TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

PULCHOWK CAMPUS

A

MINOR PROJECT REPORT

ON

"I Know You (IKY)": Face Recognition System

By:

Ram Krishna Gubhaju (063/BCT/527)

Subodh Shakya (063/BCT/539)

Swastik Singh (063/BCT/546)

A PROJECT WAS SUBMITTED TO THE DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING IN PARTIAL FULLFILLMENT OF THE REQUIREMENT FOR THE BACHELOR'S DEGREE IN COMPUTER ENGINEERING

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING
LALITPUR, NEPAL

MARCH, 2010

COPYRIGHT

ACKNOWLEDGEMENT

We are very thankful to Institute of Engineering (IOE), Pulchowk Campus for offering the course on minor project. We also thank all teachers and staffs of Electronics and Computer Engineering Department who assisted during the project conduction period by giving suitable suggestions and lectures on different subject matters relating to the conduction and achievement of the project goals.

We are very grateful to our teachers Surendra Shrestha, Deepen Chapagain and Manoj Ghimire who helped a lot in the development of this project. We also express our gratitude to all the friends who helped a lot to learn about the development tools and programming languages.

Members of Project

Ram Krishna Gubhaju (063/BCT/527)

Subodh Shakya (063/BCT/539)

Swastik Singh (063/BCT/546)

ABSTRACT

We present a report on the Project titled “I Know You: face recognition system”. The application we developed is capable of detecting and recognizing any of human face that is acquired by the system using web camera or bitmap file. This system first detects the face from the input image by using detection algorithms or using manual methods. Then the system recognizes the person by comparing the characteristics of the face to those of known individuals. To the system the known individuals are those faces that are in its database. The developed system deals only with the 2-D faces represented in terms of bitmap object. Entire face detection and recognition procedure is based on Principle Component Analysis (PCA). Idea behind the use of PCA for recognition is to project each of the detected faces in face space that encodes the variation among known faces. The face space is defined by the “Eigen faces” that are eigenvectors calculated from set of faces. When a face is input to the system, it recognizes it by calculating the deviation of the previously projected faces in term of Euclidian distance. The known faces in the database with minimum Euclidian distance best matches with the input face and finally system recognizes the face.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

IKY: I Know You

PCA: Principle Component Analysis

1-D: 1-Dimension

2-D: 2-Dimension

SQL: Structured Query Language

ADO: Abstract Data Object

1 INTRODUCTION

The developed system titled “I Know You (IKY)” is a face recognition system having the capability to detect the face, add the face to the learning set and finally use those faces from the learning set to recognize the input face. Technique used for detection and recognition of the face in this application is Principle Component Analysis (PCA). The main idea behind the use of PCA for the face recognition is to construct a large 1-D vector of pixel from 2-D facial images. And then express those large 1-D vectors of pixel into the compact principle components of the feature space. This can be called eigenspace projection. Eigen space is calculated by identifying the Eigen vectors of the covariance matrix derived from the set of facial images (vectors). Now the face images is transformed into a small set of characteristic feature images called “Eigen faces”, which are the principle components of the initial training set of face images. Finally recognition is performed by projecting the new image into the faces space and then classifying the face by calculating its position in the face space and comparing its position with the position of known faces in the training set. The comparison of the new face and faces in the training set is done by comparing Euclidian distance. The face for which the Euclidian distance from new (unknown) face is minimum is the best match of the unknown face. Hence the new face is recognized.

2 LITERATURE REVIEW

To accomplish the goal of the project, the application finally build must have two of the capability; face detection and face recognition. Both of the mentioned features use PCA technique so we explain PCA followed by its use in face detection and recognition.

2.1 Principle Component Analysis (PCA)

2.1.1 Introduction to PCA

Principle Component Analysis (PCA) is fastest and most successful technique used in the image recognition and compression. Key concept of PCA is to reduce large dimensionality of the data space into smaller intrinsic dimensionality of feature space which are needed to describe data economically. This feature space defines the strong correlation between observed variable (in our case, face is observed variable). Different jobs done by PCA are prediction, redundancy removal, feature extraction and data compression. Redundancy removal is only feature we used in the development of face recognition system.

2.1.2 Mathematics of PCA

Mathematical operation required for the face recognition involves different steps. First step is to find the Eigen faces and second step is Face recognition based on calculated Eigen faces.

Steps involved in the calculation of the Eigen faces

i> Transformation of 2-D facial image to 1-D vector
ii> Calculation of the mean images
iii> Define mean centered images
iv> Calculating covariance matrix C
v> Simplification(Decrease the size of covariance matrix C)
vi> Finding orthonormal vectors e_i (Eigen vector)
vii> Projecting facial image into the face space

Detail explanation of these steps is as follows:

