horizontal;
%regions_horizontal = image_dilate2;
%%
% Dilate regions more
%se3 = strel( 'disk', 1);
%image_dilate2 = imdilate( regions_horizontal, se3 );
%figure, imshow( image_dilate2 ), title ( 'dilated2' );
```

```
%=============================================================================
%          Straighten face so that eyes are on a straight line
%=============================================================================

function [straightened_face, straightened_bw] =
straightened_face_area(image_face, image_dilate2)

figure, imshow( image_face ), title( 'centroids on features' )
hold on
image_labeled = bwlabel( image_dilate2 );
s  = regionprops(image_labeled, 'centroid');
centroids = cat(1, s.Centroid);
x_centroids = centroids(:,1);
y_centroids = centroids(:,2);
plot(x_centroids, y_centroids, 'w*')
hold off

%Get dimensions of image
[m n d] = size( image_face );
img_height = size(image_face, 1);
img_width = size(image_face, 2);
img_half_height = img_height/2;
left_eye = img_width/3;
right_eye = img_width/3*2;

%%
% Get centroid positions
number_centroids = size(x_centroids,1);
[rows1 cols1] = size(x_centroids);
for m = 1:rows1
   for n = 1:cols1
        x = x_centroids(m, n);
        y = y_centroids(m, n);
        if x <= left_eye
            x0 = round(x);
            y0 = round(y)
        elseif x >= right_eye
            x1 = round(x);
            y1 = round(y)
        else
            x2 = round(x);
            y2 = round(y);
        end
   end
end

x3 = (x0 + x1)/2;
y3 = (y0 + y1)/2;

%%
% Straighten face image
if y0 > y1
    l = x1 - x0;
    h = y1 - y0;
    rotation_angle_rad = atan( h/l );
    rotation_angle_degrees = 180*rotation_angle_rad/pi;
    straightened_face = imrotate( image_face, rotation_angle_degrees );
    straightened_bw = imrotate( image_dilate2, rotation_angle_degrees );
end

if y0 < y1
    l = x1 - x0;
    h = y1 - y0;
```

```
    rotation_angle_rad = atan( h/l );
    rotation_angle_degrees = 180*rotation_angle_rad/pi;
    straightened_face = imrotate( image_face, rotation_angle_degrees );
    straightened_bw = imrotate( image_dilate2, rotation_angle_degrees );
end

if y0 == y1
    straightened_face = image_face;
    straightened_bw = image_dilate2;
end

figure, imshow( straightened_bw ), title( 'Straightened BW of face' );
figure, imshow( straightened_face ), title( 'Straightened face' );
```

```
%=======================================================================
%                          Crop out face area
%=======================================================================

function [image_cropped, bw_cropped] = crop_face(straightened_face,
straightened_bw);

% Crop out area with selected regions

%crop from top
[row col] = size(straightened_bw);
edge = 1;
top = 0;
while edge
    top = top + 1;
    if any(straightened_bw(top,:))
        edge = 0;
    end
end
straightened_bw = imcrop (straightened_bw, [1 top  col-1 row-1]);
image_cropped = imcrop (straightened_face, [1 top  col-1 row-1]);

%crop from bottom

[row col] = size(straightened_bw);
edge = 1;
bottom = row + 1;
while edge
    bottom = bottom - 1;
    if any(straightened_bw(bottom,:))
        edge = 0;
    end
end
straightened_bw = imcrop (straightened_bw, [1 1 col-1 bottom-1]);
image_cropped = imcrop (image_cropped, [1 1 col-1 bottom-1]);

% Crop from left

[row col] = size(straightened_bw);
edge = 1;
left = 0;
while edge
    left = left + 1;
    if any(straightened_bw(:,left))
        edge = 0;
    end
end
straightened_bw = imcrop (straightened_bw, [left 1 col-1 row-1]);
image_cropped = imcrop (image_cropped, [left 1 col-1 row-1]);

%crop from right

[row col] = size(straightened_bw);
right = col + 1;
while edge
    right = right - 1;
    if any(straightened_bw(:,right))
        edge = 0;
    end
end
bw_cropped = imcrop (straightened_bw, [1 1 right-1 row-1]);
image_cropped = imcrop (image_cropped, [1 1 right-1 row-1]);
figure, imshow( image_cropped ), title( 'cropped face region' );
```

```
function [t_template, image_cropped] = obtain_t_template(image_cropped,
bw_cropped);

