A PROJECT REPORT

ON

"EAR BIOMETRICS BASED ON PRINCIPAL COMPONENT ANALYSIS"

SUBMITTED

BY

DEVENDER SINGH CHAUDHARY(B3223015)

OM KUMAR SINGH (B3223031)

PRASHOVE KAUSHIK(B3223036)

VIVEK KHAJURIA(B3223057)

UNDER THE GUIDANCE OF

PROF. J.B. JAWALE

FOR THE PARTIAL FULFILLMENT OF

B.E. (ELECTRONICS & TELECOMMUNICATION) 2010-2011

TO

DEPARTMENT OF ELECTRONICS AND TELECOMMUNICATION

ARMY INSTITUTE OF TECHNOLOGY

DIGHI HILLS, PUNE – 411015

2010-2011

UNDER

UNIVERSITY OF PUNE

CERTIFICATE

This is to certify that

Devender Singh Chaudhary(B3223015)

Om kumar singh (B3223031)

Prashove Kaushik(B3223036)

Vivek Khajuria(B3223057)

have successfully submitted the project report on

"EAR BIOMETRICS BASED ON PRINCIPAL COMPONENT ANALYSIS"

During the academic year 2010-2011 in the partial fulfilment towards completion of Bachelors degree Program in Electronics and Telecommunication Engineering under University of Pune.

Project Guide	HOD E&TC	Principal
Prof. J.B.Jawale	Dr. R.K. Prasad	Dr. V P Gosavi

ACKNOWLEDGEMENT

Throughout the course of this project, the project provided us with great opportunity to experiment and learn. A successful project is the result of teamwork, which put in their logic, skills, knowledge and also those who guide us.

With due respect we would like to thank our Project Guide, Prof. **J.B. Jawale** for his co-operation, untiring support and guidance throughout the year.

We would also like to take opportunity for expressing gratitude & sincere thanks towards our staff of the E&TC Department and to the Head of Department

Mr. R. K.Prasad for providing computer lab and other valuable support for making our project.

We are indebted to all of you.

ABSTRACT

With the increase in need for strong security systems, the biometric systems are becoming more and more popular. These systems are based on human traits which, unlike passwords or pins, cannot be lost, stolen or forgotten. One such trait is ear. With the initial doubts on uniqueness of ear, ear biometrics could not attract much attention. But after it has been said that it is almost impossible to find two ears with all the parts identical, ear biometrics has gained its pace. In this project a simple principal component analysis has been implemented to extract features and match the test image with the database.

INDEX

.

1. INTRODUCTION

Biometric systems play a significant role in almost all the security aspects. As it uses human traits for the identification purpose which cannot be stolen or lost, they are proving to be a better solution than pins and passwords. There are many human traits that can be used as a biometric like fingerprint, face, voice and iris. Voice being a behavioral characteristic can vary with changes in emotion or health. Automating identification through biometrics especially face and iris recognition have been extensively studied in machine vision. Despite extensive research many problems in these remain largely unsolved due to the inherent difficulty of feature extraction. A wide variety of imaging problems (e.g., lighting, shadows, scale, and translation) further plague the attempt in this direction. An alternative to this is ear biometrics. It has been seen that finding two ears which are completely identical is almost impossible and ear does not change much with time, unlike face. Moreover, ear satisfies all the properties that should be possessed by a biometric.

There exist some systems developed for ear recognition using 2D and 3D images. Burge and Burger proposed an approach based on voronoi diagrams. Hurley, Nixon and Carter have given a system based on force field feature extraction. The system proposed by Choras is based on geometric feature extraction. Ear recognition from 2D images is developed in Chen and Bhanu proposed an approach based on contour matching. Yan et al. have used ICP on 3D images.

In this project we have implemented ear biometrics by using principal component analysis.
PCA is a useful statistical technique that has found application in fields such as face recognition and image compression, and is a common technique for finding patterns in data of high dimension.

It is a way of identifying patterns in data, and expressing the data in such a way as to highlight their similarities and differences. Since patterns in data can be hard to find in data of high dimension, where the luxury of graphical representation is not available, PCA is a powerful tool for analysing data.

The main advantage of PCA is that once you have found these patterns in the data, and you compress the data, ie. by reducing the number of dimensions, without much loss of information. This technique used in image compression.

2. PROBLEM STATEMENT

The aim of the project is to implement Ear Biometrics Based on Principal Components Analysis.

For the same a database is maintained, comprising of images of Ears of different individuals. Various features of these images are derived using a geometrical approach and are stored in a separate database.

A Graphical User Interface (GUI) is to be built to load the test image and make it pass through feature extraction process, and these features are compared with the feature's database. If the result of comparison is within the threshold limit with any of the database image, then test image belongs to that individual, and record of that individual is to be displayed in a window.

Hence a user friendly system is to be developed for verifying the identity of the individual using the image of his ear.

3. LITERATURE SURVEY

PCA is a useful statistical technique that has found application in fields such as face recognition and image compression, and is a common technique for finding patterns in data of high dimension.

We will discuss the different concepts used in the PCA .

It covers:

Standard deviation

Covariance

Eigen vectors

Eigen values.

This background knowledge is meant to make the PCA section very straightforward.

Statistics

The entire subject of statistics is based around the idea that you have this big set of data and you want to analyse that set in terms of the relationships between the individual points in that data set. We are going to look at a few of the measures you can do on a set of data, and what they tell you about the data itself.

Standard Deviation

To understand standard deviation, we need a data set. Statisticians are usually concerned with taking a sample of a population. To use election polls as an example, the population is all the people in the country, whereas a sample is a subset of the population that the statisticians measure. The great thing about statistics is that by only measuring (in this case by doing a phone survey or similar) a sample of the population, you can work out what is most likely to be the measurement if you used the entire population.

The Standard Deviation (SD) of a data set is a measure of how spread out the data is.

The English definition of the SD is: "The average distance from the mean of the data set to a point". The way to calculate it is to compute the squares of the distance from each data point to the mean of the set, add them all up,
divide by n-1,where is the number of items in the data set, and take the positive square root.

$$s = \sqrt{\frac{\sum_{i=1}^{n}(X_i - \bar{X})^2}{(n-1)}}$$

The above is the formula used for standard deviation.

The Standard Deviation is a measure of how spread out numbers are.
Its symbol is **σ** (the greek letter sigma)
The formula is easy: it is the **square root** of the **Variance.**

Mean

The mean doesn't tell us a lot about the data except for a sort of middle point.