2.1.2.1 Transformation of 2-D facial image to 1-D vector

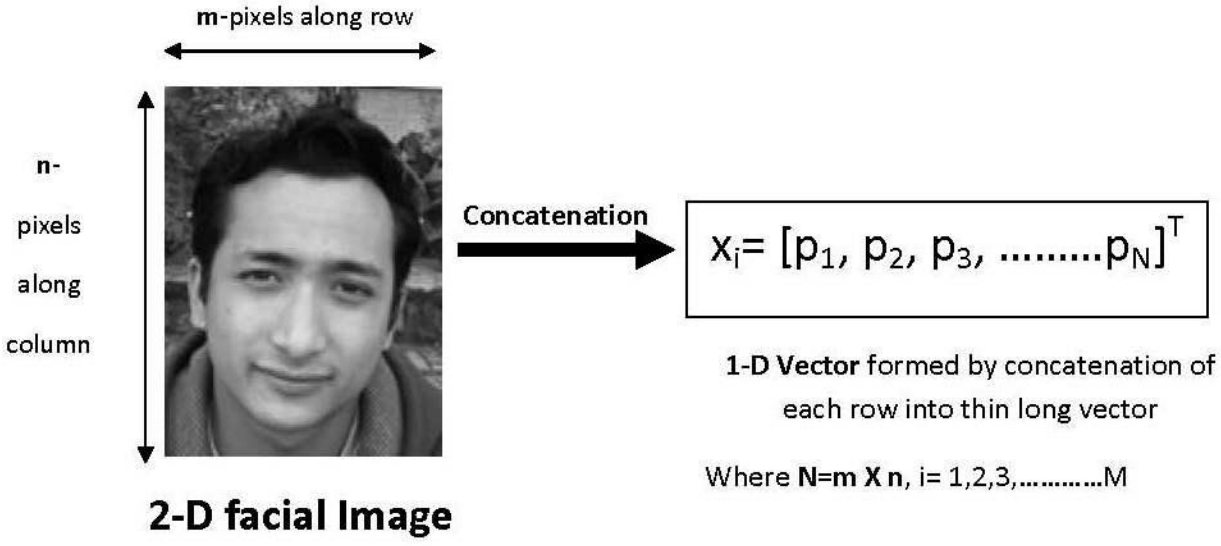

Figure 2.1 Transformation of 2-D facial image to 1-D vector

A 2-D facial image is represented by row X columns arrangement of pixels. This 2-D facial image is transformed into a long thin vector x by concatenation of each rows in 2-D into a vector having single row. Let each face be represented as vector of size N (=rows X columns of image). Let there be M such faces(vectors) defined by S. Then 1-D vector is given by

$$x_i = [p_1, p_2, p_3, \ldots\ldots\ldots p_N]^T$$

$$S = \{ x_1, x_2, \ldots\ldots\ldots\ldots\ldots , x_M \}$$

where p_j 's represent pixel values.

2.1.2.2 Calculation of Mean image

Let m be the mean image then this mean image for given M images is calculated as

$$m = \frac{1}{M}\sum_{i=1}^{M} xi$$

Figure 2.2 Mean face

2.1.2.3 Define mean centered image

Let Φi define the mean centered image then this mean centered image is calculated as

$$\Phi i = x_i - m$$

2.1.2.4 Calculating covariance matrix C

$$C = \frac{1}{m}\sum_{1}^{m} \phi_n \phi_n^T = AA^T$$

Where $A = [\phi_1 \phi_2 \ldots\ldots\ldots \phi_m]$

2.1.2.5 Simplification(Decrease the size of covariance matrix C)

Consider the matrix A^TA(M X M matrix). Here the eigenvector of matrix V_i of A^TA is computed.

$A^TAV_i=\mu_iV_i$

Where, μ_i is the eigenvalue of A^TA.

Or $A^TAV_i=\mu_iV_i$

Multiplying both sides by A we get

Or, $AA^TAV_i=A\mu_iV_i$

Or, $CAV_i= A\mu_iV_i$

Replacing AV_i by U_i we get

Or, $CU_i=\mu_iU_i$ $(AV_i=U_i)$

Thus, AA_T and $A^T A$ have the same eigenvalues and their eigenvectors are related as follows:

$AV_i=U_i$

Note:

- AA^T can have up to N^2 eigenvalues and eigenvectors.
- $A^T A$ can have up to M eigenvalues and eigenvectors.
- The M eigenvalues of $A^T A$ (along with their corresponding eigenvectors) correspond to the M *largest* eigenvalues of AA^T (along with their corresponding eigenvectors)

2.1.2.6 Finding orthonormal vectors e_i (Eigen vector)

Eigen vectors of Covariance matrix C is calculated usual mathematical procedure and is assigned to vector V. Top 5 eigenfaces is shown below.

Figure 2.3 Eigen Faces

2.1.2.7 Projecting facial image into face space

Projection of facial image into the face space is calculation of feature faces. A new face image (X) is transformed into its eigenface components (projected onto "face space") by a simple operation.

$W_k=U_k^T\Phi$ k=1,….,m.

