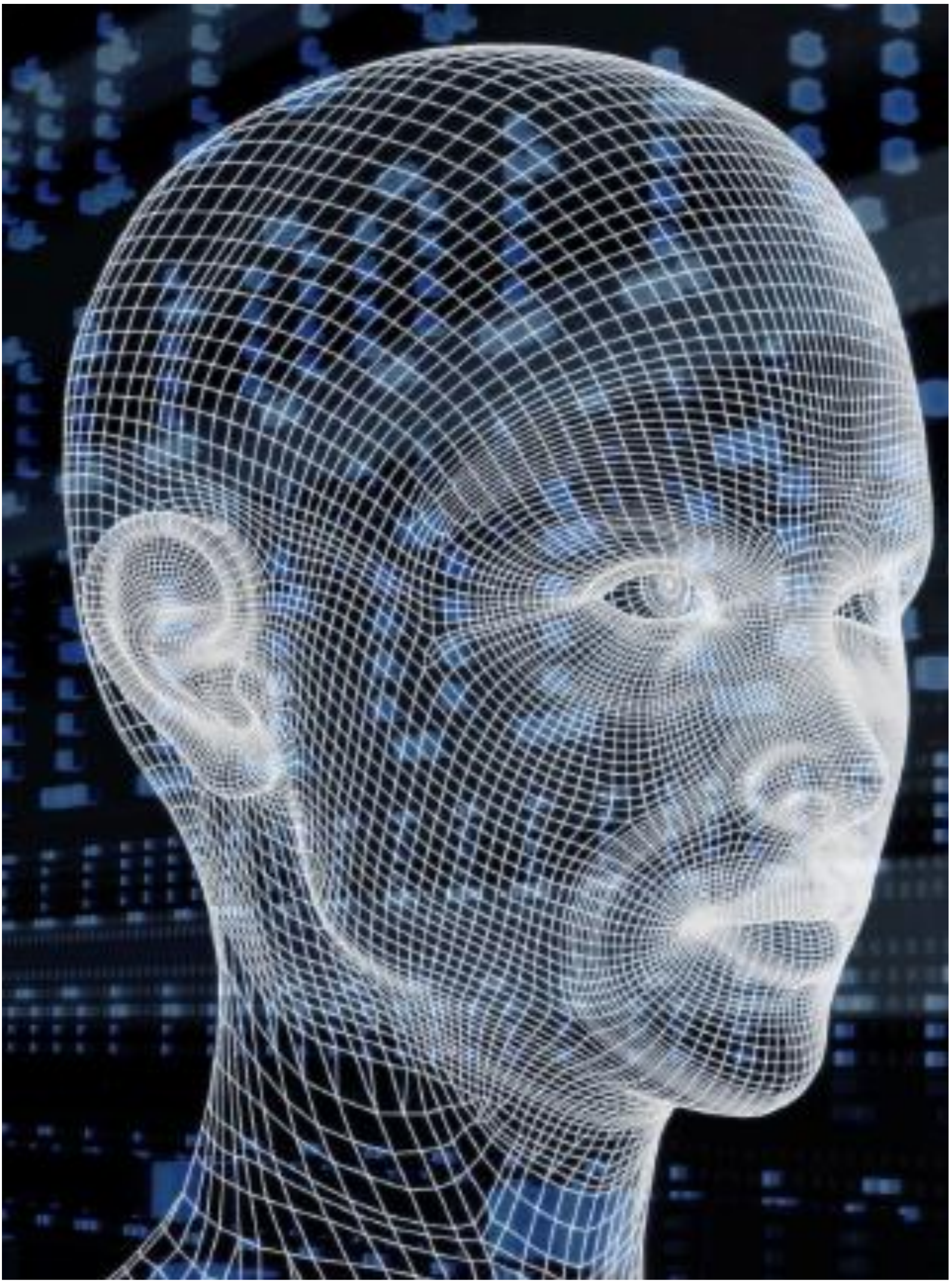
2015

Face Recognition System Using Machine Learning



GITAM University

3/30/2015

Face Recognition System Using Machine Learning

A Project Report submitted in partial fulfillment of the requirements
for the award of the degree of

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE AND ENGINEERING

Project Guide

T.Srikanth
Associate Professor

Project Members

Matta Sree Keerthi	-	1210311231
N.Anirudh	-	1210311203
Rama Manohar	-	1210311244
R.B.Srinivas	-	1210311245
G.Sai Shanmukh	-	1210311250

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
GITAM INSTITUTE OF TECHNOLOGY
GITAM UNIVERSITY
VIAKHAPATNAM-530045
ANDHRA PRADESH
INDIA

2014-2015

GITAM University

Department of Computer Science and Engineering

CERTIFICATE

This is to certify that the project entitled, ***"Face Recognition System using machine learning"*** *submitted by Matta Sree Keerthi (Roll: 1210311231) N.Anirudh (Roll:1210311203), I.Manohar (Roll:1210311244), G.S.S.Srinivas (Roll:1210311250) and R.B.Srinivas(Roll:1210311245) in fulfillment of requirements for the award of* ***Bachelors in Technology*** *degree in Department of Computer Science and Engineering at GITAM University, Visakhapatnam is a bona fide work carried out by them under my supervision and guidance.*

Prof Nageswara Rao	**Prof.T.Srikanth**
Head of the Department	Associate Professor

DECLARATION

We hereby declare that the project entitled "**Face Recognition System using machine learning**" is a bonafide work done in department of Computer Science and Engineering, GITAM Institute of Technology, GITAM University, Visakhapatnam submitted in partial fulfillment for the award of the degree of Batchelor of Technology in Computer Science and Engineering. This project is not submitted to any other University or Institution for the award of the Degree.

BY

NAME	REGD NO	SIGNATURE
Matta Sree Keerthi	1210311231	
N Anirudh	1210311203	
Rama Manohar	1210311244	
G.S.S.Srinivas	1210311250	
R.B Srinivas	1210311245	

ACKNOWLEDGEMENT

We would like to express our gratitude and greatest appreciation towards **Prof.T.Srikanth** for giving us an opportunity to work under him for the project. Without his encouragement and guidance this project would not have been materialized and completed. His contributions were vital for the completion of the project.

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

We would like to thank **Dr.T.Srinivasa Rao**(Associate Professor) and **Prof.K.Srinivasa Rao**(Associate Professor)for their valuable guidance at various stages of the project.

We would also like to thank all the academic and non-academic staffs for their generous help in various ways for the completion of this project. Last but not least, my sincere thanks to all my friends who have patiently extended all sorts of help for accomplishing this undertaking.

Project Team :

Matta Sree Keerthi
N.Anirudh
I.Manohar
R.B.Srinivas
G.Shanmukh

CONTENTS

Abstract

The importance of utilizing biometrics is to establish personal authenticity and to detect impostors is the present scenario of global security concern. Development of a biometric system for personal identification, which fulfils the requirements for access control of secured areas and other applications like identity validation for social welfare, crime detection, ATM access, computer security, etc., is felt to be the need of the day. Face recognition has been evolving as a convenient biometric mode for human authentication for more than last two decades. It plays an important role in applications such as video surveillance, human computer interface, and face image database management. A lot of techniques have been applied for different applications. Robustness and reliability becomes more and more important for these applications especially in security systems.

Basically Face Recognition is the process through which a person is identified by his facial image. With the help of this technique it is possible to use the facial image of a person to authenticate him into any secure system. Face recognition approaches for still images can be broadly categorized into holistic methods and feature based methods. Holistic methods use the entire raw face image as an input, whereas feature based methods extract local facial features and use their geometric and appearance properties.

This work studies the different approaches for a Face Recognition System. The different approaches like PCA and DCT have been studied with the help of Euclidean distance as a classifier.

The results have been computed with respect to a standard database i.e. ORL (Olivetti Research Lab) database which contains 400 images (each with 112×92 pixels) corresponding to 40 persons in 10 poses each including both male and female. The
ORL database image has been resized to 128×128 pixels.

Chapter 1

Introduction

1.1 Background

Biometrics is a class of Pattern Recognition problem. Biometrics is automated method of identifying a person or verifying the identity of a person based on a physiological or behavioral characteristic. Examples of physiological characteristics include hand or finger images, facial characteristics. Biometric authentication requires comparing a registered or enrolled biometric sample (biometric template or identifier) against a newly captured biometric sample (for example, captured image during a login).

During enrolment, as shown in the figure below, a sample of the biometric trait is captured, processed by a computer, and stored for later comparison. Biometric recognition can be used in Identification mode, where the biometric system identifies a person from the entire enrolled population by searching a database for a match based solely on the biometric. Sometime identification is called "one-to-many" matching.

A system can also be used in Verification mode, where the biometric system authenticates a person’s claimed identity from their previously enrolled pattern. This is also called “one-to-one” matching. In most computer access or network access environments, verification mode would be used.

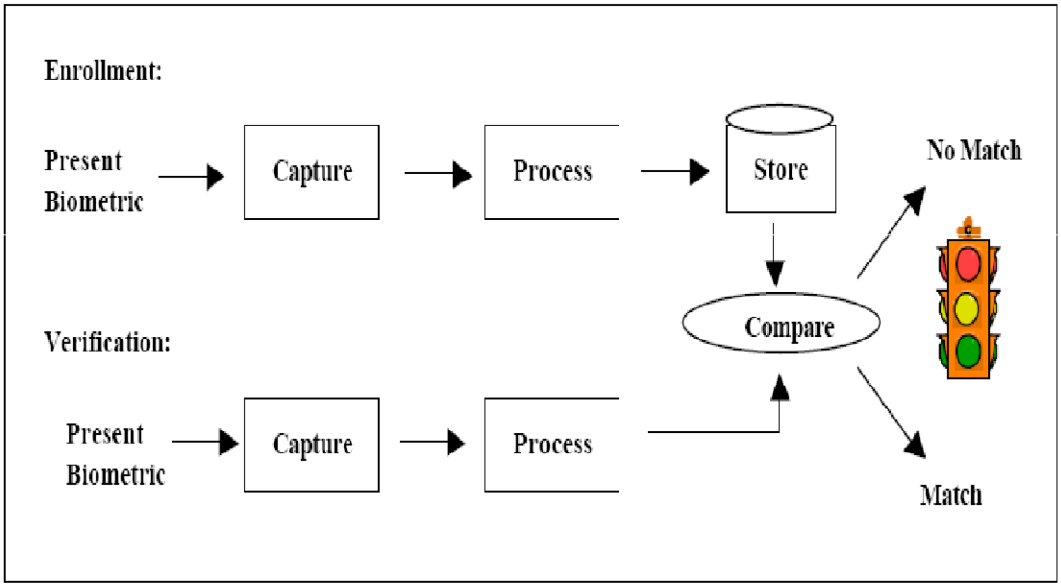


Fig 1.1: A model of Face Recognition System

Face Recognition

The identification of a person by their facial images can be done in a number of different ways such as by capturing an image of the face in the visible spectrum using an inexpensive camera or by using the infrared patterns of facial heat emission. Facial Recognition in visible light typically model key features from the central portion of the facial image using a wide assortment of cameras in visible light system extract features from the captured images that do not change over time while avoiding superficial features such as facial expression or hair. Several approaches to model facial images in the visible spectrum are Principal Component Analysis (PCA), local feature analysis, Neural Network, multi-resolution analysis etc.