The mean is calculated using the formula given below.

$$\bar{X} = \frac{\sum_{i=1}^{n} X_i}{n}$$

Variance

Variance is another measure of the spread of data in a data set. In fact it is almost identical to the standard deviation. The formula is this:

$$s^2 = \frac{\sum_{i=1}^{n}(X_i - \bar{X})^2}{(n-1)}$$

$$var(X) = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(X_i - \bar{X})}{(n-1)}$$

You will notice that this is simply the standard deviation squared, in both the symbol (s^2) and the formula (there is no square root in the formula for variance). s^2 is the usual symbol for variance of a sample. Both these measurements are measures of the spread of the data. Standard deviation is the most common measure, but variance is also used.

Variance is another measure of the spread of data in a data set. In fact it is almost identical to the standard deviation.

Work out the Mean (the simple average of the numbers)

Then for each number: subtract the Mean and then square the result (the squared difference).

Then work out the average of those squared differences.

Covariance

The last two measures we have looked at are purely 1-dimensional. Data sets like this could be: heights of all the people in the room, marks for the last exam etc. However many data sets have more than one dimension, and the aim of the statistical analysis of these data sets is usually to see if there is any relationship between the dimensions. For example, we might have as our data set both the height of all the students in a class, and the mark they received for that paper. We could then perform statistical analysis to see if the height of a student has any effect on their mark.

Standard deviation and variance only operate on 1 dimension, so that you could only calculate the standard deviation for each dimension of the data set independently of the other dimensions. However, it is useful to have a similar measure to find out how much the dimensions vary from the mean with respect to each other.

Covariance is such a measure. Covariance is always measured between 2 dimensions.

If you calculate the covariance between one dimension and itself, you get the variance.

The formula for covariance between two variables X and Y is given by:

$$cov(X,Y) = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{(n-1)}$$

“For each data item, multiply the difference between the value and the mean of X , by the the difference between the Y value and the mean of Y. Add all these up, and divide by n-1” .

The exact value of Covariance is not as important as it’s sign (ie. Positive or negative). If the value is positive, then that indicates that both dimensions increase together.

If the value is negative, then as one dimension increases, the other decreases. If we had ended up with a negative covariance here, then that would have said the opposite.

In the last case, if the covariance is zero, it indicates that the two dimensions are independent of each other.

Since the covariance value can be calculated between any 2 dimensions in a data set, this technique is often used to find relationships between dimensions in high-dimensional data sets where visualisation is difficult.

You might ask “is con(X,Y) equal to con(Y,X)”? Well, a quick look at the formula for covariance tells us that yes, they are exactly the same since the only difference between con(X,Y) and con(X,Y) is that $(X_i - \bar{X})(Y_i - \bar{Y})$ is replaced by $(Y_i - \bar{Y})(X_i - \bar{X})$. And since multiplication is commutative, which means that it doesn’t matter which way around I multiply two numbers, I always get the same number, these two equations give the same answer.

The covariance between two real-valued random variables *X* and *Y* with finite second moments is

$$\operatorname{Cov}(X,Y) = \operatorname{E}\big[(X - \operatorname{E}[X])(Y - \operatorname{E}[Y])\big],$$

where E[*X*] is the expected value of *X*. By using some properties of expectations, this can be simplified to

$$\operatorname{Cov}(X,Y) = \operatorname{E}\big[XY\big] - \operatorname{E}[X] \cdot \operatorname{E}[Y].$$

For random vectors *X* and *Y* (of dimensions $m \times 1$ and $n \times 1$ respectively) the $m \times n$ covariance matrix is equal to

$$\operatorname{Cov}(X,Y) = \operatorname{E}\big[(X - \operatorname{E}[X])(Y - \operatorname{E}[Y])'\big] = \operatorname{E}\big[XY'\big] - \operatorname{E}[X]\operatorname{E}[Y]',$$

where *M* ′ is the transpose of *M*.

The (i,j)-th element of this matrix is equal to the covariance $\mathrm{Cov}(X_i, Y_j)$ between the i-th scalar component of X and the j-th scalar component of Y. In particular, $\mathrm{Cov}(Y, X)$ is the transpose of $\mathrm{Cov}(X, Y)$.

Random variables whose covariance is zero are called uncorrelated.

The units of measurement of the covariance $\mathrm{Cov}(X, Y)$ are those of X times those of Y. By contrast, correlation, which depends on the covariance, is a dimensionless measure of linear dependence.

The covariance Matrix

Recall that covariance is always measured between 2 dimensions. If we have a data set with more than 2 dimensions, there is more than one covariance measurement that can be calculated.

For example, from a 3 dimensional data set (dimensions x, y, z) you could calculate $cov(x, y)$, $(cov(x, z)$, and $cov(y, z)$. In fact, for an n dimensional data set, you can calculate $\frac{n!}{(n-2)!*2}$ different covariance values.

A useful way to get all the possible covariance values between all the different dimensions is to calculate them all and put them in a matrix.

So, the definition for the covariance matrix for a set of data with n dimensions is:

$$C^{n\times n} = (c_{i,j},\ c_{i,j} = cov(Dim_i, Dim_j))$$

where $C^{n \times n}$ is a matrix with n rows and n columns, and Dim_x is the xth dimension. All that this ugly looking formula says is that if you have an n dimensional data set, then the matrix has n rows and columns (so is square) and each entry in the matrix is the result of calculating the covariance between two separate dimensions.

We'll make up the covariance matrix for an imaginary 3 dimensional data set, using the usual dimensions x, y and z. Then, the covariance matrix has 3 rows and 3 columns, and the values are this:

$$C = \begin{pmatrix} cov(x,x) & cov(x,y) & cov(x,z) \\ cov(y,x) & cov(y,y) & cov(y,z) \\ cov(z,x) & cov(z,y) & cov(z,z) \end{pmatrix}$$

Some points to note: Down the main diagonal, you see that the covariance value is between one of the dimensions and itself. These are the variances for that dimension. The other point is that since $cov(a,b) = cov(b,a)$ the matrix is symmetrical about the main diagonal.

In probability theory and statistics, a **covariance matrix** is a matrix whose element in the *i*, *j* position is the covariance between the *i* th and *j* th elements of a random vector (that is, of a vector of random variables). Each element of the vector is a scalar random variable, either with a finite number of observed empirical values or with a finite or infinite number of potential values specified by a theoretical joint probability distribution of all the random variables.

