**CHAPTER 1**

**INTRODUCTION**

* 1. **DESCRIPTION**

Face recognition is a method of identifying or verifying the identity of an individual using their face. Face recognition systems can be used to identify people in photos, video, or in real-time. Face recognition systems use computer algorithms to pick out specific, distinctive details about a person’s face. These details, such as distance between the eyes or shape of the chin, are then converted into a mathematical representation and compared to data on other faces collected in a face recognition database. The data about a particular face is often called a face template and is distinct from a photograph because it’s designed to only include certain details that can be used to distinguish one face from another.In our project,  face recognition is performed using Linear Discriminant Analysis based dimension reduction techniques.

* 1. **EXISTING SYSTEM**

Face recognition algorithms identify faces by extracting landmarks or features from an image.For example, an algorithm may analyze the relative position, size, and/or shape of the eyes, nose, cheekbones and jaw. These features are then used to search for other images with matching features. Other algorithms normalize a gallery of face images and then compress the face data, only saving the data in the image that is useful for face detection.A probe image is then compared with the face data. One of the earliest successful systems is based on template matching techniques applied to a set of salient facial features, providing a sort of compressed face representation.

* 1. **PROBLEM DEFINITION**

This process attempts to make use of a Linear discriminant analysis(LDA) method to perform the task of face recognition.When the user gives an test image it should be able to recognize the face when the test face is already present in the database.The characteristic features called ‘Euclidean distance’ are calculated from the stored images using which the system is trained for subsequent recognition of new images.

* 1. **PROPOSED SYSTEM**

The proposed face recognition system is done by using Linear discriminant analysis(LDA) that overcomes the disadvantages of existing systems. It is based on extracting the dominating features of a set of human faces stored in the database and calculating Euclidean distance of the images.Hence when a new image is fed into the system for recognition the main features are extracted and computed to find the distance between the input image and the stored images.Thus, the system will be able to recognize the new face and identify who the person is.

* 1. **ORGANIZATION OF THE PROJECT**
* Literature review of already existing proposals are discussed in chapter 2.
* Chapter 3 has system specification which tells about the software and hardware requirements.
* Chapter 4 discuss the overall project and design which tells the brief description of each of the modulus in this project.
* Chapter 5 has the implementation and experimental result of the project.
* Chapter 6 deals with the conclusion and future work.
* Finally chapter 7 deals with the references.

**CHAPTER 2**

**LITERATURE REVIEW**

**2.1 Crime Identification using 3-D Face Recognition**

**2.1.1 DESCRIPTION**

The objective of this paper is to assess confront discovery and acknowledgment procedures and give a total answer for picture based face location and acknowledgment with higher exactness, better reaction rate and an underlying advance for video observation. Arrangement is proposed in light of performed tests on different face rich databases as far as subjects, stance, feelings and light.

**2.2.2 ALGORITHM USED**

* Neural Network.
* PCA(Principal Component Analaysis).

**2.1.3 MERITS**

* To Identify the criminals.
* Detect 3D images.

**2.1.4 DEMERITS**

* Less Performance with regardless of skin color.
* Less accuracy.

**2.2 Face Recognition with Partial Face Recognition and Convolutional Neural Network**

**2.3.1 DESCRIPTION**

Face recognition has been a dynamic research area in the pattern recognition and computer vision domains. Each face in this world has uniqueness. Therefore it is an identity of human. This identity due to its uniqueness can be used for authentication and access control in different application. In this review paper the face recognition and facial feature is a key area of investigation. Therefore this paper includes the survey of different recently developed face recognition technique and methods which are claimed to provide an effectiveand accurate method of face recognition. In addition of that a new model for recognizing face is also introduced in this paper. That model is implemented in near future and their performance is compared with similar approach.

**2.2.2 ALGORITHM USED**

* LDA(Linear Discriminant Analysis).
* CNN(Convolutional Neural Network).
* Deep Learning.

**2.2.3 MERITS**

* Cost Effective.
* Provide Authentication and secure access control.

**2.2.4 DEMRITS**

* Performance is comparatively less.
* Large Data set is needed**.**

**2.3 Face Recognition using Approximate Arithmetic**

**2.3.1 DESCRIPTION**

Throughout the years this ﬁeld evolved and there are many approaches and many diﬀerent algorithms which aim to make the face recognition as eﬀective as possible. The use of diﬀerent approaches such as neural networks and machine learning can lead to fast and eﬃcient solutions however, these solutions are expensive in terms of hardware resources and power consumption. A possible solution to this problem can be use of approximate arithmetic. In many image processing applications the results do not need to be completely precise and use of the approximate arithmetic can lead to reduction in terms of delay, space and power consumption. In this paper we examinepossible use of approximate arithmetic in face recognition using Eigenfaces algorithm.