The challenges of facial recognition in the visible spectrum include reducing the impact of variable lightning and detecting a mask or photograph. Some facial recognition systems may require a stationary or posed user in order to capture image through many systems, though many systems use a real time process to detect a person's head and locate the face automatically. Major benefits of facial recognition are that it is non intrusive, hand free, continuous and accepted by most users.

Most research on face recognition falls into two main categories (Chellappa et al., 1995): feature-based and holistic. Geometric approaches dominated in the 1980's where simple measurements such as the distance between the eyes and shapes of lines connecting facial features were used to recognize faces, while holistic methods became very popular in the 1990's with the well known approach of Eigen-faces .

Feature-based approaches to face recognition basically rely on the detection and characterization of individual facial features and their geometrical relationships. Such features generally include the eyes, nose, and mouth. The detection of faces and their features prior to performing verification or recognition makes these approaches robust to positional variations of the faces in the input image.

Holistic or global approaches to face recognition, on the other hand, involve encoding the entire facial image and treating the resulting facial "code" as a point in a high-dimensional space. Thus, they assume that all faces are constrained to particular positions, orientations, and scales. Even though holistic methods such as neural networks are more complex to implement than their geometric counterparts, their application is much more straight forward, whereby an entire image segment can be reduced to a few key values for comparison with other stored key values and no exact measures or knowledge such as eye locations or the presence of moustaches needs to be known. The problem with this "grab all" approach was that noise, occlusions such as glasses and any other non face image attribute could be learned by the holistic algorithm and become part of the recognition result even though such factors are not unique to faces.

Feature-based approaches were more predominant in the early attempts at automating the process of face recognition. Some of this early work involved the use of very simple image processing techniques (such as edge detection, signatures, and so on) for detecting faces and their features (see, for example, Sakai et al., 1969; Kelly, 1970). In Sakai et al. (1969), an edge map was first extracted from an input image and then matched to a large oval template, with possible variations in position and size. The presence of a face was then confirmed by searching for edges at estimated locations of certain features like the eyes and mouth. Kelly (1970) used an improved edge detector involving heuristic planning to extract an accurate outline of a person's head from various backgrounds.

1.2 Motivation

Face Recognition as a concept has evolved over the years and it has been successfully used in various applications in biometric systems for more than last two decades. The importance of utilizing biometrics to establish personal authenticity and to detect impostors has grown in the present scenario of global security concern. So there is a need for the development of a biometric system for personal identification, which fulfils the requirements for access control of secured areas and other applications like identity validation for social welfare, crime detection, ATM access, computer security, etc.

Variations in lighting conditions, pose and expression makes face recognition an even more challenging and difficult task.

A lot of techniques have been applied for different applications. Robustness and reliability becomes more and more important for these applications especially in security systems.

So in this work we have studied the different approaches to the Face Recognition via Discrete Cosine Transform (DCT) and Principal Component Analysis and tried to compare the success rate of both the algorithms on a standard database namely ORL (OlivettiResearch Lab).

1.3 Achievements

In this work we have we have used Discrete Cosine Transform and Principal Component Analysis for Face Recognition and tested the algorithms on a standard database ORL. In the process following things were achieved:

a) We studied and tested the Face Recognition System based on Discrete Cosine Transform

b) We studied and tested the Face Recognition System based on Principal Component Analysis

d) Euclidean distance classifier was used.

Chapter 2

Literature Review

2.1 Introduction to Matlab

MATLAB (**mat**rix **lab**oratory) is a multi-paradigm numerical computing environment and fourth-generation programming language. Developed by MathWorks, MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, Java, Fortran and Python.

Chapter 3

Problem Specification

3.1 Problem Statement

The general face recognizing task can be formally defined as task of approximately identifying an unknown face against a set of known images in a database.We have identified that there were many proposed systems similar to ours but the disadvantage they faced was the accuracy in the system, we could achieve a maximum accuracy of 95% using our system.

3.2 Proposed System

Our proposed system i.e. "Face Recognition System using machine learning" takes an image as input and recognizes this image against the set of training images.Now our system takes this input and constructs a feature vectors using Discrete Cosine Transform and Principal Component Analysis and hence compares this input against the images in our database.

If a match is found for the input image then it returns the class label and hence displays the nearest matching image.

Chapter 4

Requirement Specification

4.1 Hardware Requirements

Operating Systems - WindowsXP,Windows 7,Windows 8,Windows 8.1

4.2 Software Requirements

IDE - MATLAB

4.3 Constraints

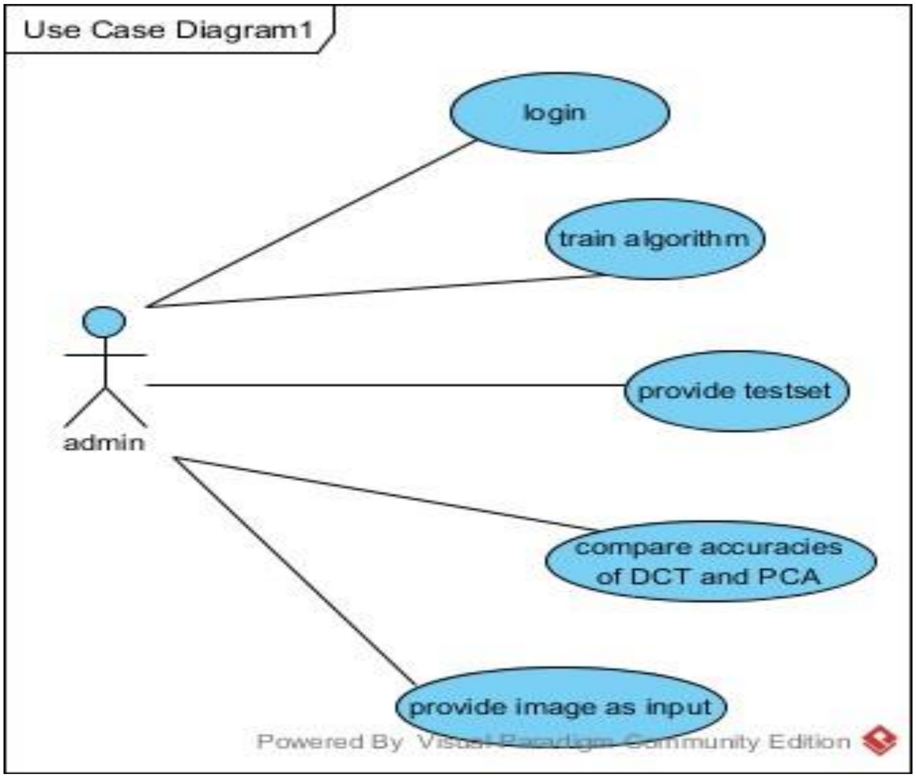
The database used here is ORL database.This system works on preprocessed database only.