Matrix Algebra

This section serves to provide a background for the matrix algebra required in PCA. Specifically We will be looking at eigen vectors and eigen values of a given matrix.

Eigen vectors and Eigen values :

As you know, you can multiply two matrices together, provided they are compatible sizes. Eigenvectors are a special case of this.

The **eigenvectors** of a square matrix are the non-zero vectors that, after being multiplied by the matrix, remain proportional to the original vector (i.e., change only in magnitude, not in direction). For each eigenvector, the corresponding **eigenvalue** is the factor by which the eigenvector changes when multiplied by the matrix. The prefix **eigen-** is adopted from the German word "eigen" for "own" in the sense of a characteristical description. The eigenvectors are sometimes also called **proper vectors**, or **characteristic vectors**. Similarly, the eigen values are also known as **proper values**, or **characteristic values**.

Eigenvectors can only be found for square matrices. And, not every square matrix has eigenvectors.

And, given an n*n matrix that does have eigenvectors, there are n of them. Another property of eigenvectors is that even if I scale the vector by some amount before I multiply it, I still get the same multiple of it as a result.

This is because if you scale a vector by some amount, all you are doing is making it longer, not changing it's direction. Lastly, all the eigenvectors of a matrix are perpendicular, ie. at right angles to each other, no matter how many dimensions you have. By the way, another word for perpendicular, in maths talk, is orthogonal. This is important because it means that you can express the data in terms of these perpendicular eigenvectors, instead of expressing them in terms of the x and y axes.

Another important thing to know is that when mathematicians find eigenvectors, they like to find the eigenvectors whose length is exactly one. This is because, as you know, the length of a vector doesn't affect whether it's an eigenvector or not, whereas the direction does. So, in order to keep eigenvectors standard, whenever we find an eigenvector we usually scale it to make it have a length of 1, so that all eigenvectors have the same length.

Eigenvalues are closely related to eigenvectors.
Eigenvectors and Eigen values always come in pairs.

4. BLOCK DIAGRAM

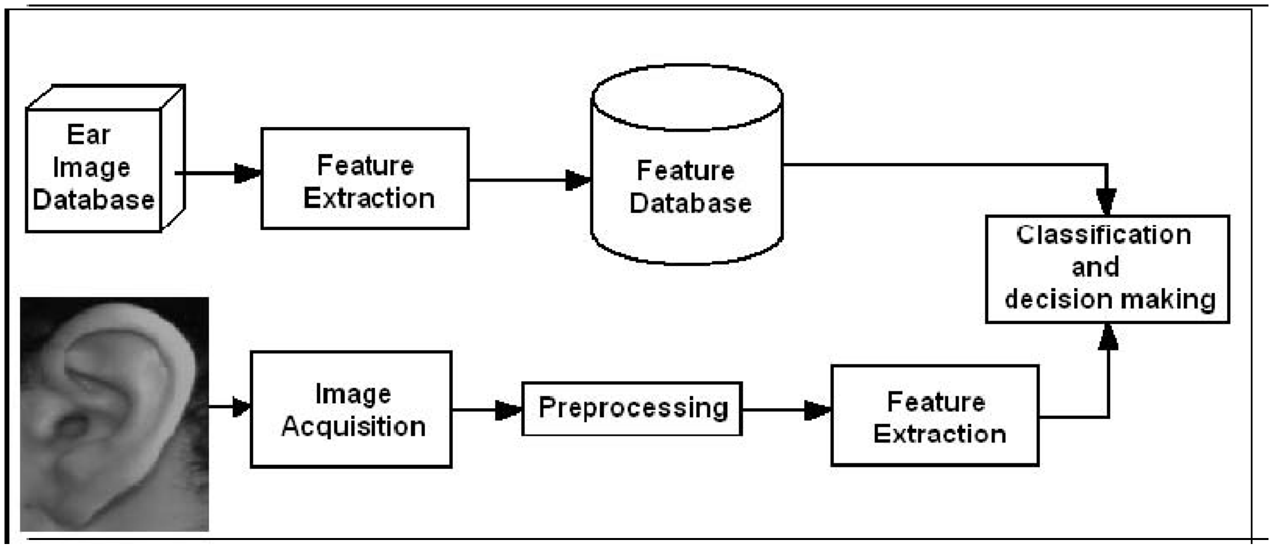


Figure no: 1

Like the other biometric recognition systems ear recognition also can be divided into five main parts –

Image acquisition – For acquiring the image

Preprocessing – For edge detection and image filtering

Feature extraction – For finding out unique features of each ear

Feature Database – For storing the extracted features in a database

Classification & Decision Making – For comparing the unique features and displaying the result

5. IMPLEMENTATION

IMAGE ACQUISITION

The side face images are acquired using Digital camera under same lightening conditions with no illumination changes (use of flash gives a fairly constant illumination). All the images are taken from the right side of the face with a distance of approximately 15-20 cm between the face and the camera. The images should be carefully taken such that the outer ear shape is preserved. The less erroneous the outer shape is, the more accurate the results are.

The images have been stored in JPEG format.

JPEG

The name "JPEG" stands for Joint Photographic Experts Group, the name of the committee that created the standard. The group was organized in 1986, issuing a standard in 1992, which was approved in 1994 as **ISO 10918-1**.

The JPEG standard specifies both the codec, which defines how an image is compressed into a stream of bytes and decompressed back into an image, and the file format used to contain that stream.

The encoding process consists of several steps:

1. The representation of the colors in the image is converted from RGB to YCbCr, consisting of component Y (luma), representing brightness, and two components, Cb and Cr (chroma), representing color. This step is sometimes skipped.

2. The resolution of the chroma data is reduced, usually by a factor of 2. This reflects the fact that the eye is less sensitive to fine color details than to fine brightness details.
3. The image is split into blocks of 8×8 pixels, and for each block, each of the Y, Cb, and Cr data undergoes a discrete cosine transform (DCT).
4. The amplitudes of the frequency components are quantized. Human vision is much more sensitive to small variations in color or brightness over large areas than to the strength of high-frequency brightness variations. Therefore, the magnitudes of the high-frequency components are stored with a lower accuracy than the low-frequency components. The quality setting of the encoder affects to what extent the resolution of each frequency component is reduced. If an excessively low quality setting is used, the high-frequency components are discarded altogether.
5. The resulting data for all 8×8 blocks is further compressed with a loss-less algorithm, a variant of Huffman encoding.