**2.3.2 ALGORITHM USED**

* Approximate Arithmetic.
* Eigenfaces algorithm.

**2.3.3 MERITS**

* Less Memory space is acquired.
* Power Consumption is less.

**2.3.4 DEMRITS**

* Approximate Arithmetic is static.

**2.4 Facial Expression Recognition Using Constructive Feedforward Neural Networks**

**2.4.1 DESCRIPTION**

A new technique for facial expression recognition is proposed, which uses the two-dimensional (2-D) discrete cosine transform (DCT) over the entire face image as a feature detector and a constructive one-hidden-layerfeedforwardneural network asafacial expressionclassifier. An input-side pruning technique, proposed previously by the authors, is also incorporated into the constructive learning process to reduce the network size without sacrificing the performance of the resulting network. The proposed technique is applied to a database consisting of images of 60 men, each having five facial expression images (neutral, smile, anger, sadness, and surprise). Images of 40 men are used for network training, and the remaining images of 20 men are used for generalization and testing**.**

**2.4.2 ALGORITHM USED**

* Constructive Feedforward Neural Network.

**2.4.3 MERITS**

* Reduced network size.
* Confusion matrices allows to evaluate the trained network.

**2.4.4 DEMERITS**

* Less accuracy**.**

**CHAPTER 3**

**SYSTEM SPECIFICATION**

**3.1 SYSTEM REQUIREMENTS**

**3.1.1 HARDWARE REQUIREMENTS**

* System : Pentium IV 2.4 GHz.
* Hard Disk : 1 TB
* Ram : 512 Mb.

**3.1.2 SOFTWARE REQUIREMENTS**

* Operating system : Windows 10.
* Coding Language : MATLAB
* Tool : MATLAB R 2012

**3.2 SOFTWARE DESCRIPTION**

**3.2.1 ABOUT MATLAB**

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include:

* Math and computation
* Algorithm development
* Modeling, simulation, and prototyping
* Data analysis, exploration, and visualization
* Scientific and engineering graphics

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar noninteractive language such as C or Fortran.

The name MATLAB stands for matrix laboratory. MATLAB was originally written to provide easy access to matrix software developed by the LINPACK and EISPACK projects, which together represent the state-of-the-art in software for matrix computation.

MATLAB has evolved over a period of years with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science. In industry, MATLAB is the tool of choice for high-productivity research, development, and analysis.

**3.2.2 CHARACTERISTICS OF MATLAB**

* High level language for scientific and engineering computing.
* Desktop environment tuned for iterative exploration, design and problem solving.
* Graphics for visualizing data and tools for creating plots.
* Apps for curve fitting, data classification, signal analysis and many other specific tasks.
* Add on tool boxes for a wide range of engineering and scientific applications.
* Tools for building applications with custom user interfaces.
* Interfaces to C/C++, Java, .NET, Python, SQL, Hadoop and Microsoft Excel.
* Royalty free deployment options for sharing MATLAB programs with end user.

**CHAPTER 4**

**PROJECT DESIGN**

**4.1 PRE-PROCESSING MODULE**

The first step is preprocessing, which consists of many types of operation such as image registration,scaling,face normalization,reducing the effect of background noise,detection and resizing, all of which affect the face recognition accuracy.

Preprocessing is done by using Non local means filter(NLM) in matlab which is used to reduce the background noise.

**4.1.1 NON-LOCAL MEAN FILTER:**

**Non-local means** is an algorithm in image processing for [image denoising](https://en.wikipedia.org/wiki/Image_denoising). Unlike local mean filters, which take the [mean](https://en.wikipedia.org/wiki/Mean) value of a group of pixels surrounding a target pixel to smooth the image, non-local means filtering takes a mean of all pixels in the image, weighted by how similar these pixels are to the target pixel. This results in much greater post-filtering clarity, and less loss of detail in the image compared with local mean algorithms.

**ALGORITHM:**

**STEP 1**:Read a grayscale image.

I = imread('cameraman.tif');

**STEP 2:**Add white Gaussian noise with zero mean and 0.0015 variance to the image using the [imnoise](https://www.mathworks.com/help/images/ref/imnoise.html) function.

noisyImage = imnoise(I,'gaussian',0,0.0015);

**STEP 3:**Remove noise from the image through non-local means filtering. The [imnlmfilt](https://www.mathworks.com/help/images/ref/imnlmfilt.html) estimates degree of smoothing based on the standard deviation of noise in the image.