Chapter 5

System Design

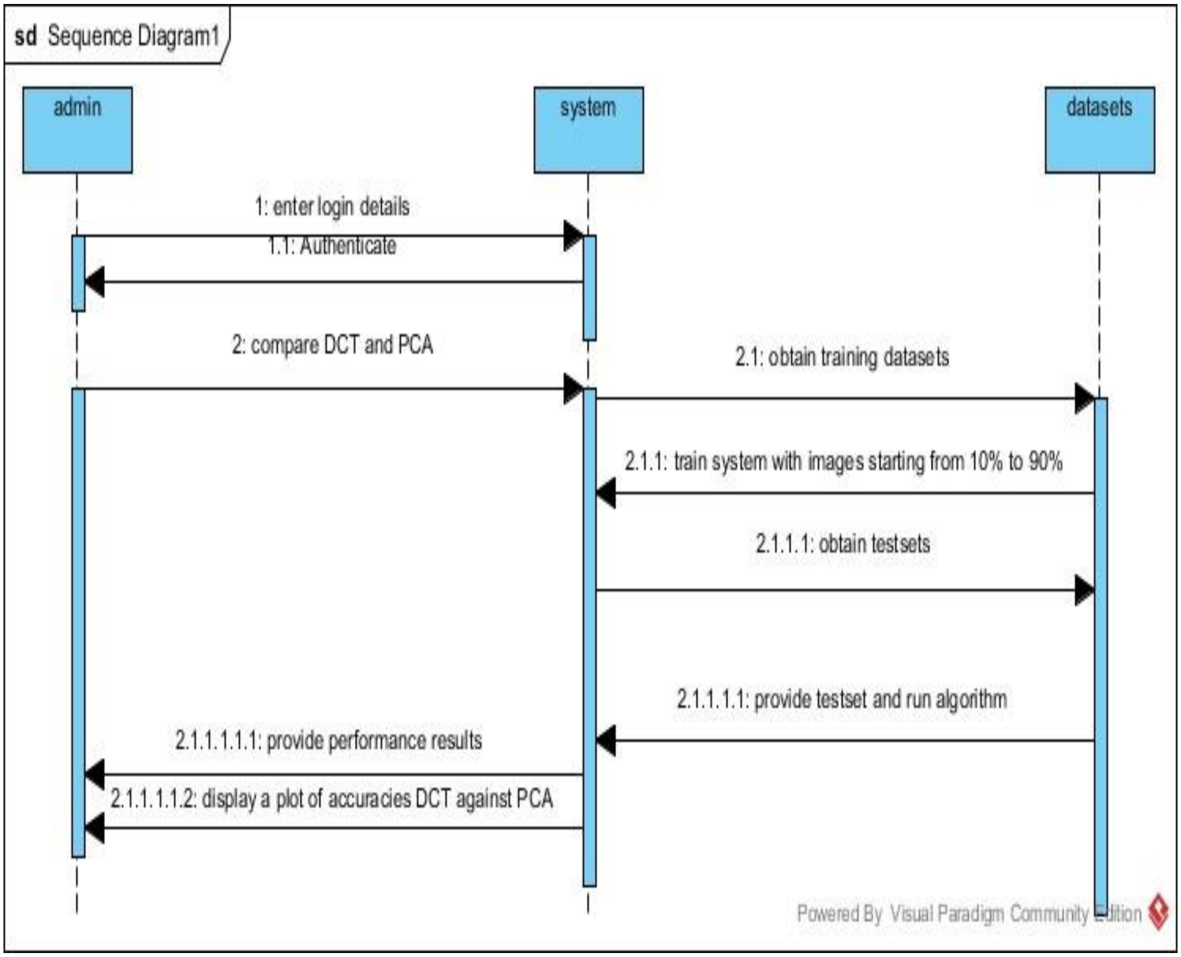
5.1 UML Diagrams

5.1.1Use Case Diagram

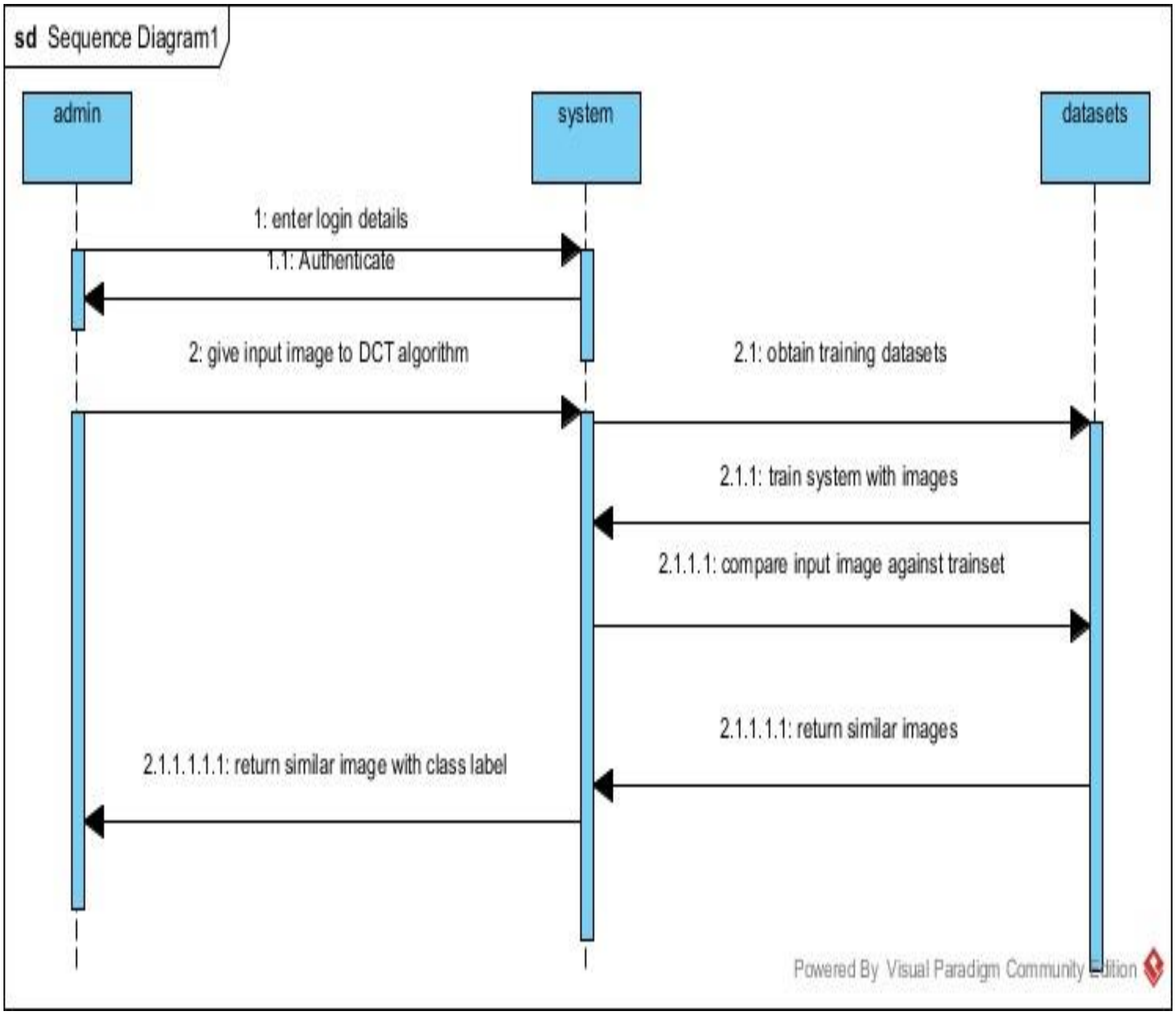


USE CASE NAME	Admin
Participating actor instances	admin
Entry conditions	Admin must have username and password in order to enter into the system
Flow of events	Admin should login first. He/she trains the system. He /she inputs an image so that the system recognizes it .

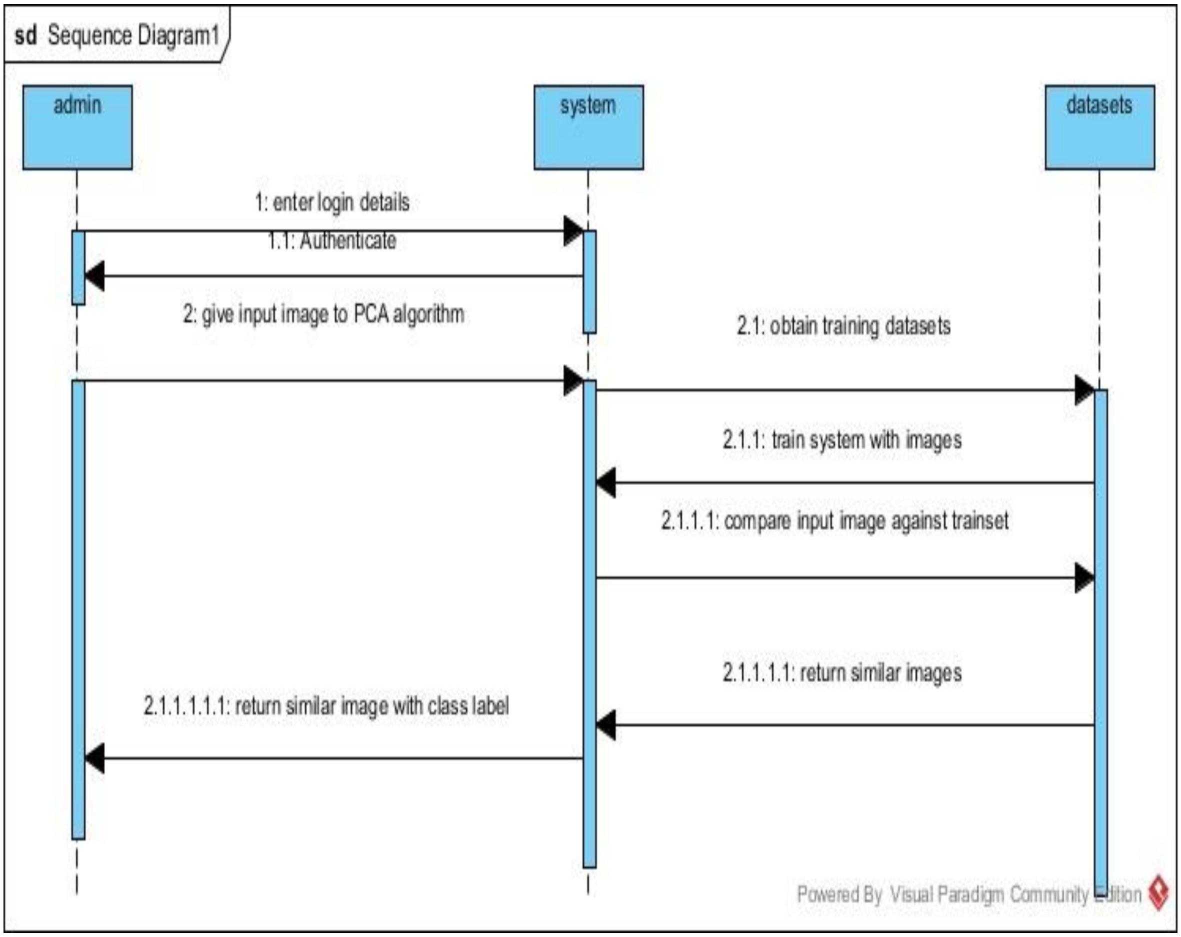
5.1.2 Sequence Diagram for comparing DCT and PCA


sd Sequence Diagram1
admin
system
datasets
1: enter login details
1.1: Authenticate
2: compare DCT and PCA
2.1: obtain training datasets
2.1.1: train system with images starting from 10% to 90%
2.1.1.1: obtain testsets
2.1.1.1.1: provide testset and run algorithm
2.1.1.1.1.1: provide performance results
2.1.1.1.1.2: display a plot of accuracies DCT against PCA
Powered By Visual Paradigm Community Edition

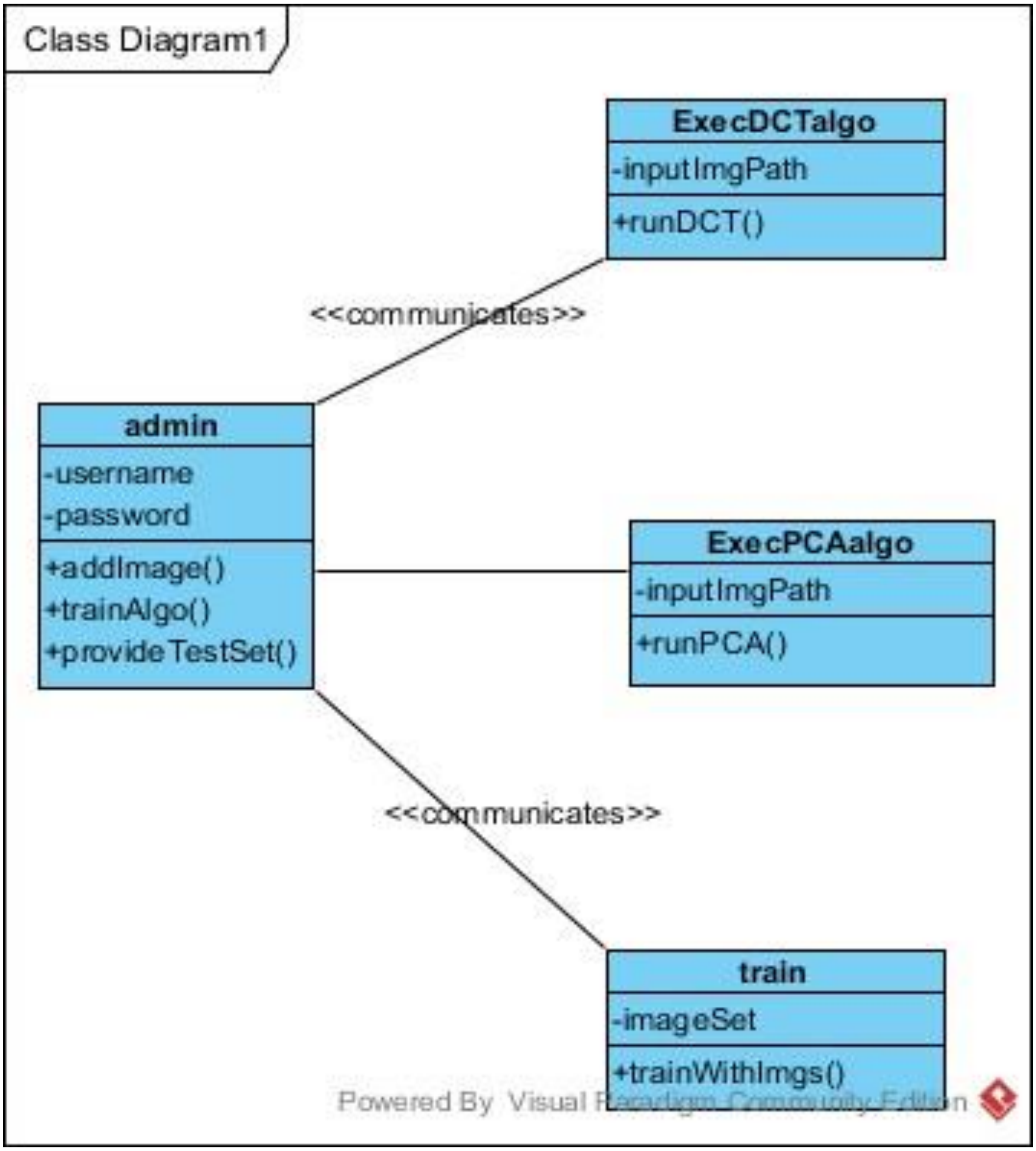
5.1.2.1 Sequence Diagram for DCT


sd Sequence Diagram1
admin
system
datasets
1: enter login details
1.1: Authenticate
2: give input image to DCT algorithm
2.1: obtain training datasets
2.1.1: train system with images
2.1.1.1: compare input image against trainset
2.1.1.1.1: return similar images
2.1.1.1.1.1: return similar image with class label
Powered By Visual Paradigm Community Edition