The weights form a feature vector,

$\Omega^T=[W_1,W_2\ldots\ldots\ldots W_m]$

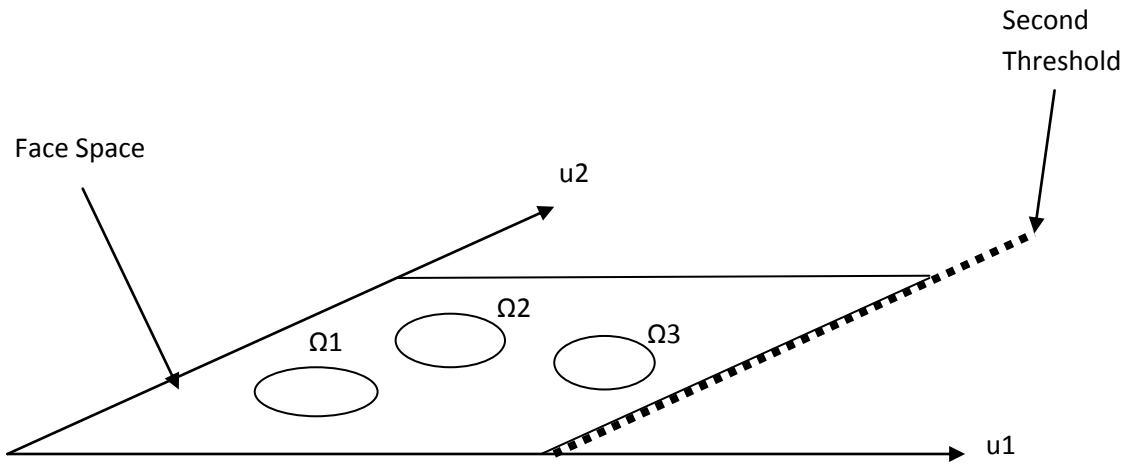

Figure 2.4 Projection of facial images into face space

2.2 Face Detection

Before application of any of the detection algorithms, the system must determine if the input image consist of face. If it does then only the system must apply any of the available face detection algorithms.

Filtering off the images not consisting of face is done by projecting the image into the feature space. If Euclidian distance for that image is greater than the second threshold then the input image is out of the face space and doesn't consist of face. So it must be rejected. However if the Euclidian distance is below threshold then system applies face detection algorithm based on skin tone. The value of second threshold is determined experimentally. We just require second threshold to reject the non facial image. In our Result and Analysis part we only calculate first threshold as we only require to reject faces beyond our face class.

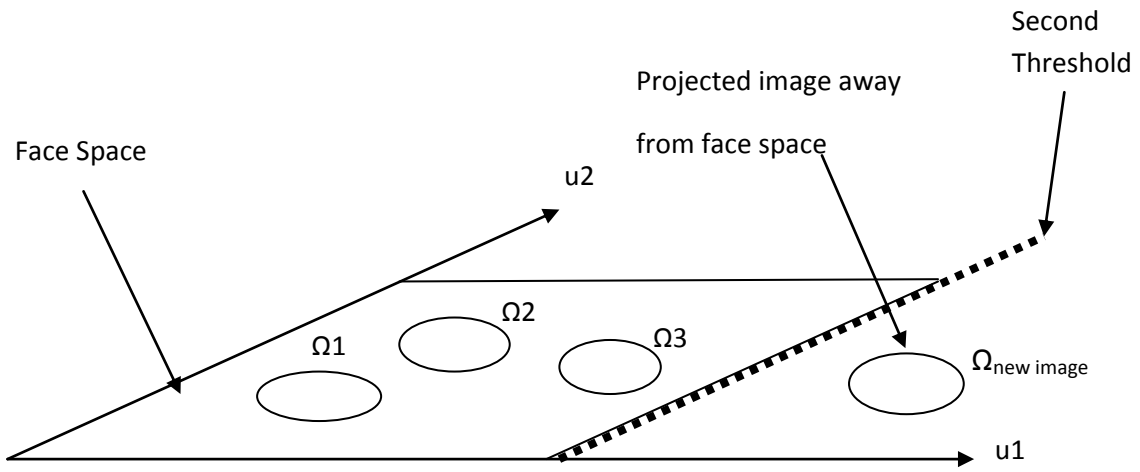

Figure 2.5 Face detection

In the figure 2.5 , $\Omega_{new\ image}$ is projected into the feature space. As it is out of the face space so it is not suitable for application of any of face detection algorithm.

2.3 Face Recognition

A new face is input to the system. For the recognition of this new face, it is projected into the face space. Then the Euclidian distance is calculated as

$$\epsilon = |\Omega - \Omega_k|$$

For each of the feature vector Ω in the training set, Euclidian distance is calculated for input face and Ω_i whose ϵ is minimum i.e. $\epsilon' = \epsilon_{min}$ is the best match for the input face and the corresponding Ω_i is the result. Hence the input face is recognized.