   The decoding process reverses these steps.

IMAGE PREPROCESSING

CROPPING

The 320*240 image which is taken from the 2M pixel camera is first opened in the MS Paint and is cropped .this cropped image is made so that the extra features capture by the camera are not included in the image which will be taken for further processing.

When you crop an image using MS Paint, the source file image is changed on the disk. This permanently alters your original image. You might want to create a copy of the image before changing it.

A cropped image is just like any other image .You can resize an image by clicking on the image and then dragging one of the handles (the square black boxes that appear around the edges and one corner of the image). The side handles make the image narrower or wider. The bottom handle drags the image taller or shorter. The corner handles change both the height and width of the image at the same time. You can maintain the height-to-width ratio by holding the shift key down on your keyboard while dragging the corner handle. This will force the image to keep the same proportion as the handle is being dragged.

GRAYSCALE CONVERSION

Conversion of a color image to grayscale is not unique; different weighting of the color channels effectively represent the effect of shooting black-and-white film with different-colored photographic filters on the cameras. A common strategy is to match the luminance of the grayscale image to the luminance of the color image.

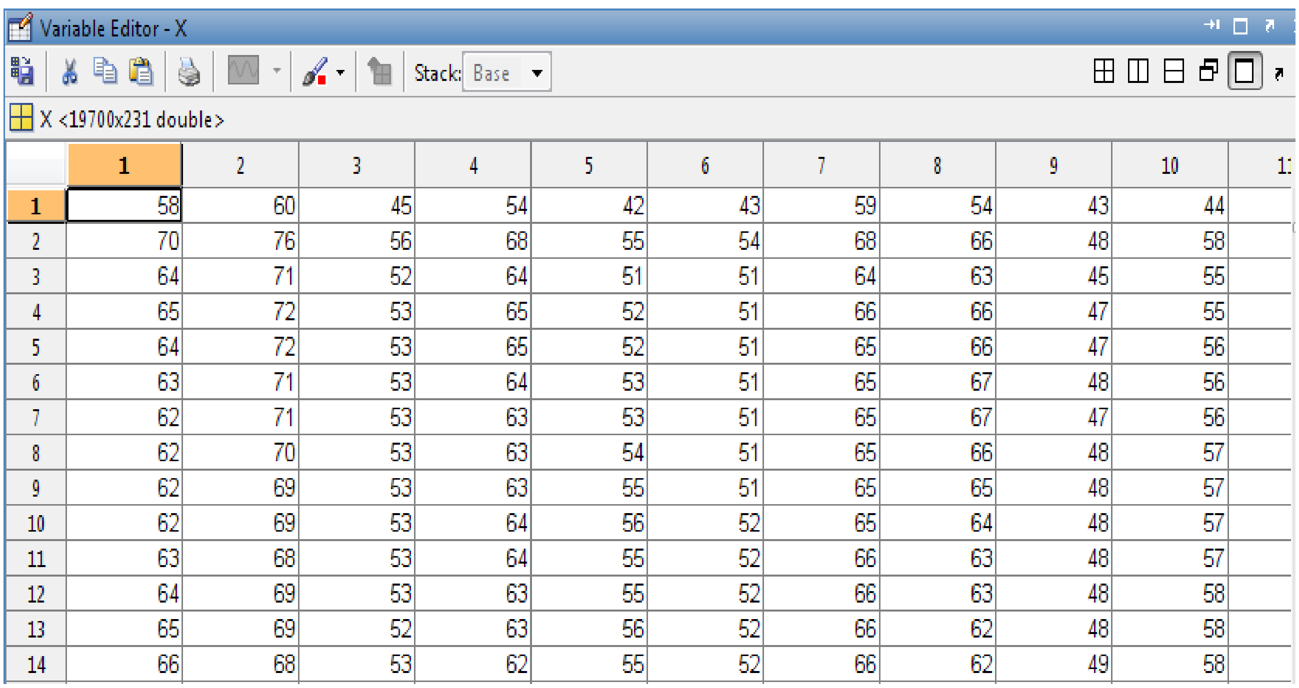
To convert any color to a grayscale representation of its luminance, first one must obtain the values of its red, green, and blue (RGB) primaries in linear intensity encoding, by gamma expansion. Then, add together 30% of the red value, 59% of the green value, and 11% of the blue value (these weights depend on the exact choice of the RGB primaries, but are typical). Regardless of the scale employed (0.0 to 1.0, 0 to 255, 0% to 100%, etc.), the resultant number is the desired linear luminance value; it typically needs to be gamma compressed to get back to a conventional grayscale representation.

To convert a color image which has been taken by the 2M pixel camera into a grayscale image the MATLAB function used is as given below

rgb2gray(RGB)

I = rgb2gray(RGB) converts the true color image RGB to the grayscale intensity image I. rgb2gray converts RGB images to grayscale by eliminating the hue and saturation information while retaining the luminance.

Features of the stored images:


Variable Editor - X
Stack: Base
X <19700x231 double>

	1	2	3	4	5	6	7	8	9	10
1	58	60	45	54	42	43	59	54	43	44
2	70	76	56	68	55	54	68	66	48	58
3	64	71	52	64	51	51	64	63	45	55
4	65	72	53	65	52	51	66	66	47	55
5	64	72	53	65	52	51	65	66	47	56
6	63	71	53	64	53	51	65	67	48	56
7	62	71	53	63	53	51	65	67	47	56
8	62	70	53	63	54	51	65	66	48	57
9	62	69	53	63	55	51	65	65	48	57
10	62	69	53	64	56	52	65	64	48	57
11	63	68	53	64	55	52	66	63	48	57
12	64	69	53	63	55	52	66	63	48	58
13	65	69	52	63	56	52	66	62	48	58
14	66	68	53	62	55	52	66	62	49	58

Figure no: 2

Calculation of the distances:

1.6962

1.7216

1.7217

1.7720

1.7874

1.7128

1.7935

1.8115

1.7170

1.8595

1.7648

1.8637

2.0566

2.0771

2.0738

1.9217

1.8853

1.9109

1.6934

By use of these we can set the value of a threshold.

The principal components analysis involves the following steps:

Step 1: Get some data

This is obtained by loading the test image

Step 2: Calculating the mean

we can calculate the mean of the sample.