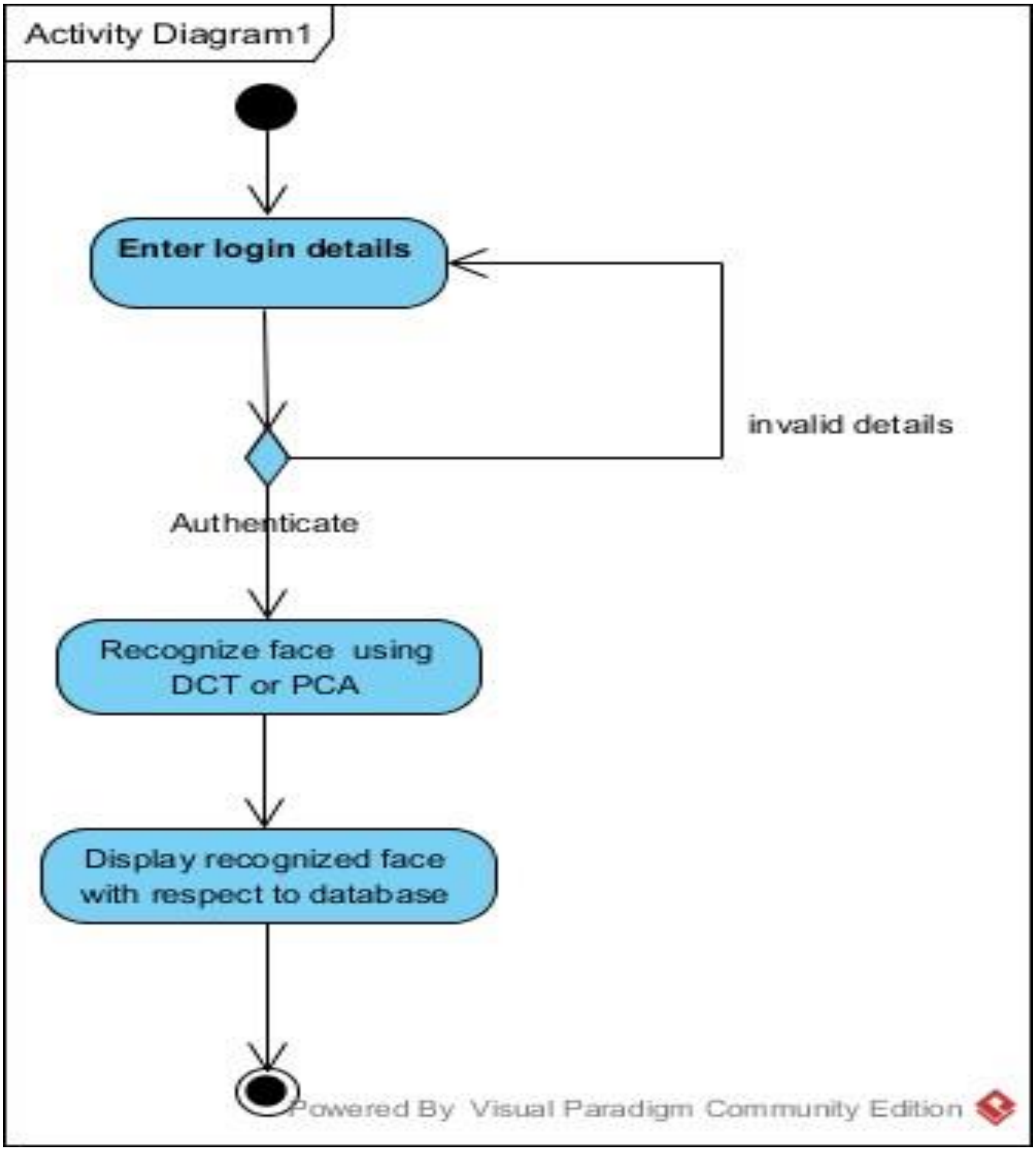
5.1.2.2 Sequence Diagram for PCA


sd Sequence Diagram1
admin
system
datasets
1: enter login details
1.1: Authenticate
2: give input image to PCA algorithm
2.1: obtain training datasets
2.1.1: train system with images
2.1.1.1: compare input image against trainset
2.1.1.1.1: return similar images
2.1.1.1.1.1: return similar image with class label
Powered By Visual Paradigm Community Edition