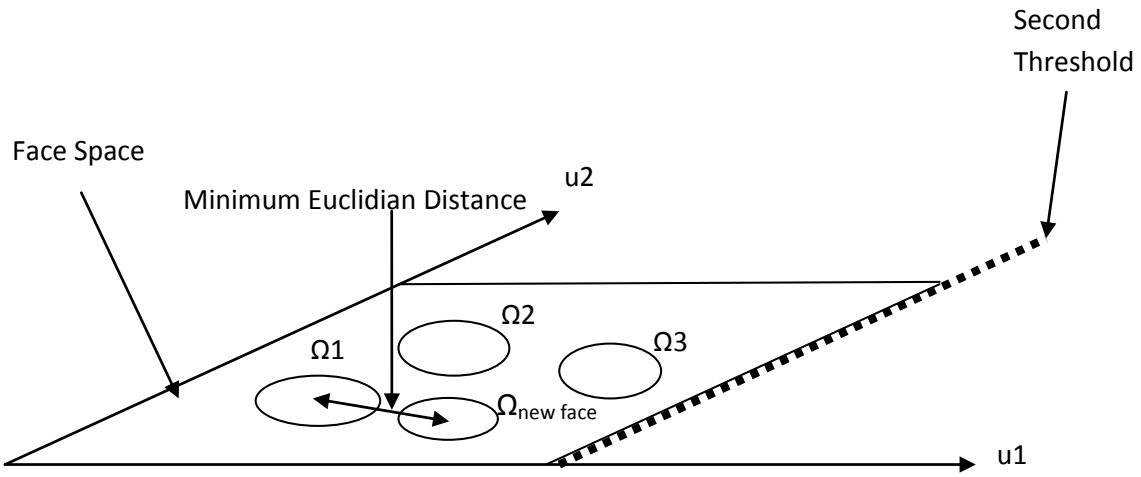

Figure 2.6 Face Recognition

In the figure 2.4, $\Omega_{new\ face}$ is projected into the face space. $\Omega_{new\ face}$ Euclidian distance from Ω_1 so Ω_1 is the best match and is displayed as the recognized face.

3. DEVELOPED SYSTEM

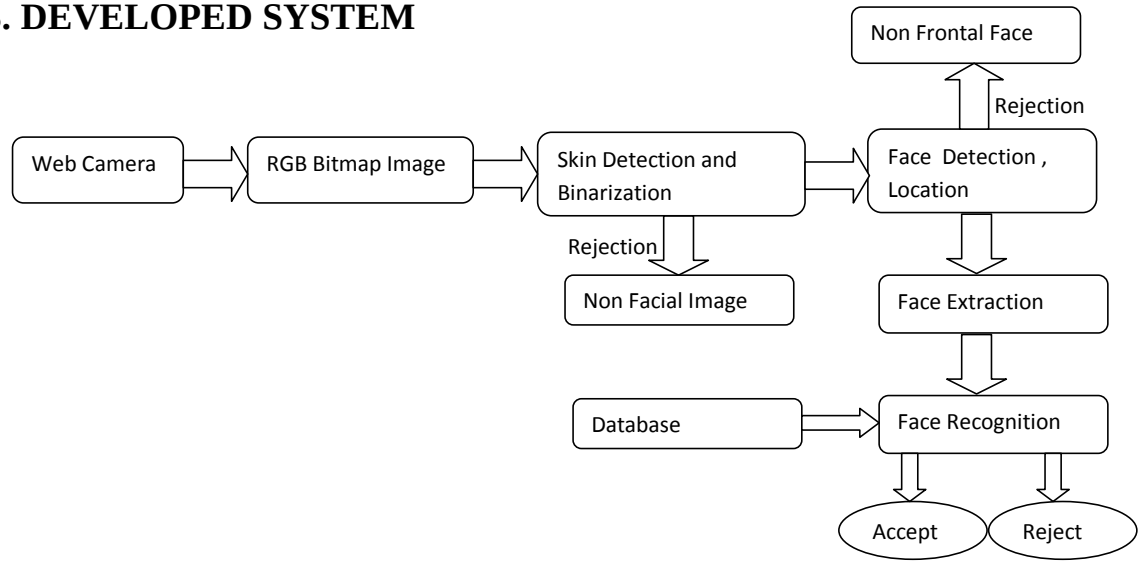

Figure 3.1: Developed system

The developed system has different features. Some of the Features are

1> Image Acquisition
2> Saving captured images
3> Cropping Images
4> Resizing and gray scaling cropped images
5> Placing data in the Server

3.1 Image Acquisition

Developed system can acquire the image of a person using two sources. First source is integrated or USB Web camera and second source is bitmap files.

Web camera – c# program interaction

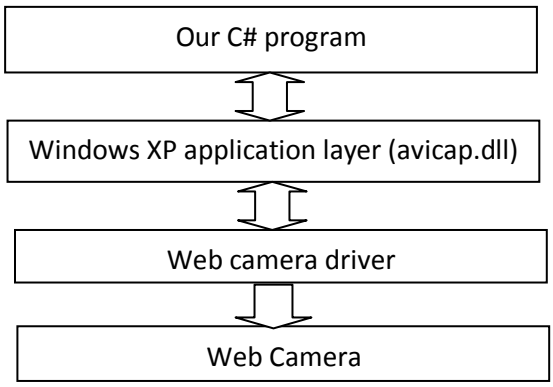

Figure 3.2: Webcam interaction with our program

Figure above shows the interaction between the IKY application developed in C# and web camera. IKY uses windows XP application layer called avicap.dll. To send command to this layer we used functions in the dynamic link library file called webcam_capture.dll. It consists of many functions but our application uses only three of it. Functions used are start, stop and capture. Start function turns on the camera. Stop turn off the camera. And capture stores on the frame from the streamed video.

Image can also be acquired using the files. To open a file and use it for processing, we are using file-open dialog box.