%-----------------------------------------------------------------------
% Marking Features
%==================

labeled_eyes_n_mouth = bwlabel (bw_cropped);
coloured_regions = label2rgb(labeled_eyes_n_mouth, 'hsv', 'black', 'shuffle');
figure, imshow( coloured_regions ), title( 'eyes & mouth' );


% Search all bounding boxes
region_props2 = regionprops( labeled_eyes_n_mouth, 'BoundingBox' );
region_num = length( region_props2 );

% Show image captured by webcam
figure, imshow( image_cropped ), title( 'eyes & mouth marked' )
hold on

%Get dimensions of image
[m n d] = size( image_cropped );
img_height = size(image_cropped, 1)
img_width = size(image_cropped, 2)
img_half_height = img_height/2;
left_eye = img_width/3;
right_eye = img_width/3*2;


% Mark eyes and mouth
for b = 1:region_num

    % Retrieve label parameters
       region_params2 = region_props2(b).BoundingBox;
       x = floor(region_params2(1))+1
       y = floor(region_params2(2))+1
       width = floor(region_params2(3))-3;
       height = floor(region_params2(4))-3;

    if x > img_width/5 && x < img_width*2/5
        t_width = width
    end

    if x < img_width/4 && y < img_half_height
        height_1 = height;
    end

    if x > img_width/2 && y < img_half_height
        height_2 = height;
    end

    % Draw rectangle
    rectangle ('Position', [x,y,width,height]);

end

s  = regionprops(labeled_eyes_n_mouth, 'centroid');
centroids = cat(1, s.Centroid)
x_centroids = centroids(:,1);
y_centroids = centroids(:,2);
plot(x_centroids, y_centroids, 'w*')
hold off
```

```
% Get centroid positions
number_centroids = size(x_centroids,1);
[rows1 cols1] = size(x_centroids);
for m = 1:rows1
   for n = 1:cols1
        x = x_centroids(m, n);
        y = y_centroids(m, n);
        if x <= left_eye
            x0 = round(x);
            y0 = round(y);
        elseif x >= right_eye
            x1 = round(x);
            y1 = round(y);
        else
            x2 = round(x);
            y2 = round(y);
        end
   end
end

x3 = x2;
y3 = (y0 + y1)/2;


%----------------------------------------------------------------------------
%Obtain the T-Template
%=====================
if height_1 > height_2
    t_height = height_1;
else
    t_height = height_2;
end


T_on_face = func_DrawLine(image_cropped,y0,x0,y1,x1,0);
T_on_face = func_DrawLine(T_on_face,y2,x2,y3,x3,0);
figure, imshow( T_on_face ), title( 'centroids joined' );

[a b c] = size(image_cropped);
black_background = imread( 'ttemplate.jpg' );
black_resized = imresize( black_background, [a b], 'nearest' );
t_skeleton = func_DrawLine(black_resized,y0,x0,y1,x1,255);
t_skeleton = func_DrawLine(t_skeleton,y2,x2,y3,x3,255);
figure, imshow( t_skeleton ), title( 'T skeleton' );
se = strel( 'rectangle', [t_height t_width] );
t_template = imdilate( t_skeleton, se );
figure, imshow( t_template ), title( 'T template' );
```

```
function [T_edge, T_intensity] = extract_t_template(t_template, image_cropped)

%==========================================================================
%           Obtain extracted T (edge detected & YIQ)
%==========================================================================

% Perform edge detection on face image
face_grayscale = rgb2gray( image_cropped );
face_median = medfilt2( face_grayscale, [2 2] );
face_clahe = adapthisteq( face_median );
edge_face = edge(face_clahe,'canny');
edge_face = im2uint16( edge_face );
figure, imshow( edge_face ), title( 'edge detected face' );

% Extract T-template of edge detected face
t_template = im2uint16( t_template );
t_template = imcomplement( t_template );
T_edge = imadd( edge_face, t_template );
figure, imshow( T_edge ), title( 'extracted t-template of edge detected face' );

% Obtain Intensity plane of face image
face_yiq = rgb2ntsc( image_cropped );
intensity_plane = face_yiq( :,:,1 );
figure, imshow( intensity_plane ), title( 'intensity plane of face image' );

% Extract T-template of face intensity plane
intensity_plane = im2uint16( intensity_plane );
T_intensity = imadd( intensity_plane, t_template );
figure, imshow( T_intensity ), title( 'extracted t-template of face intensity
plane' );
```

APPENDIX B : GANTT CHART

Month	1	2	3	4	5	6	7	8	9	10	11
Research on face detection											
Code for face detection											
Research for face recognition											
Code for face recognition											
Create GUI											
Modify and test											
Final report											

Figure 16 Project Gantt Chart