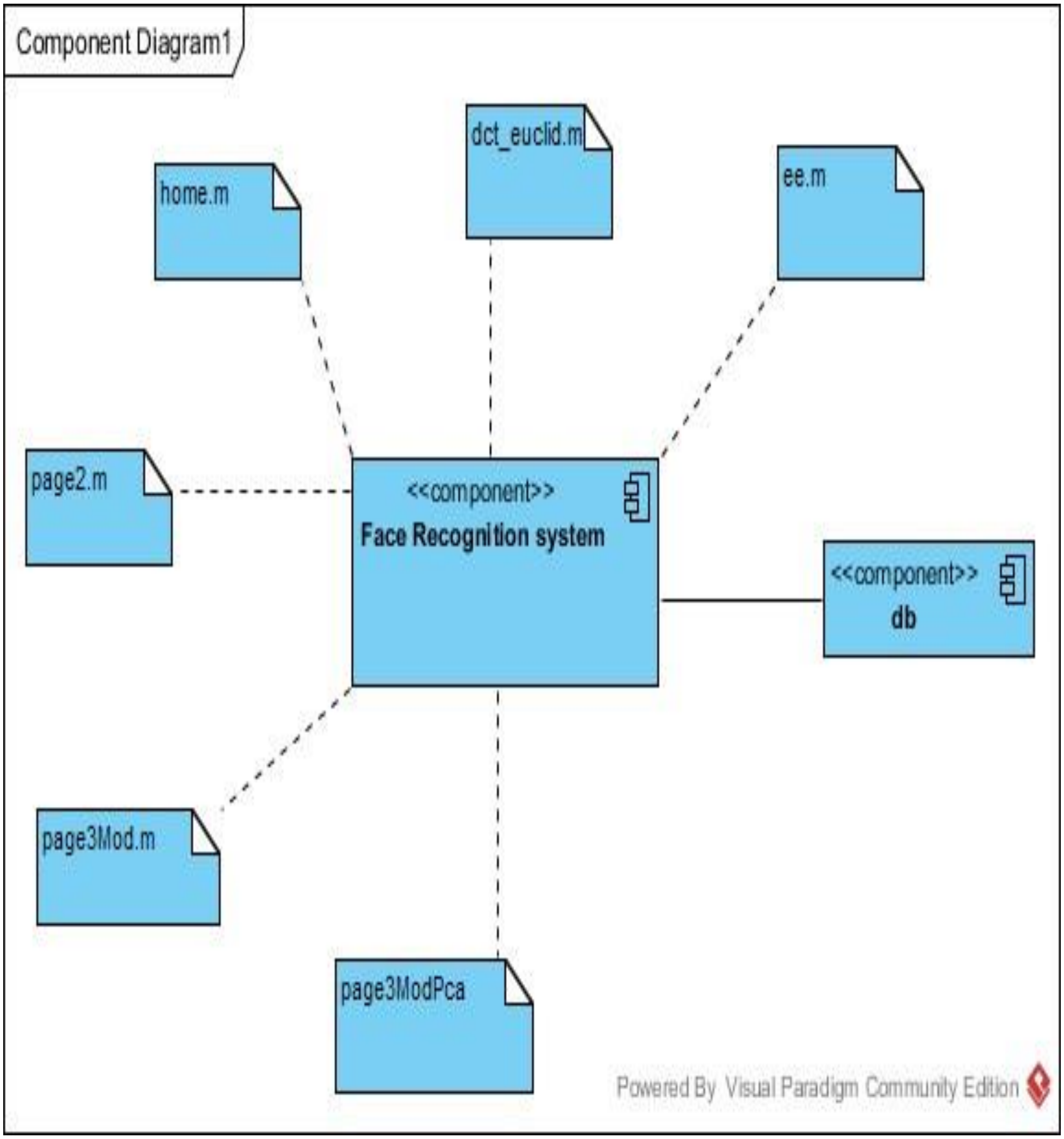
5.1.3 Class Diagram



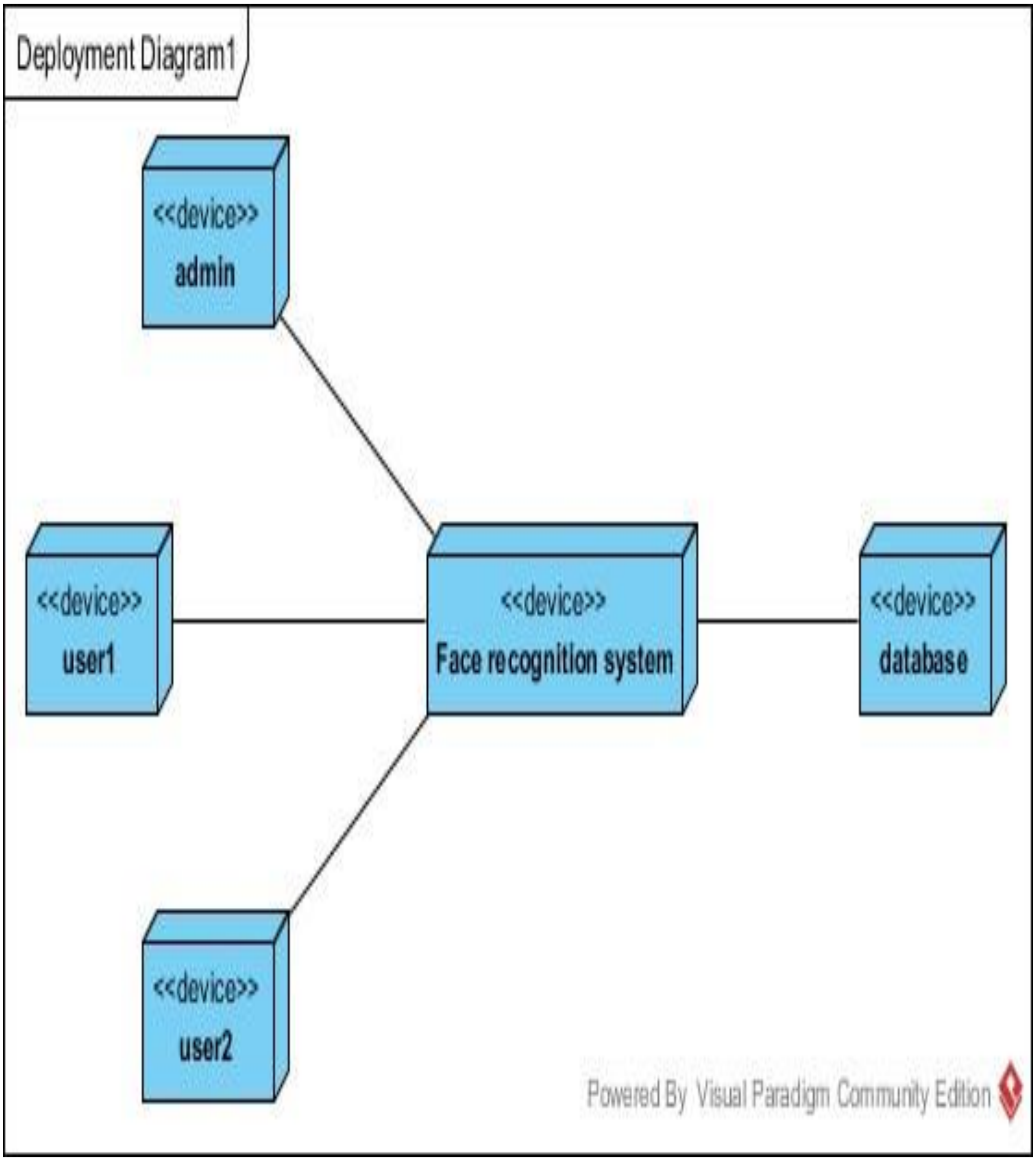
5.1.4Activity Diagram


Activity Diagram1
Enter login details
invalid details
Authenticate
Recognize face using DCT or PCA
Display recognized face with respect to database
Powered By Visual Paradigm Community Edition

5.1.5 Component Diagram



5.1.6Deployment Diagram



Chapter 6

System Analysis & Methodology

6.1 Face Recognition using DCT

6.1.1 Background

Data compression is very much essential for computer signal processing. Linear transforms play a very important role in the signal and image processing areas .

A transform is a mathematical operation which is applied to a signal that is being processed converting into different domain and then again is converted back to the original domain.

These transforms generate a set of coefficients from which it is possible to restore the original samples of the signal. A mathematical transform has an important property: when applied to a signal, i.e., they have the ability to generate de-correlated coefficients, concentrating most of the signal's energy in a reduced number of coefficients [7].

DCT is a very well known signal analysis tool used in compression standards due to its compact representation power. It has data independent nature. It is an invertible linear transform that expresses a finite sequence of data points in terms of a sum of cosine functions oscillating at different frequencies.

The DCT coefficients reflect the importance of frequencies that are present in it. The very first coefficient refers to the signal's lowest frequency and usually carries the majority of the relevant information from the original signal. The last coefficients refer to the signal's higher frequencies and these generally represent the more detailed or fine information of signal. The rest of the coefficients carry different information levels of the original signal. Since the DCT is related to the discrete Fourier transform (Rao and Yip, 1990), it can be computed efficiently. It is these properties of the DCT that we seek for face recognition.

6.1.2 Definition of DCT

Ahmed, Natarajan, and Rao (1974) first introduced the discrete cosine transform (DCT) in the early seventies. Ever since, the DCT has grown in popularity,
and several variants have been proposed (Rao and Yip, 1990). In particular, the DCT was categorized by Wang (1984) into four slightly different transformations named DCT-I, DCT-II, DCT-III, and DCT-IV [3].

Given an input sequence *u (n)* of length *N*, its DCT, *v (k)*, is obtained by the following equation:

$$v(k) = \alpha(k)\sum_{n=0}^{N-1} u(n)\cos\left(\frac{(2n+1)\pi k}{2N}\right) \qquad (2.1a)$$

$$0 \le k \le N-1$$

where

$$\alpha(0) = \sqrt{\frac{1}{N}}, \alpha(k) = \sqrt{\frac{2}{N}} \qquad (2.1b)$$

$$1 \le k \le N-1$$

Alternatively, we can think of the sequence *u (n)* as a vector and the DCT as a transformation matrix applied to this vector to obtain the output *v (k)*. In this case, the DCT transformation matrix, *C* = {*c (k, n)*}, is defined as follows:

$$c(k,n) = \begin{cases} \sqrt{\frac{1}{N}} k = 0, 0 \le n \le N-1 \\ \sqrt{\frac{2}{N}} \cos\left(\frac{(2n+1)\pi k}{2N}\right) 1 \le k \le N-1, 0 \le n \le N-1 \end{cases} \qquad (2.2)$$

where *k* and *n* are the row and column indices, respectively.

Using Eq. (2.2), the DCT of the sequence *u (n)* (or vector **u**) is simply

$$\mathbf{v} = C\mathbf{u} \qquad (2.3)$$

The inverse discrete cosine transform permits us to obtain *u (n)* from *v (k)*. It is defined by:

$$u(n) = \sum_{n=0}^{N-1} \alpha(k)v(k)\cos\left(\frac{(2n+1)\pi k}{2N}\right) \qquad (2.4)$$

$$0 \le n \le N-1$$

with *α(k)* as given in Eq. (2.1b). Using Eq. (2.3), the inverse discrete cosine transform, **u**, of a vector **v** is obtained by applying the inverse of matrix *C* to **v**. That is, the inverse discrete cosine transform is found from

$$u = C^{-1}v \qquad (2.5)$$

From these definitions, we observe that by applying the discrete cosine transform to an input sequence, we simply decompose it into a weighted sum of basis cosine sequences. This is obvious from Eq. (2.4) in which *u (n)* is reconstructed by a summation of cosines which are weighted by the DCT coefficients obtained from Eq. (2.1) or (2.3). These basis sequences of the

DCT are the rows of the matrix *C*.

6.1.3 Basic Algorithm

The basic Face Recognition Algorithm is discussed below as depicted in Fig: 2.1. Both normalization and recognition are involved in it. As can be seen from Fig. 2.1, the system receives as input an image containing a face .The normalized (and cropped) face is obtained and then it can be compared to other faces, under the same nominal size, orientation, position, and illumination conditions. This comparison is based on features extracted using the DCT. The basic idea here is to compute the DCT of the normalized face and retain a certain subset of the DCT coefficients as a feature vector describing this face.

This feature vector contains the low-to-mid frequency DCT coefficients, as these are the ones having the highest variance and contain the maximum information.

For recognizing a particular input face, the system compares the face’s feature vector to the feature vectors of the database faces using a Euclidean distance nearest-neighbour classifier (Duda and Hart, 1973).

If the feature vector of the probe is **v** and that of a database face is **f**, then the Euclidean distance between the two is

$$d = \sqrt{(f_0 - v_0)^2 + (f_1 - v_1)^2 + ... + (f_{M-1} - v_{M-1})^2}$$

where

$$v = [v_0, v_1, ...v_{M-1}]^T$$

$$f = [f_0, f_1, ...f_{M-1}]^T$$

and *M* is the number of DCT coefficients retained as features. A match is obtained by minimizing *d*.

This approach computes the DCT on the entire normalized image. This is different from the use of the DCT in the JPEG compression standard (Pennebaker and Mitchell, 1993), in which the DCT is computed on individual subsets of the image.
So in this approach we haven’t assumed any threshold on *d*. So the system described always assumes that the closest match is the correct match, and no probe is ever rejected as unknown.

If a threshold *q* is defined on *d*, then the gallery face that minimizes *d* would only be output as the match when $d < q$. Otherwise, the probe would be declared as unknown. In this way, we can actually define a threshold to achieve 100% recognition accuracy, but, of course, at the cost of a certain number of rejections.

In other words, the system could end up declaring an input face as unknown even though it exists in the gallery. Suitable values of *q* can be obtained using the so-called Receiver Operating Characteristic curve (ROC).

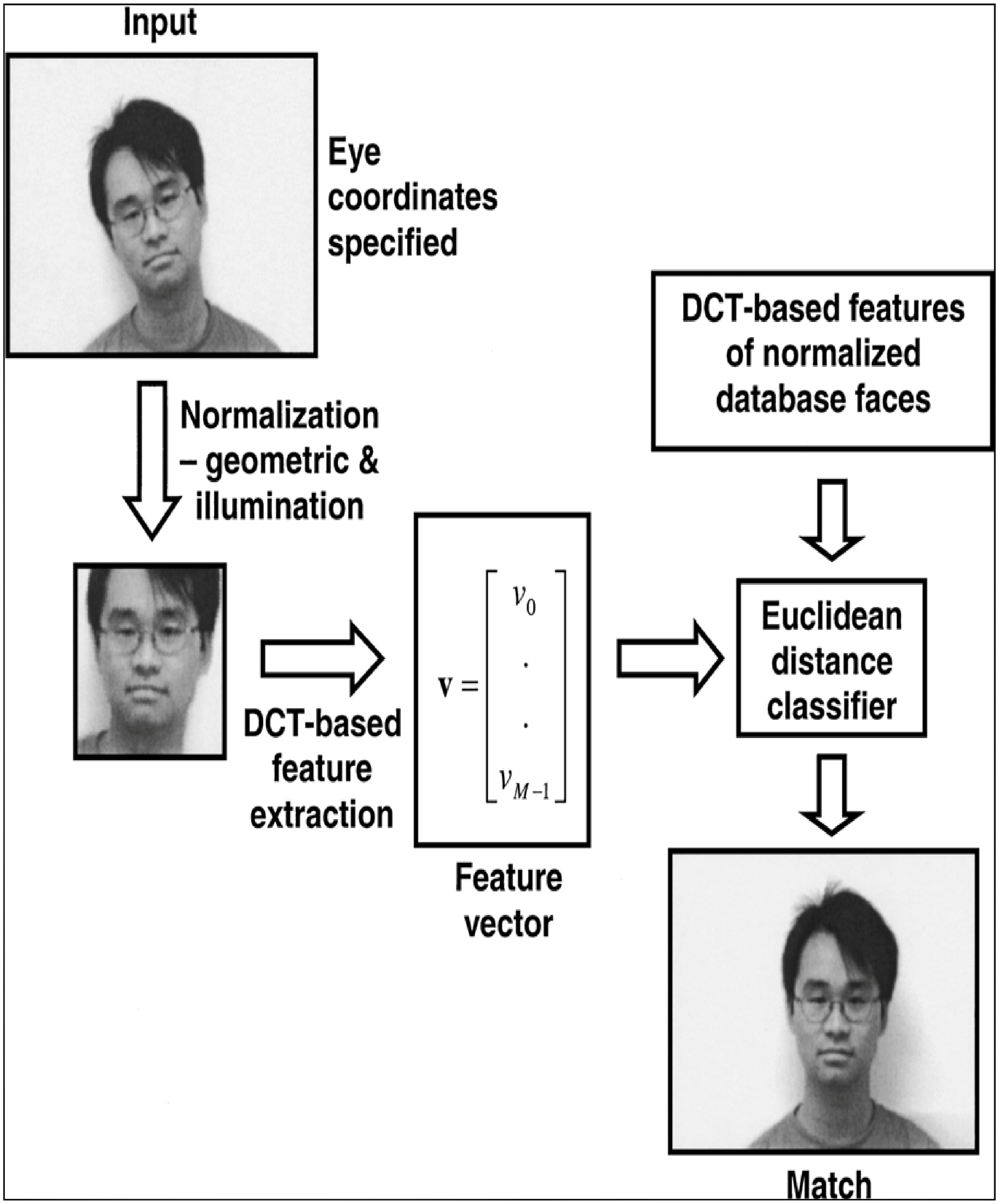

Input
Eye coordinates specified
Normalization – geometric & illumination
DCT-based feature extraction
$\mathbf{v} = \begin{bmatrix} v_0 \\ \cdot \\ \cdot \\ v_{M-1} \end{bmatrix}$
Feature vector
DCT-based features of normalized database faces
Euclidean distance classifier
Match

Fig 2.1: DCT Based Face Recognition System

6.2.1 Feature Extraction

The features are extracted with the help of DCT. So the DCT is computed for obtaining the feature vector representing a face and only a subset of the obtained coefficients is retained.