3.2 Saving captured image

Image captured by the web camera can be saved in two modes; i> Single file save mode and ii> Burst mode

i> Single file save mode: In this mode, an image is captured and can be saved to specified location.

ii> Burst mode: In burst mode, continuous frames from the web camera are saved to many files. Starting file number can be specified from a textbox.

3.3 Cropping Image

When application uses automatic or manual face detection algorithm, it locates the face and places a rectangle around it. For recognition procedure, only face from that image is required. So it is cropped and resized to required dimension. While cropping the face region, application uses the 4:3 height to width ratio. The cropped face is placed in the picture box for comparison with training set faces.

3.4 Resizing and gray scaling

Captured image is of the resolution 640 X 480 pixels. Decreasing this size to 288 X 216 decreases the time required for the completion of face detection algorithm as nested loop has to process only less number of pixels. Thus the captured image is resized to 288 X 216 and after the detection of the face, detected region is mapped in the 640 X 480 pixel image. This mapping is required as it maintain the quality of the cropped face region. If the cropping of the face from smaller size bitmap is done then there requires the interpolation of pixel to get required size so there will be loss of information.

To represent the two dimensional image into single dimension using PCA, only grayscale value of individual pixels are required so the cropped face is converted to grayscale image.

3.5 Placing data in Server

Schema to store the data related to the face is placed in SQL server. The server compatible with IKY application is Microsoft SQL server 2005 Express. As data can fit into a single table so schema diagram consists of single table and no relations are required.

Figure 3.2 IKYfacedata table

Primary key used in the table is faceID. Using faceID, IKY application can uniquely access a row and process photos in that row to compare with the input face to the application. Data access framework used by Visual C#.NET is ADO.NET (Abstract Data Object). ADO.NET consist of data access and data representation classes. Using classes like dbConnection, connection with the SQL server can be established. Then using dbCommand and dbReader classes, different queries can be executed to access any row of the table in the database.

4. IMPLEMENTED ALGORITHMS

4.1 Skin detection algorithm

Face detection algorithm used in the system is based on the skin tone. Different persons have different skin tone so we need to find the probable skin regions on the basis of colorspace. With variation in the lighting condition, algorithms based on different color space work with variable efficiency. We have used three skin detection algorithms. Detail explanation of the algorithm is below.

4.1.1 Skin detection algorithm in RGB color space

Face detection algorithm in RGB color space doesn't consider the effect due to the lighting condition. Segmentation of the given image into skin and non skin region is done of the basis of the heuristics. These heuristics are result of the experiments performed. The RGB region is classified a skin if

R>95, G>40,B>20, Max{R,G,B}-min{R,G,B}<25, |R-G|>25, R>G,R>B

These conditions are not arbitrarily assigned. They resulted from experiments performed on the images consisting of the skin region.

4.1.2 Skin detection algorithm in YCbCr color space

There are two ways of segmenting the image based on skin color: concerting the RGB picture to YCbCr space or to HSV space. A YCbCr space segments the image into a luminosity component and color components, whereas an HSV space divides the image into the three components of hue, saturation and color value. The main advantage of converting the image to the YCbCr domain is that influence of luminosity can be removed during our image processing. In the RGB domain, each component of the picture (red, green and blue) has a different brightness. However, in the YCbCr domain all information about the brightness is given by the Ycomponent, since the Cb (blue) and Cr (red) components are independent from the luminosity. The Cb and Cr components give a good indication on whether a pixel is part of the skin or not.

Color space transformation from RGB to YCbCr is carried by following mathematical operation:

$$Y=0.3R+0.6G+0.1B$$

Where Y is luminance.

Now let τ be defined as

$\tau = 1.4$ if $Y<64$
$\tau = 0.6$ if $Y<192$
$\tau = 1$ otherwise

We can determine the compensated image having chrominance(Cr) as:

$Cr=0.5R'- 0.419G'- 0.081B$

Where $R'=R^{\tau}$ and $G'=G^{\tau}$

Let S represent two segments namely skin and non skin. If 0 represents skin and 1 represents non skin then,

$S=0$ if $10<S<45$
$S=1$ otherwise

4.1.3 Skin detection algorithm in YUV - YIQ color space

Image in the RGB color space is converted into the YUV-YIQ color space by following matrix operation

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.2990 & 0.5870 & 0.1140 \\ -0.1471 & -0.2888 & 0.4359 \\ 0.6148 & -0.5148 & -0.1000 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

$$ch = \sqrt{|U|^2 + |V|^2}, \theta = \tan^{-1}(|V|/|U|) * (180/3.14)$$

θ = Hue, ch=Saturation and U and V holds chromatic information

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.2990 & 0.5870 & 0.1140 \\ 0.5957 & -0.2745 & 0.3213 \\ 0.2115 & -0.5226 & -0.3111 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Where I is the red-orange axis, and Q is roughly orthogonal to I.
The less I value means the less blue-green and the more yellow.

Any region is skin if the following conditions are satisfied

$20< I < 90$ and $30 < \theta < 75$ and $20 < ch < 220$

4.2 Face detection algorithm

After skin region is detected, face detection algorithm is carried as

i> Read the RGB values of the individual pixel.

ii> To remove the effect of the luminosity we are performing the color space transformation from RGB to YCbCr. Or us any of the other two color space.