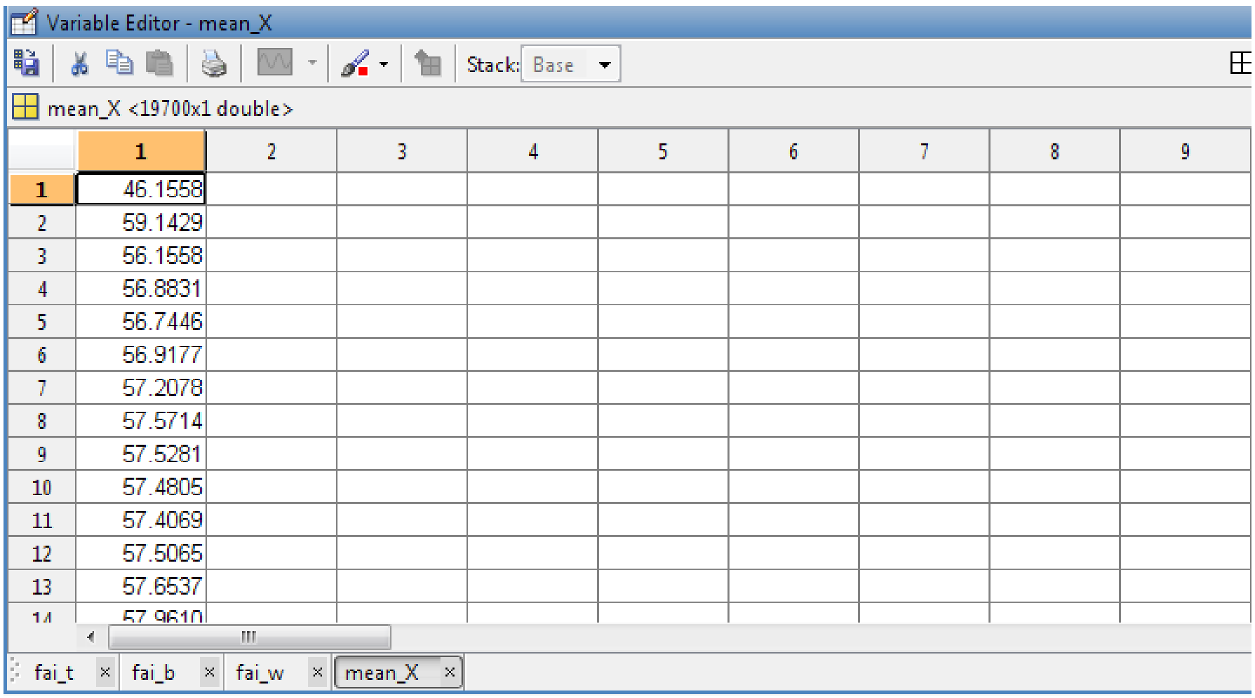


Figure no: 3

Step 3: Subtract the mean

For PCA to work properly, you have to subtract the mean from each of the data dimensions.

The mean subtracted is the average across each dimension. So, all the x values have the mean of the x values of all the data points subtracted, and all the y values have the mean of the y values of all the data points subtracted from them. This produces a data set whose mean is zero.

Step 4: Calculate the covariance matrix

Step 5: Calculate the eigenvectors and eigenvalues of the covariance Matrix

Step 6: Choosing components and forming a feature vector.

6. DATABASE

Our database contains of three parts:

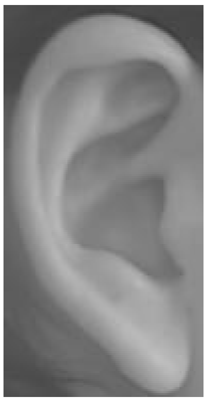
1. Ear Image Database – It contains the ear images of different people used in the context of ear recognition project carried out. Images of around 73 people are taken and stored in a folder with their names as filenames. This is named as the test database.

2. Ear Database – This database contains atleast 3 images for each person from the above database.we search for the match of our test image in this database.

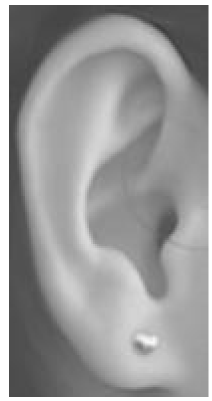
3. Face Database – It contains the faces of different persons who's ear are under test .This is used to display the face of the person when a match has been found.

4. Feature database –The calculated features for the image are stored in the matlab database.

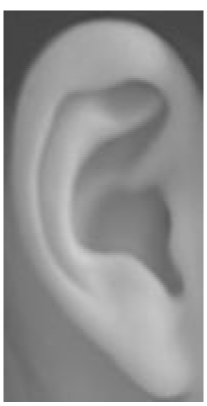
The figure below shows the ear image of different persons



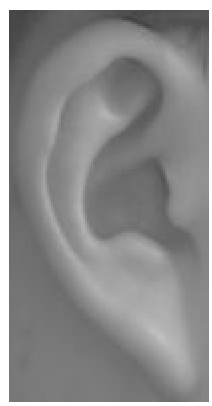
1.



2.



3.



4.

Figure no: 4

CLASSIFICATION & DECISION MAKING

In this case the extracted features are stored in database.

Then the features of the test image are extracted.

Then minimum distance of these features is calculated from that used in database.

The image which shows closest result will be the true image.

Variable Editor - X

Stack: Base

X <19700x231 double>

	1	2	3	4	5	6	7	8	9	10
1	58	60	45	54	42	43	59	54	43	44
2	70	76	56	68	55	54	68	66	48	58
3	64	71	52	64	51	51	64	63	45	55
4	65	72	53	65	52	51	66	66	47	55
5	64	72	53	65	52	51	65	66	47	56
6	63	71	53	64	53	51	65	67	48	56
7	62	71	53	63	53	51	65	67	47	56
8	62	70	53	63	54	51	65	66	48	57
9	62	69	53	63	55	51	65	65	48	57
10	62	69	53	64	56	52	65	64	48	57
11	63	68	53	64	55	52	66	63	48	57
12	64	69	53	63	55	52	66	63	48	58
13	65	69	52	63	56	52	66	62	48	58
14	66	68	53	62	55	52	66	62	49	58

The above figure shows the features for the images.