The size of this subset is chosen such that it can sufficiently represent a face. It is observed that the DCT coefficients exhibit the expected behavior in which a relatively large amount of information about the original image is stored in a fairly small number of coefficients. Most of the discarded coefficients have magnitudes less than 1.

So the main concept taken into account by the proposed feature selection stage is that low frequency DCT coefficients do concentrate more energy than others. Also it is not true that these high amplitude coefficients are always located in the lower part of the spectrum.

The database which we have used, there is no need of pre-processing. In this way we have reduced the computational complexity. So direct DCT based features are extracted.

We have used a standard database, ORL (Olivetti Research Lab) which contains 400 images (each with 112×92 pixels) corresponding to 40 persons in 10 poses each including both male and female. The ORL database image has been resized to 128×128 pixels.

All the images are taken against a dark homogeneous background, little variation of illumination, slightly different facial expression and details (open/close eyes, smiling/non smiling, glasses/no glasses etc.)

6.2.1.1 Euclidean Distance Classifier

For ORL 150 DCT coefficients are taken as there is very large variation. .

Above approach used in the selection of DCT coefficients is fast and simple.

6.2.2 Testing

The testing was done as described in the Fig 2.2. Euclidean distance classifier was used for the ORL database.

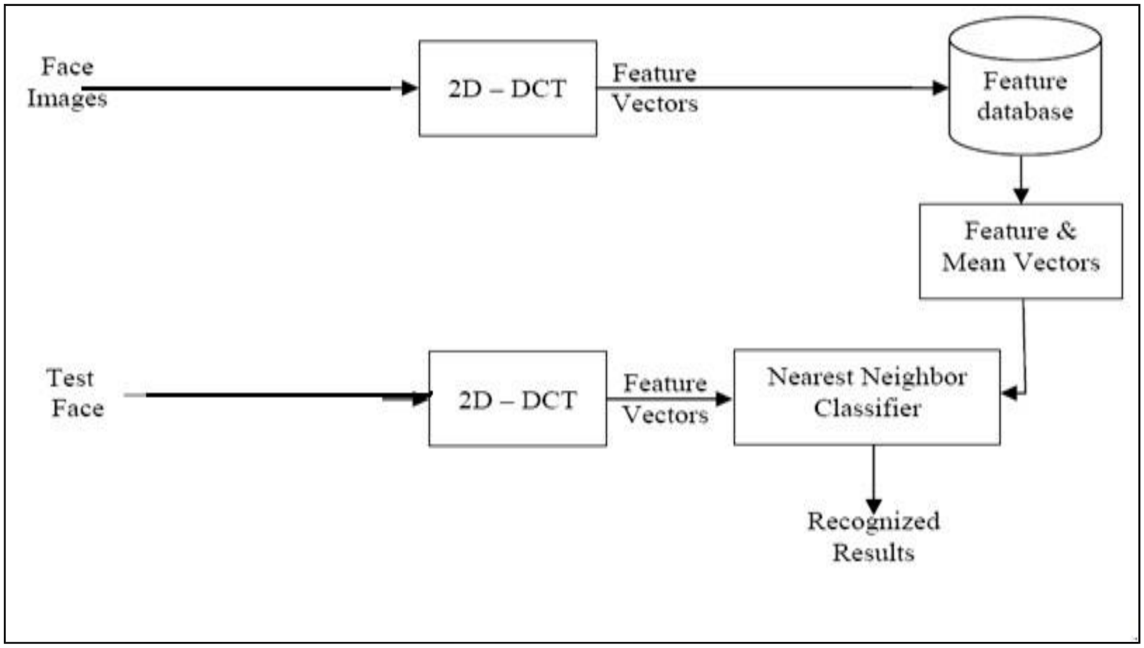


Fig 6.2: Face Recognition System using DCT & Euclidean Distance Classifier

6.2.3 Experimental Set up

6.2.3.1 Euclidean Distance Classifier

The whole experiment was done with the help of the standard database ORL.

ORL (Olivetti Research Lab) contains 400 images (each with 112×92 pixels) corresponding to 40 persons in 10 poses each including both male and female. The ORL database image has been resized to 128×128 pixels.
For ORL 150 DCT coefficients were used.

For each and every person, the number of training faces has been varied from 1 to 9 and correspondingly, the number of test faces from 74 to 64.

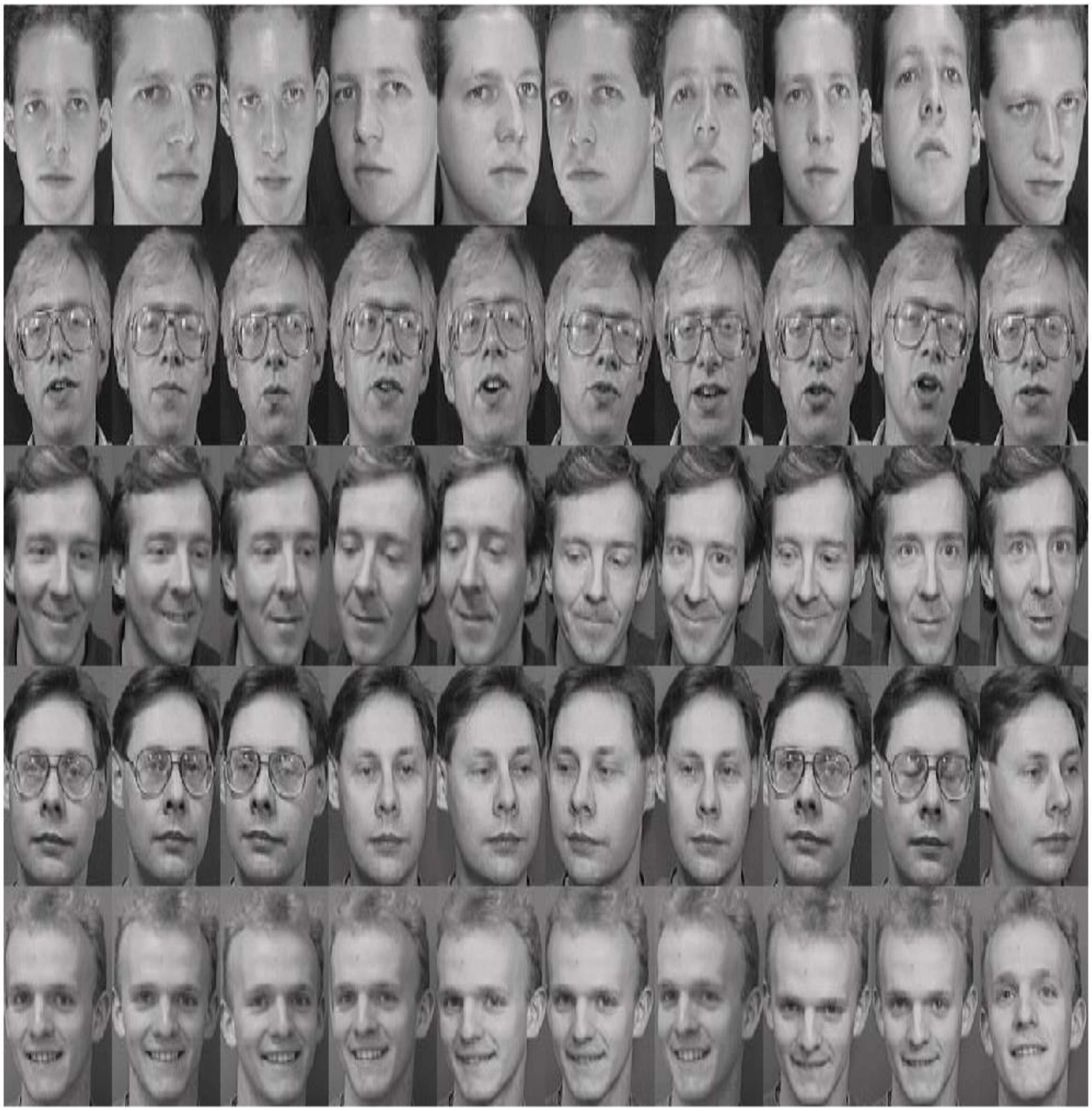


Fig 6.2: Views included in ORL Database

6.2.4Experimental Results

Percentage of images for training	Number of images for training /per person	DCT Performance
10	1	69.72
20	2	80.93
30	3	84.64
40	4	87.91
50	5	91
60	6	97.5
70	7	96.66
80	8	96.25
90	9	95

6.3 Conclusion

For ORL face database ,the Table 2.3 shows success rate to be around 82%.
The reason is clear for less success rate with this face data base. In ORL face data base there are large pose variations. So DCT based recognition system is not giving high success rate. So from the above discussion, we can conclude that though DCT based recognition system is simple, but is not suitable for the conditions where there are large pose
or illumination variations.

6.4 Feature Extraction using PCA

6.4.1
Background

PCA is a statistical dimensionality reduction method, which produces the optimal linear squares decomposition of a training set.It reduces the dimensionality of a data set while retaining the majority of the variation present in the data set (Jolliffe 1986).

It has been successfully used in Face Recognition systems. In the case of Face Recognition system based on PCA, it seeks to capture the variation in a collection of face images and use this information to encode and compare images of individual faces in a holistic manner. Eigen- face Method is verified well for the recognition strategy and in controlled condition.

An image space can be considered as a space having dimensions equal to the number of pixels making up the image and having values in the range of the pixels values and an image can be thought as a point in the image space by converting the image to a long vector by concatenating each column of the image one after the other [4].

When all the face images are converted into vectors, they will group at a certain location in the image space as they have similar structure, having eye, nose and mouth in common and their relative position correlated. This correlation is the main point to start the Eigen-face analysis.