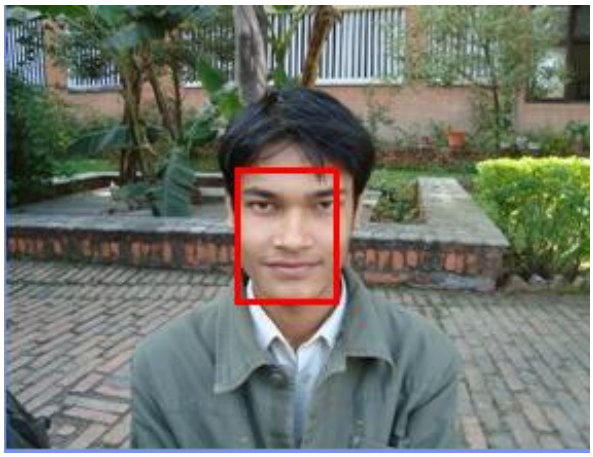

Figure 4.1 Face detection rectangle

iii> Using the experimental results we are differentiating skin and non skin region. We represent the skin by white and non skin by black.

Figure 4.2 Binarization

iv> Now the clustering of the skin region is the next step. In clustering of skin region, we are scanning the image from top-left to bottom-right. We are using 16x16 pixel blocks. If number of white region(skin) is greater than

50% then we paint the entire block with white representing skin else we paint it with black.

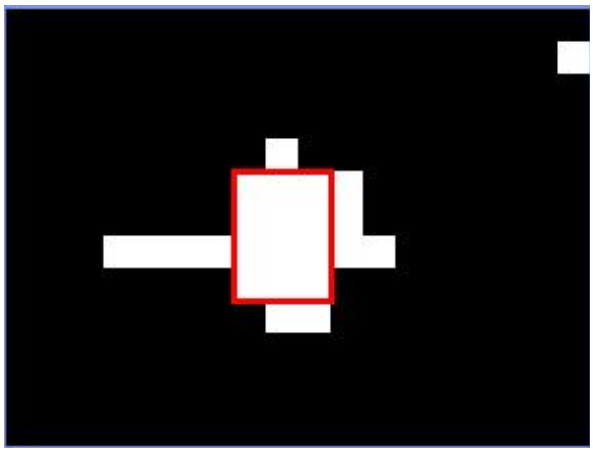

Figure 4.3 Scanning and location of face

v> Finally we scan from top-left to bottom-right with rectangles of different sizes. Generally human face's height to width ration is 1 to 1.6. So if this condition is fulfilled then the region must be face. One of the eg. Is as presented in the three images in steps ii,iii and iv. We thank our friend Pratik for his photo in the above example.

4.3 Face recognition algorithm

Face recognition is implemented by following the Principle Component Analysis as mentioned earlier.