Figure no: 5

7. GRAPHICAL USER INTERFACE

For creating a GUI we have used "GUIDE", the MATLAB graphical user interface development environment that provides a set of tools for creating graphical user interfaces (GUIs). These tools simplify the process of laying out and programming GUIs.

We have made three GUI's namely-

1. A GUI to see the different type of the ears.

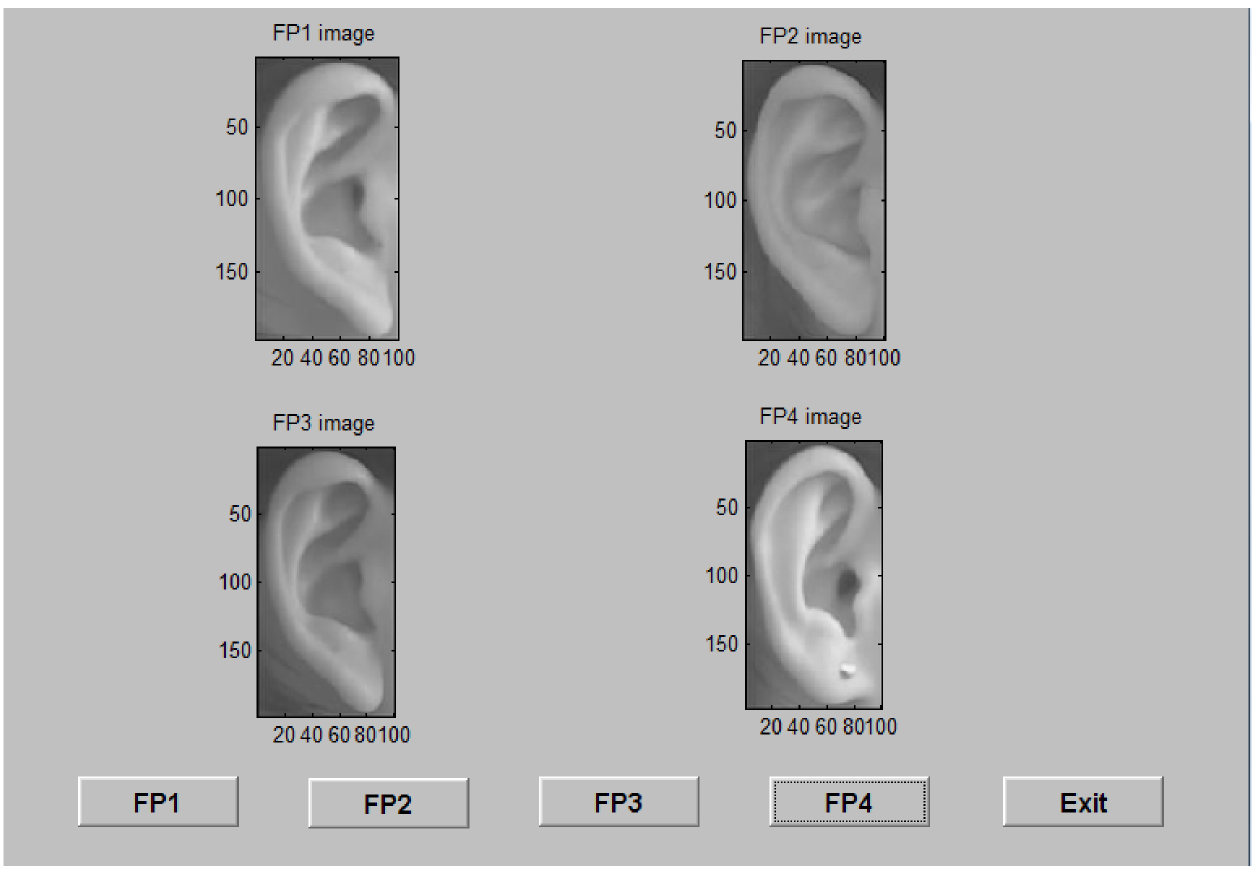


Figure no: 6

2. It allows the user to load the image of the ear, and then it gives him option.

a) MATCH- For extracting the feature vectors of the loaded image and to compare them with all the feature vectors in the database