The Eigen-face method tries to find a lower dimensional space for the representation of the face images by eliminating the variance due to non-face images; that is, it tries to focus on the variation just coming out of the variation between the face images. So Eigen-face method aims to build a face space which better describes the faces. The basis vectors of this face space are called the principal component and the Eigen-face method is the implementation of Principal Component Analysis (PCA) over images.

6.4.2 Eigen-faces for Recognition

In the language of Information theory, we want to extract the relevant information in a face image, encoding with a database of models encoded similarly as efficiently as possible, and compare one face encoding with a database of models, encoded similarly. So, mathematically we wish to find the principal components of the distribution of faces, or the Eigen-vectors of the covariance matrix of the set of face images.

A simple approach to extract the information contained in an image of a face is to somehow capture the variation in a collection of face images, independent of any judgment of features, and use this information to encode and compare individual face images.

In mathematical terms, Eigen-face method finds the principal components of the distribution of faces, or the Eigen-vectors of the covariance matrix of the set of face images, treats an image as point (or vector) in a very high dimensional space. The Eigen-vectors are ordered, each one accounting for a different amount of the variation among the face images [5].

These Eigen-vectors can be thought of as a set of features that together characterize the variation between face images. Each image location contributes more or less to each Eigen-vector, so that we can display the Eigen-vector as a sort of ghostly face which is why we call this by Eigen-face. Each Eigen-face deviates from uniform gray where some facial feature differs among the set of training faces; they are a sort of map of the variations between faces. Each individual face can be represented exactly in terms of a linear combination of the Eigen-faces.

The number of possible Eigen-faces is equal to the number of face images in the training set. However we can also represent the faces by approximating these by the best Eigen-faces having largest Eigen-values which in turn account for the most variance within the set of face images. This increases the computational efficiency.

The following steps are involved in the recognition process [5]:

1) Initialization: The training set of face images is acquired and Eigen-faces are calculated which define the face space.
2) When a new face is encountered, a set of weights based on input image and M Eigen-faces is calculated by projecting the input image onto each of the Eigen-faces.
3) The image is determined to be face or not by checking if it is sufficiently close to face space.
4) If it is a face, the weight patterns are classified as either a known person or an unknown one.

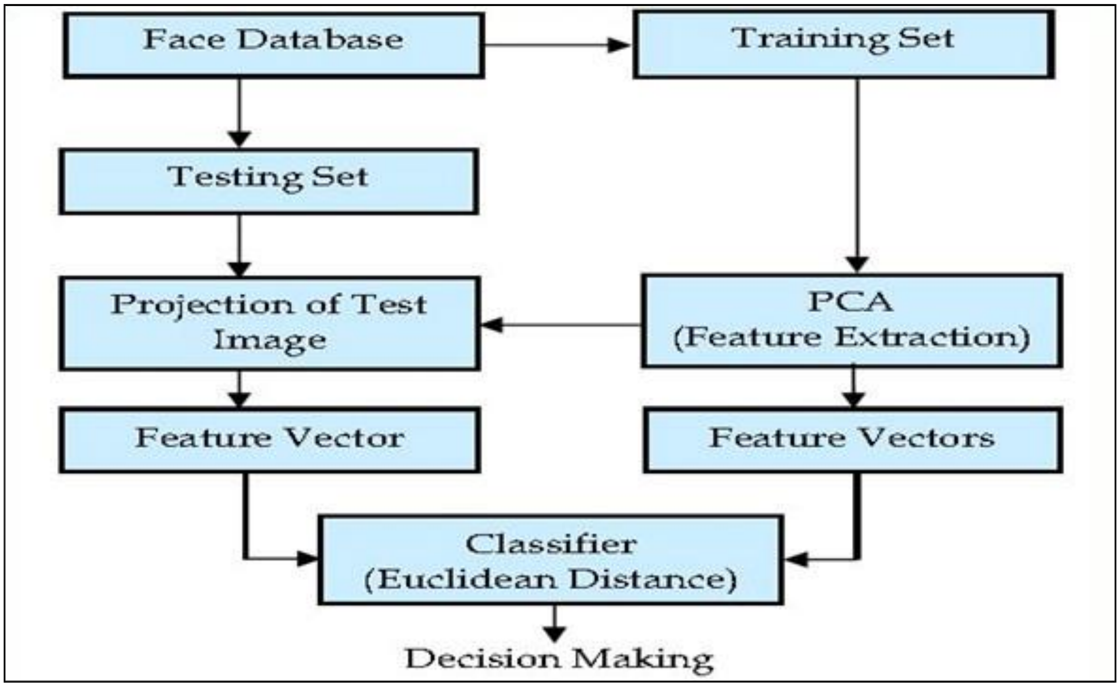


Fig 3.1: Face Recognition System using PCA

6.4.2.1 Calculating Eigen-faces

Since the images of faces are similar in overall configuration, they are not randomly distributed in the image space and thus are described by a relatively low dimensional subspace. The main idea of the PCA is to find the vectors which best account for the distribution of face images within the entire image space [4].

Mathematically, let the image be denoted by I.

Image I: (N x N) pixels (3.1)

Now the image matrix I of size (N x N) pixels is converted to the image vector Γ of size (P x 1) where P = (N x N); that is the image matrix is reconstructed by adding each column one after the other.

Let the training set be denoted by Γ

Training Set: $\Gamma = [\Gamma_1\ \Gamma_2\ ...\ \Gamma_M]$ (3.2)

is the training set of image vectors and its size is (P x M) where M is the
number of the training images.
Now the Mean face is calculated by the equation:

Mean Face: $\Psi = \frac{\square}{?}\sum_{\square ? \square}^{\square ? ?}\Gamma ?$ (3.3)

is the arithmetic average of the training image vectors at each pixel point and its size is P×1.

Mean Subtracted Image: $\Phi = \Gamma - \Psi$ (3.4)

is the difference of the training image from the mean image.

Difference Matrix: $A = [\Phi_1\ \Phi_2\ ...\ \Phi_M\ ]$ (3.5)

is the matrix of all the mean subtracted training image vectors and its size is (P×M).

Covariance Matrix: $X = A \cdot A^T = \frac{1}{M}\sum_{i=1}^{M}\varphi_i\varphi_i^T$ (3.6)

is the covariance matrix of the training image vectors of size (P×P).

An important property of the Eigen-face method is obtaining the Eigen-vectors of the covariance matrix. For a face image of size (N x N) pixels, the covariance matrix is of size (P x P), P being (N x N). This covariance matrix is very hard to work with due to its huge dimension causing computational complexity.

On the other hand, Eigen-face method calculates the Eigen-vectors of the (M x M) matrix, M being the number of face images, and obtains (P x P) matrix using the Eigen-vectors of the (M x M) matrix.

Initially, a matrix Y is defined as,

$Y = A^T.A = \frac{1}{M}\sum_{i=1}^{M}\Gamma_i\Gamma_i^T$ (3.7)

Which is of size (M×M).

Then the Eigen-vectors v_i and Eigen-values μ_i are obtained,

$Y.\ v_i = \mu_i\ .v_i$ (3.8)

The value of Y is put in this equation,

$A^T .A\ .v_i = \mu_i\ .v_i$ (3.9)

Now both the sides are multiplied by A on left side,

$A.\ A^T .A.v_i = A.\ \mu_i\ .v_i$ (3.10)

which can be represented as

$A.\ A^T .A\ .v_i = \mu_i\ .A.v_i$ (3.11)

$X\ .A.v_i = \mu_i\ .A.v_i$ (3.12)

Now let us group $A.v_i$ and call it υ_i
It is now easily seen that

$\upsilon_i = A.\ v_i$ (3.13)

is one of the Eigen-vectors of $X= A.A^T$

Thus, it is possible to obtain the Eigen-vectors of X by using the Eigen-vectors of Y. A matrix of size (M x M) is utilized instead of a matrix of size (P x P) (i.e. [{N x N} x {N x N}]). This formulation brings substantial computational efficiency.

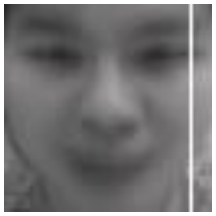


Fig 6.3: Mean training face

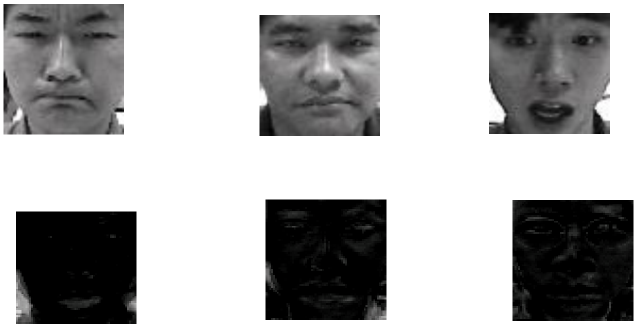


Fig 6.4: Eigen-faces

Instead of using M of the Eigen-faces, M'≤ M of the Eigen-faces can be used for the Eigen-face projection. This is achieved to eliminate some of the Eigen-vectors with small Eigen-values, which contribute less variance in the data.

In the next step, the training images are projected into the Eigen-face space and thus the weight of each eigenvector to represent the image in the Eigen-face space is calculated. This weight is simply the dot product of each image with each of the Eigen-vectors.

Projection: $w_k=v_k$ (3.14)

is the projection of a training image on each of the Eigen-vectors where k=1,2,3,.... M′

Weight Matrix: $\Omega= [w_1\ w2\ \ldots\ w_{M'}]^T$ (3.15)

is the representation of the training image in the Eigen-face space and its size is M×1.

So the images are just composed of weights in the Eigen-face space, simply like they have pixel values in the image space. The important aspect of the Eigen-face transform lies in this property. Each image is represented by an image of size (N x N) in the image space, whereas the same image is represented by a vector of size (M′ x 1) in the Eigen-face space. Moreover, having the dimension structure related to the variance of the data in hand makes the Eigen-face representation a generalized representation of the data.

This makes the algorithm a solution to the curse of dimensionality problem.

6.4.2.2 Classification of test image

For the classification of a new test image, it is also mean subtracted first and projected onto the Eigen-face space and then Nearest Mean algorithm [10] is used for the classification of the test image vector in the standard Eigen-face method; that is, the test image is assumed to belong to the nearest class by calculating the Euclidean distance of the test image vector to the mean of each class of the training image vectors.

Test image vector: Γ_T (3.16)

is the test image vector of size P×1.

Mean subtracted image: $\Phi = \Gamma_T - \Psi$ (3.17)

is the difference of the test image from the mean image of size P×1.

Projection = $v_k{}^{T}\text{ʓ}\Phi = v_k{}^{T}\text{ʓ}\,(\Gamma - \Psi)^{T}$ (3.18)

is the projection of a training image on each of the Eigen-vectors where k=1,2,3,... M′

Weight Matrix: $\Omega_T = [w_1 \; w_2 \; ... \; w_{M'}]^{T}$ (3.19)

is the representation of the test image in the Eigen-face space and its size is
(M′ x 1).

A similarity measure is defined as the Euclidean distance between the test image vector and i^{th} face class.

6.4.3 Experimental Set up

The whole experiment was done with the help of the database ORL.