5 APPLICATION OUTPUT

5.1 Main user interface

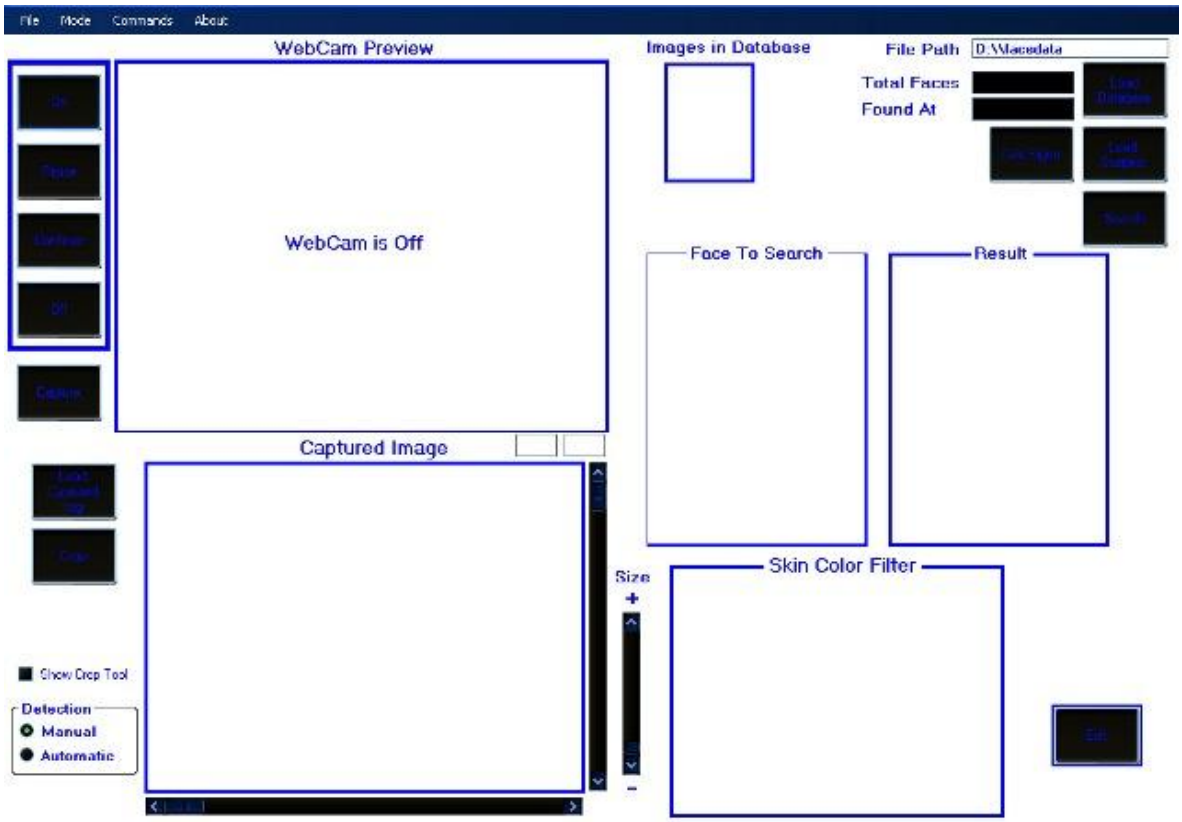

Figure 5.1 Main User Interface

Some of the basic controls of the user interfaces are

i> Web camera controls: It turns on, off the web camera and it also enables to capture image.

ii> Face detection controls: Face detection can be done in the captured image using this control either in automatic mode or manual mode. In automatic mode, in build algorithms are used while in manual mode, face is located manually.

iii> Face recognition controls: For face recognition, load database load the facial data in the server or files in the hard disk into the RAM. Calc Eigen button calculates Eigen faces for the training set faces. Load input face loads the face to be compared. Finally search the input face in the training set.

5.2 Update interface

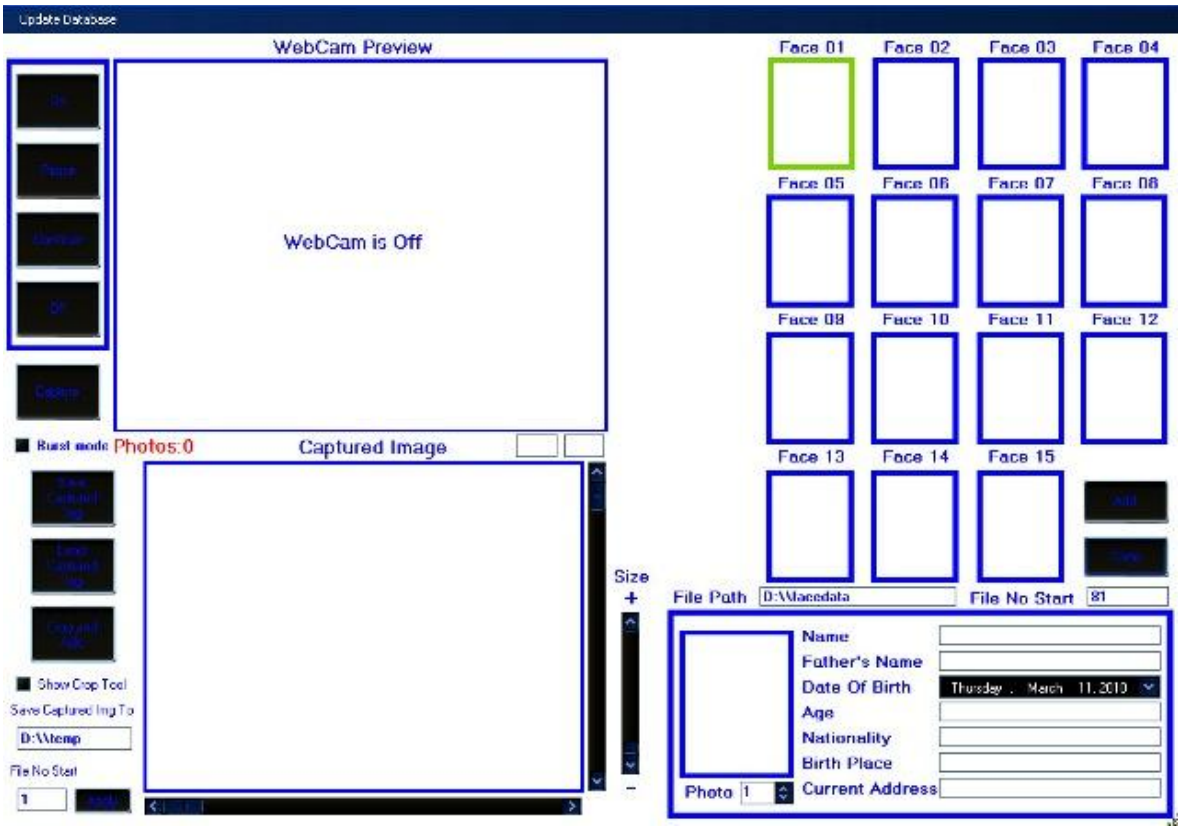

Figure 5.2 Update Interface

Some of the basic controls in update interface:

Web camera controls and face detection controls are same as in main user interface. The additional control is cropping tool control. This control help user to move the position of the are to be cropped and finally from the face and copy to the input faces collection that is finally added to the training set.