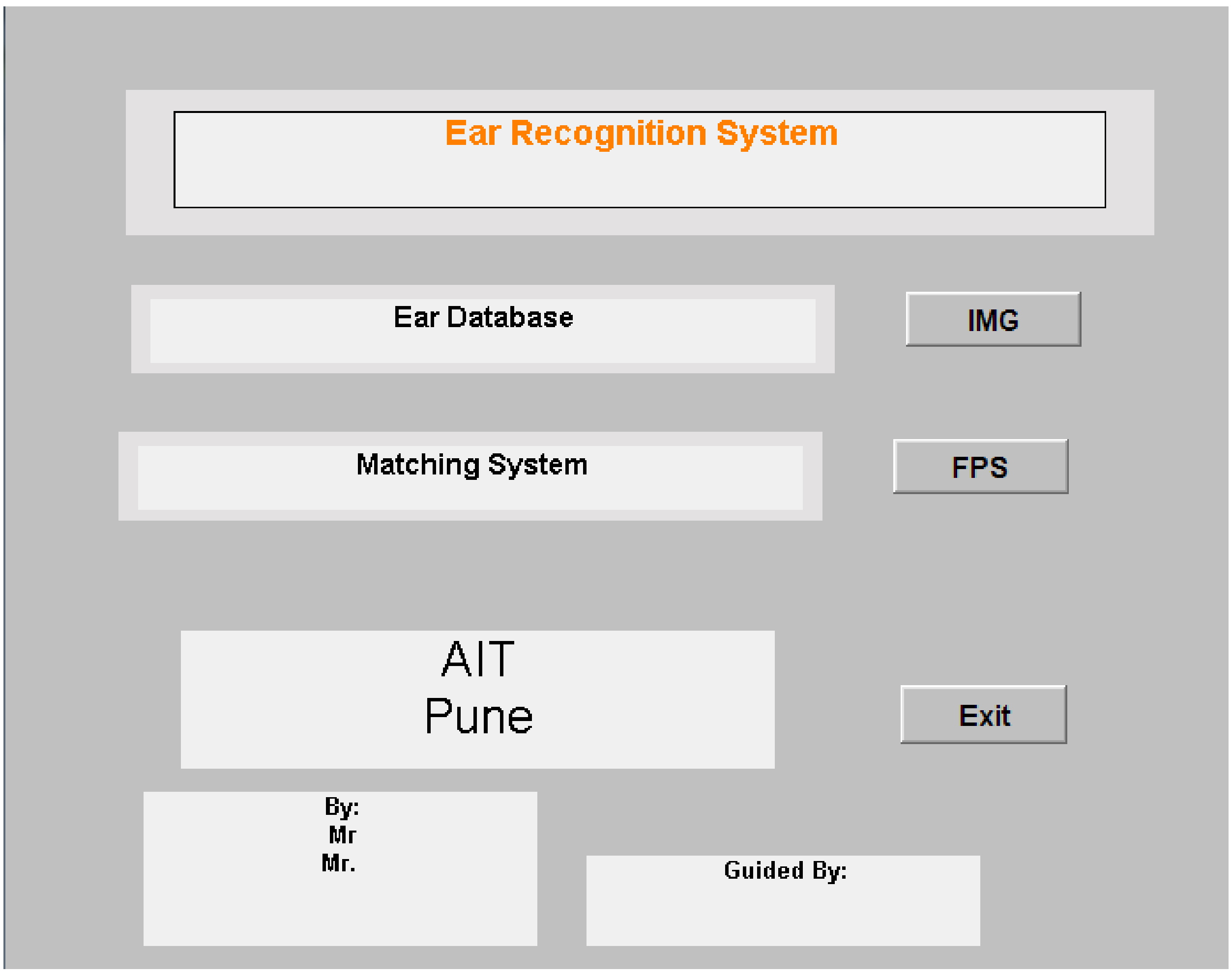


Figure no: 7

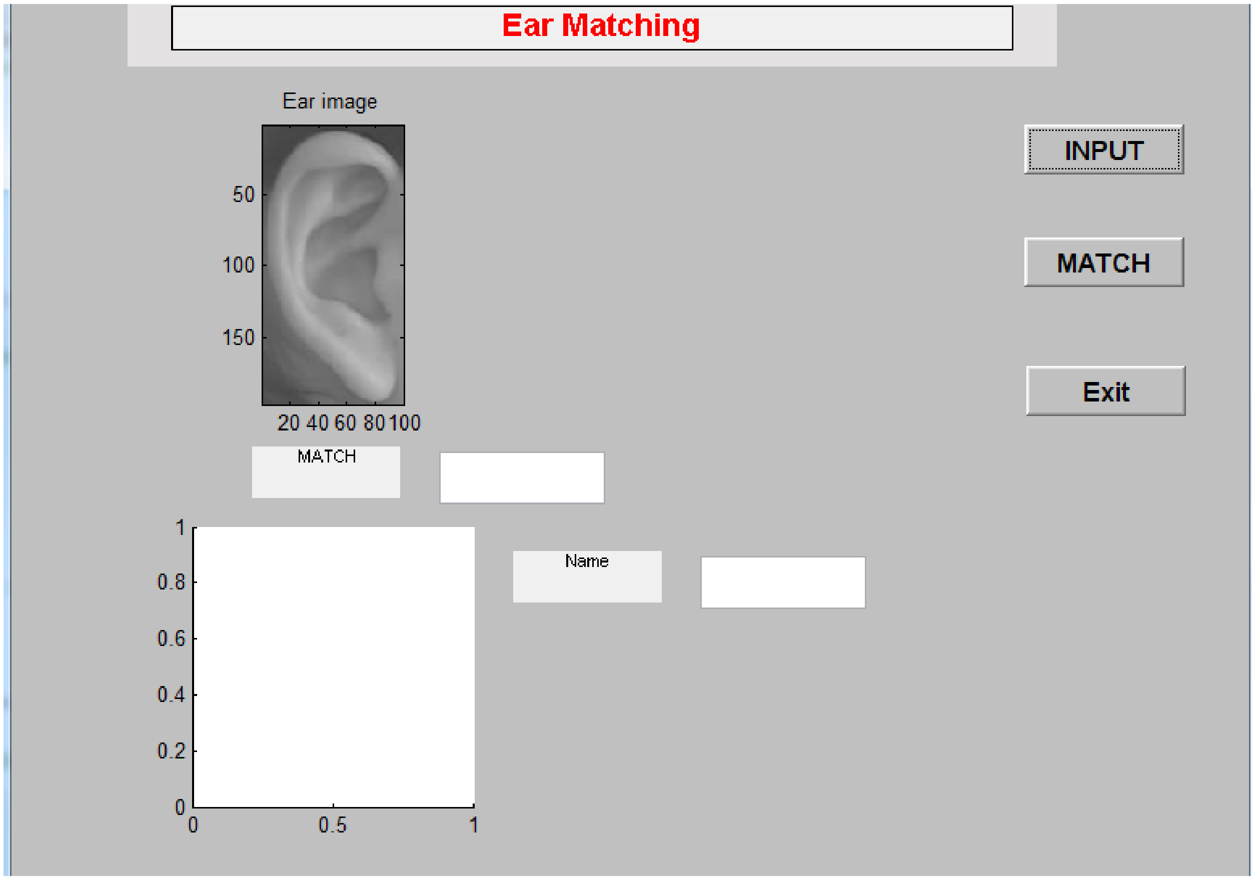


Figure no: 8

3. This GUI is for displaying the image and record of the person whose feature vectors have matched with the feature vectors of the ear image under test.

So when the user loads an image and clicks onto MATCH option in the GUI "p1" and if the software finds a match for the ear image, then the record of the user, with whom the feature vectors of the test image have matched, is displayed using GUI "p2".