ORL (Olivetti Research Lab) contains 400 images (each with 112×92 pixels) corresponding to 40 persons in 10 poses each including both male and female. The ORL database image has been resized to 128×128 pixels.

6.4.4 Experimental Results

For the ORL database, testing is done both by varying the number of the training face per class. The result is tabulated as shown in the Table 3.2.

Table 3.2

Percentage of images for training	**Number of images for training /per person**	**PCA Performance**
10	1	62.22
20	2	70.62
30	3	72.85
40	4	76.66
50	5	78
60	6	89.37
70	7	91.67
80	8	93.75
90	9	95

6.4.5 Conclusion

PCA is basically a technique to represent the feature vector in the lower dimensionality space. So, by considering all the pixel values of image as the feature vector, we are getting better representation of image. That is why this technique is working better than DCT based technique even in large pose and illumination variations. This is evident from the results shown for both AMP face database as well as for ORL database in Table 3.1 and Table 3.2 respectively.
So PCA based face recognition system is appropriate for both types of database ,one having less variations and the other having large variations. But the problem is the complexity of the
PCA based system.

Chapter 7

Coding

Home.m

```
function pushbutton1_Callback(hObject, eventdata, handles)
x = get(handles.edit1,'String');
y = get(handles.edit2,'String');
 if isempty(x)||isempty(y)
  uiwait(warndlg('Error: Enter some Text into pasword or username
field'));
 else
     match1 = strcmp(x,'projectX');
     match2 = strcmp(y,'secret');

        if match1 == 1 && match2 == 1
        close all force;
        page2
        else
        uiwait(warndlg('Error:Invalid Details'));
        end
 end
```

page2.m

```
function pushbutton3_Callback(hObject, eventdata, handles)
ee



% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
close all force;
page3ModPca

function pushbutton1_Callback(hObject, eventdata, handles)
close all force;
page3Mod
```

page3Mod.m

```
function pushbutton1_Callback(hObject, eventdata, handles)
[filename pathname]=uigetfile('*.*','Select An Image');
s=strcat(pathname,filename);
sinput=strcat('db\',filename);
set(handles.text1,'String',s);
mat=imread(s);
x=dct_gui(mat);
output =strcat( 'db\s',int2str(x),' (1).pgm');
set(handles.text3,'String',output);
axes(handles.axes1);
imshow(sinput);
axes(handles.axes2);
imshow(output);
```

page3ModPca.m

```
function pushbutton1_Callback(hObject, eventdata, handles)
[filename pathname]=uigetfile('*.*','Select An Image');
s=strcat(pathname,filename);
sinput=strcat('db\',filename);
set(handles.text1,'String',s);
mat=imread(s);
x=pca_gui(mat);
output =strcat( 'db\s',int2str(x),' (1).pgm');
set(handles.text3,'String',output);
axes(handles.axes1);
imshow(sinput);
axes(handles.axes2);
imshow(output);

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
close all force;
page2
```

dct_fun.m

```
function  dct_per= dct_fun( x )

zz=1;
noc=40;             %no_of_classes
nots=x;                %no_of_training_set
[face,MAP]=imread('db\s1 (1).pgm');
[a,b]=size(face);
y_train=[];
% Reading face from the databese for training set
counter=0;
for i=1:noc
    for j=1:nots
        file=['db\s' int2str(i) ' (' int2str(j) ').pgm'];
        grayface=imread(file);
        grayface = im2double(grayface);
        T = dctmtx(8);
        B = blkproc(grayface,[8 8],'P1*x*P2',T,T');
        mask = zeros(8,8);mask(1,1)=1;
        B2 = blkproc(B,[8 8],'P1.*x',mask);
        vector_face=zigzag(B2);
        vector_face=vector_face(:);
        vector_face=removezeros(vector_face);
        counter=counter+1;
        Covar_train(:,counter)=vector_face;
    end
end
```

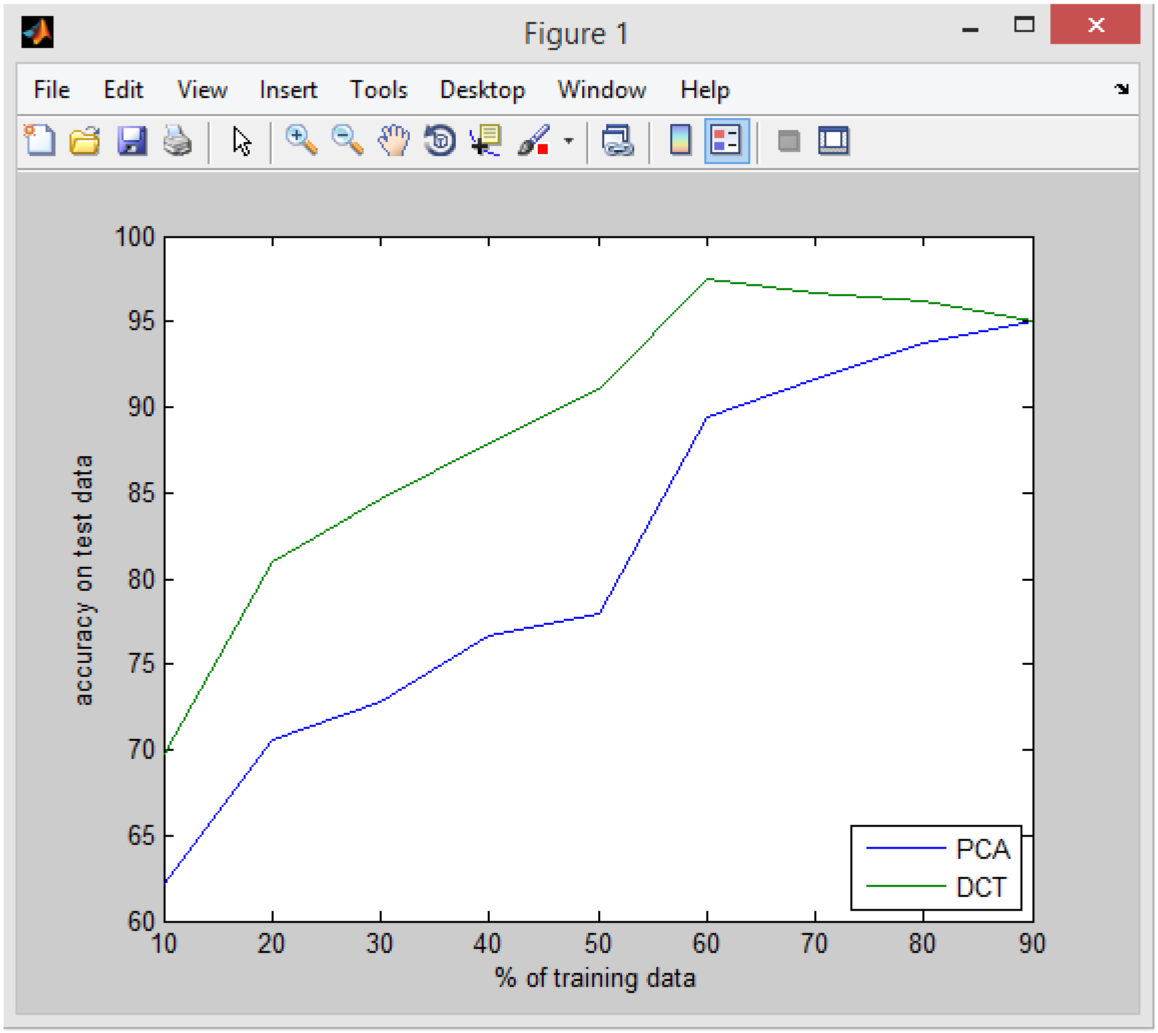
```
% Reading face from the databese for test set
counter=0;
for i=1:noc
    for j=nots+1:10
        file=['db\s' int2str(i) ' (' int2str(j) ').pgm'];
        grayf=imread(file);
        grayf = im2double(grayf);
        T = dctmtx(8);
        B = blkproc(grayf,[8 8],'P1*x*P2',T,T');
        mask = zeros(8,8);mask(1,1)=1;
        B2 = blkproc(B,[8 8],'P1.*x',mask);
        vector_face=zigzag(B2);
        vector_face=vector_face(:);
        vector_face=removezeros(vector_face);
        counter=counter+1;
        Covar_test(:,counter)=vector_face;
    end
end
clear memory
counter=0;

for i=1:noc*(10-nots)
    error=[];
    for j=1:noc*nots
        temp=(Covar_test(:,i)-Covar_train(:,j))';
        distance=sqrt(temp*temp');
        error=[error distance];
    end
    Minimum_Error=max(error);
    for k=1:noc*nots
        if error(k)<Minimum_Error
            Minimum_Error=error(k);
            holder=k;
        end
    end
    if ceil(holder/nots)==ceil(i/(10-nots))
        counter=counter+1;
    end
end

dct_per=(counter/(noc*(10-nots)))*100

end
```

Chapter 8

Conclusion

8.1 Comparison between DCT and PCA

From the above experiments the graph plotted displaying DCT and PCA algorithms are:



8.2 Conclusion

From the above experiments the following points can be concluded:

1) If there are very less pose and illumination variations in the faces then DCT based recognition is a very good option as it is very simple to implement and has less computation complexity. But if there are large variations in pose or illumination then DCT based recognition is not a good option ,as in this case to represent the image we have to take large number of DCT coefficients, hence the system will be complex. Also in large pose and illumination condition DCT coefficients do not represent the face well. So results is not that much satisfactory.

2) In such conditions simple PCA is a good choice, as representation by any of them is very good. So, we are getting good recognition rate.

References

References

[1] Zhi-Kai Huang, Wei-Zhong Zhang, Hui-Ming Huang, Ling-Ying Hou, *Using Gabor Filters Features for Multi-Pose Face Recognition in Color Images*, Second International Symposium on Intelligent Information Technology Application

[2] Aman R. Chadha, Pallavi P. Vaidya, M. Mani Roja *for* Face Recognition Using Discrete Cosine Transform for Global and Local Features, Proceedings of the 2011 International Conference on Recent Advancements in Electrical, Electronics and Control Engineering (IConRAEeCE) IEEE Xplore: CFP1153R-ART; ISBN: 978-1-4577-2149-6

[3] Surya Kant Tyagi and Pritee Khanna *for* Face Recognition Using Discrete Cosine Transform and Nearest Neighbor Discriminant Analysis, IACSIT International Journal of Engineering and Technology, Vol. 4, No. 3, June 2012

[4] ZIAD M. HAFED AND MARTIN D. LEVINE *for* Face Recognition Using the Discrete Cosine Transform, International Journal of Computer Vision 43(3), 167–188, 2001

[5] Urvashi Bakshi ,Rohit Singhal *for* A NEW APPROACH OF FACE RECOGNITION USING DCT, PCA, AND NEURAL NETWORK IN MATLAB, International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)

[6] K Manikantan1 , Vaishnavi Govindarajan , V V S Sasi Kiran , S Ramachandran *for* Face Recognition using Block-Based DCT Feature Extraction, Journal of Advanced Computer Science and Technology, 1 (4) (2012) 266-283 c Science Publishing Corporation