6 RESULTS AND ANALYSIS

6.1 Results

The result presented below is based on the input faces considered within same lighting conditions. However different result may occur if input faces have different lighting condition used than that used below.

To observe the result from the application developed we input 40 facial images of 8 different people. These faces are cropped from the image that is captured in same background and lighting conditions.

Figure 6.1 Faces in database of 8 different persons

Figure 6.1 shows the collection of faces in database under similar lightning conditions. These images in our database are each of resolution 101x76.

First we determined the threshold for the facial images contained in the database to determine the range of our face class. The threshold is determined in two cases. The first with 20 signatures for each image and second with 25 signatures for each image.

Initially we used 20 signatures for each image. And we took sample of 40 facial images (different from images stored in database) of individuals contained in the database for testing our software. Similarly we took sample 45 facial image of individual not contained in our database. And we got following results:-

Mean eucledian distance=258290.9

Standard Deviation=177693.3

Threshold chosen =mean eucledian distance +Standard deviation=435984.2

Recognition error I(mismatch with facial images contained in database)

= 25/40*100

= 25%

Recognition error II(match with facial images not contained in database)

=13/14*100

=29%

Recognition success =27/40*100 =67.5%

Unrecognizable(Threshold exceeded)=3/40*100= 7.5%

Finally, we used 25 signatures for each image. And we took sample of 40 facial images (different from images stored in database) of individuals contained in the database for testing our software. Similarly we took sample 45 facial image of individual not contained in our database. And we got following results:-

Mean eucledian distance= 363580.5

Standard Deviation= 202310.9

Threshold chosen =mean eucledian distance +Standard deviation= 565891.4

Recognition error I(mismatch with facial images contained in database)

= 10/40*100

= 25%

Recognition error II(match with facial images not contained in database)

= 10/45*100

=22.22%

Recognition success =25/40*100 =62.5%

Unrecognizable (Threshold exceeded) =5/40*100= 12.5%

6.2 Analysis

We actually performed analysis of our algorithm of face recognition under two conditions first with 20 signatures of each face image and second with 25 signatures of each face image. We observed that as the number of signatures of each image is increased then the range of our face class also increased which is visualized by the increased in threshold. And with the increase in signatures for each facial image we expected to increase the recognition success and decrease in Recognition error I, Recognition error II and unrecognizable conditions of images in database. But since our face image resolution is only 101*76, we cannot get significant improvement Recognition error I, Recognition error II and unrecognizable conditions of images in database with the increase of signatures.

Possible causes of inefficiency of recognition in our case:

1> First cause of error in recognition is non uniform back ground.
2> Variation of proportion of face region with in the input image. That is in an input image, size of face will be large and in the other it may be small.
3> Small facial size image.

We used small facial size image just to achieve fast computation.

7 CONCLUSIONS AND FUTURE WORK

7.1 Conclusion

Hence with the development of the IKY application within the planned time period and observing the efficiency of the program above 50%, we to some extent achieved our project goal. All the documentation of the project is done within the code. All function and purpose of the variables and procedures used in the code is defined as a comment by the side of each variables and procedure.

7.2 Future enhancement

So our future enhancements to our application are

i> Present face detection system is cannot detect face in real time video streaming. So the future work will be to make the system capable of real time face detection.

ii> System cannot recognize faces in noisy background. In future we planned to implement the background subtraction algorithm to get facial data only.

iii> We also planned to extend this application running only in single computer to run in different computers through network which makes it capable of sharing the data with different computers within a network.

iv> Implementing Client-Server model. Server acting as the facial data storage unit and clients requesting these data for processing.

REFERENCES

1. Lindsay I Smith, *"A tutorial on PCA"* , PP 1-26 ,2002
2. Turk, M., and Pentland, A., *"Eigenfaces for recognition"*, Journal of Cognitive Neuroscience, Vol. 3, pp. 71-86, 1991
3. Yuille, A. L., Cohen, D. S., and Hallinan, P. W., *"Feature extraction from faces using deformable templates"*, Proc. of CVPR, pp. 99-111,1989.
4. M. Kirby and l. Sirovich, *"Application of the karhunen-Loeve Procedure For The Characterization Of Human Faces"* ,IEEE Transactions on Pattern Analysis And Machine Recognition, Vol. 12, No. 1, Jan. 1990
5. M.-H. Yang, D. Kriegman, and N. Ahuja, *"Detecting Faces in Images: A Survey"*, *IEEE Trans. PAMI*, Vol. 24, No. 1, pp.34-58, Jan. 2002.
6. V. Vezhnevets, V. Sazonov, and A. Andreeva, *"A Survey on Pixel-based Skin Color Detection Techniques"*, *Proc. Graphicon 2003*, Moscow, Russia, September 2003.
7. E. Hjelmas, and B.K. Low, *"Face Detection: A Survey"*, *Computer Vision and Image Understanding*, Vol. 83, No. 3, pp.236-274, 2001.
8. M. R. Girgis, T. M. Mahmoud, and T. Abd-El-Hafeez *, "An approach to Image Extraction and Accurate Skin Detection Using From Web Pages"*, World Academy of Science, Engineering and Technology ,pp.367-375, 2007