If a match is found then following figures are displayed :-

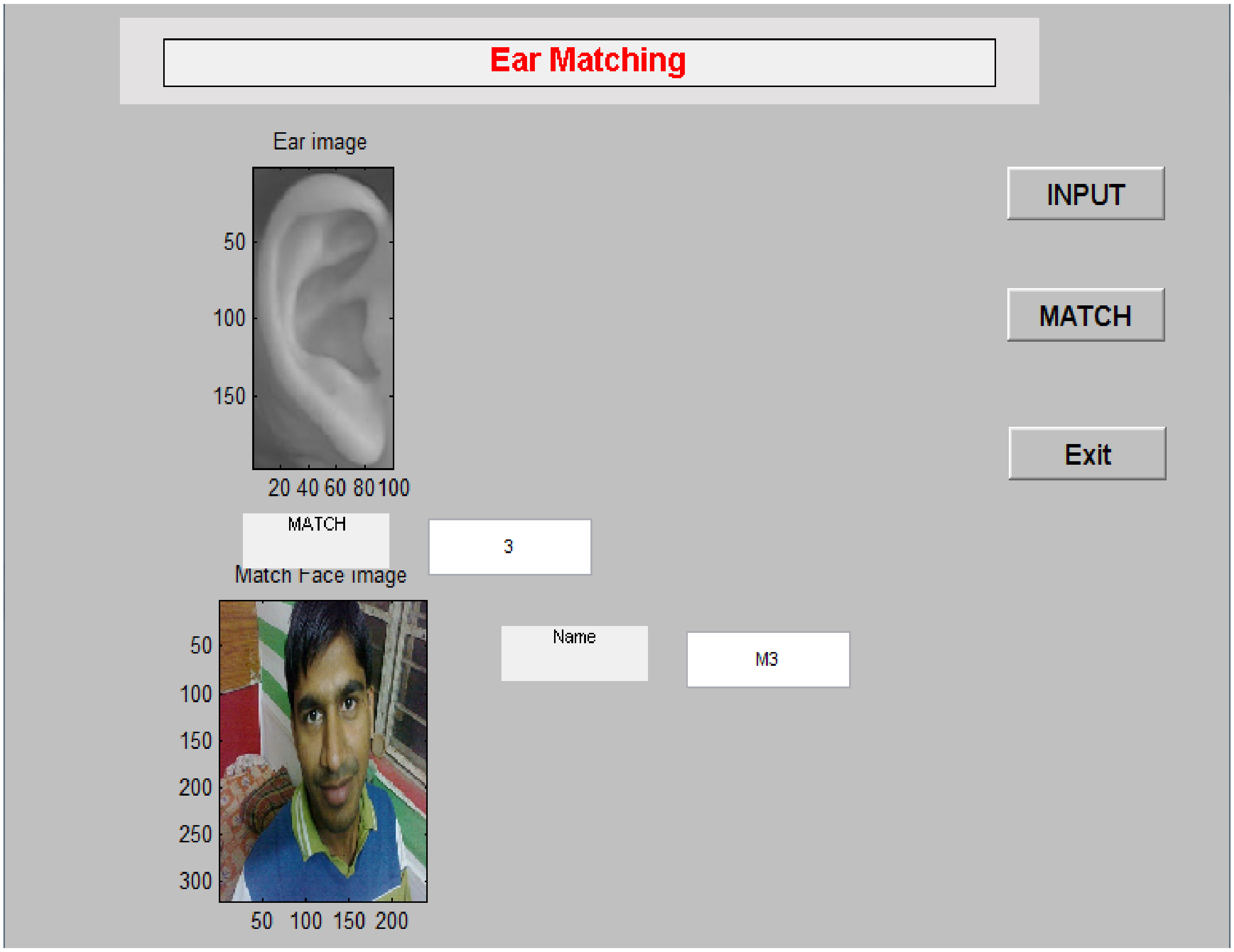

Ear Matching
Ear image
50
100
150
20 40 60 80 100
INPUT
MATCH
Exit
MATCH
3
Match Face image
50
100
150
200
250
300
50 100 150 200
Name
M3

Figure no: 9

8. RESULTS

- Below is a sample of the output from our main code.
- We are using Principal Components Analysis.
- We are using standard database for performing test
- We have 3 images for each person.
- The GUI shows the required information about the persons
- PCA is a well defined method in image biometrics.
- Overall efficiency of 92 % for our technique of obtaining 2 feature vectors from the same contour.
- Computational complexity for our project is very less.
- And time for processing is reduced as only one contour is used making identification faster.

9. ADVANTAGES & APPLICATIONS

ADVANTAGES

- Ear recognition uses physiological characteristics which can neither be manipulated (i.e. forged) neither forgotten (as in the case of pin or password).
- Ear does not change during human life, and face changes more significantly with age than any other part of human body.
- Cosmetics, facial hair, hair styling and emotions expressing different states of mind like sadness, happiness, fear or surprise don't interfere with the identification process.
- Colour distribution is more uniform in ear than in human face, iris or retina.
- Not much information is lost while working with the greyscale or binarized images.
- Ear is also smaller than face, which means that it is possible to work faster and more efficiently with the images with the lower resolution.
- Ear images cannot be disturbed by glasses, beard or make-up.
- Little or no maintenance is required which would be otherwise necessary in case of finger print recognition system.

APPLICATIONS

- This system can be used for security purposes to monitor and limit the entry and exit of authorized personal.
- Used by forensics for criminal identification.

FUTURE SCOPE

- The future scope of our project is development of a more robust system which can identify a person with more efficiency, by enhancing the information contained in the feature vectors.

- Another scope is development of a system which can be used in high security systems where in person can be identified from a distance without his/her knowing.

- Even a system can be developed where in user can take image from his cell, connect with the internet and identify a person from a large worldwide person database.

10. CONCLUSION

A user friendly system for the identification of a person using his ear image has been developed based on a simple two stage geometric approach for ear recognition using MATLAB.

The implemented algorithm depends mainly on the outer shape of the ear and the accuracy in finding the max line. Since the images are not ideal there may be an error in the outer shape of the ear which may result in the failure of the whole approach.

An alternative is to test the query image with different max-lines in different angles. In conclusion we have used ear biometrics for passive identification of a person.

11. APPENDIX

LIST OF FIGURES

REFERENCES

[1] M. Burge, and W. Burger, Ear biometrics for Computer vision, In 23rd Workshop Austrian Association for Pattern Recognition, 2000

[2] Michał Chora´s and Ryszard S. Chora´s, Geometrical Algorithms of Ear Contour Shape Representation and Feature Extraction, Proceedings of the Sixth International Conference on Intelligent Systems Design and Applications (ISDA'06)

[3] Dasari Shailaja and Phalguni Gupta, A Simple Geometric Approach for Ear Recognition, 9th International Conference on Information Technology (ICIT'06)

[4] A tutorial on Principal Components Analysis -Lindsay I Smith

[5] R. C. Gonzalez, R.E. Woods and S. L. Eddins, Digital Image Processing Using MATLAB

[6] J.B. Jawale, S.D. Jadhav, Digital Image Processing

[7] www.wikipedia.com

[8] www.engineerstoday.com