[filteredImage,estDoS] = imnlmfilt(noisyImage);

**STEP 4:**Display the noisy image (left) and the non-local means filtered image (right) as a montage. Display estimated degree of smoothing, estDoS as the figure title.

The non-local means filter removes noise from the input image but preserves the sharpness of strong edges, such as the silhouette of the man and buildings. This function also smooths textured regions, such as the grass in the foreground of the image, resulting in less detail when compared to the noisy image.

montage({noisyImage,filteredImage})

title (['Estimated degree of smoothing, ', 'estDoS = ',num2str(estDoS)]);

**4.2 FEATURE EXTRACTION**

Facial feature extraction is the process of extracting face component features like eyes, nose, mouth, etc from human face image. Facial feature extraction is very much important for the initialization of processing techniques like face tracking, facial expression recognition or face recognition.

Feature extraction is done by using LINEAR DISCRIMINANT ANALYSIS(LDA) and FISHERFACE CORE algorithm in matlab.

**4.2.1 LINEAR DISCRIMINANT ANALYSIS:**

Linear Discriminate Analysis (LDA). LDA is a method to find a linear combination of features which characterize or separate two or more classes of objects or events. The resulting combination may be used as a linear classifier. In computerized face recognition, each face is represented by a large number of pixel values. Linear discriminant analysis is primarily used here to reduce the number of features to a more manageable number before classification. Each of the new dimensions is a linear combination of pixel values which form a template.

**ALGORITHM:**

* Compute the class scatters using complete feature samples.
* Calculate the Total class scatter matrix.
* Compare eigen values and eigen vectors.
* Compute the eigen vectors.
* Evaluate the contribution of each feature vector.

**4.2.2 FISHERFACE CORE:**

Fisherface is one of the popular algorithms used in face recognition, and is widely believed to be superior to other techniques, such as eigenface because of the effort to maximize the separation between classes in the training process. The algorithm used in the process for image recognition is fisherfaces algorithm while for identification or matching face image using minimum euclidean.

**ALGORITHM:**

**STEP 1:** Construct the Imagematrix X with each column representing an image. Each image is a assigned to a class in the corresponding class vector C.

**STEP 2:**Project X into the (N-c)-dimensional subspace as P with the rotation matrix WPca identified by a Principal Component Analysis, where

* + N is the number of samples in X
  + c is unique number of classes (length(unique(C)))

**STEP 3:**Calculate the between-classes scatter of the projection P as Sb = \sum\_{i=1}^{c} N\_i\*(mean\_i - mean)\*(mean\_i - mean)^T, where

* + mean is the total mean of P
  + mean\_i is the mean of class i in P
  + N\_i is the number of samples for class i

**STEP 4**:Calculate the within-classes scatter of P as Sw = \sum\_{i=1}^{c} \sum\_{x\_k \in X\_i} (x\_k - mean\_i) \* (x\_k - mean\_i)^T, where

* + X\_i are the samples of class i
  + x\_k is a sample of X\_i
  + mean\_i is the mean of class i in P

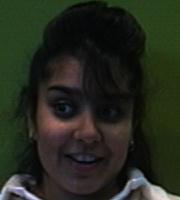
**STEP 5:**Apply a standard Linear Discriminant Analysis and maximize the ratio of the determinant of between-class scatter and within-class scatter. The solution is given by the set of generalized eigenvectors Wfld of Sb and Sw corresponding to their eigenvalue. The rank of Sb is atmost (c-1), so there are only (c-1) non-zero eigenvalues, cut off the rest.

**STEP 6**:Finally obtain the Fisherfaces by W = WPca \* Wfld.

**4.3 DATABASE:**

**4.3.1 TEST IMAGES:**









**4.3.2 TRAINED IMAGES:**





















**CHAPTER 5**

**5.1 IMPLEMENTATION AND RESULTS**

Implementation is the most crucial stage in achieving a successful system and giving the users confidence that new system is effective and workable. Implementation of this project refers to the installation of the packages in its real environment to the full satisfaction of the users and operations of the system. Testing is done individually at the time of development using the data and verification is done the way specified in the program specification. Implementation constitutes all activities that are required to put an already tested and completed package into operation. The success of any information system lies in its successful implementation. System implementation is the stage in the project where the theoretical design is turned into a working system. The most critical stage is achieving a successful system and in giving confidence on the new system for the user that it will work efficiently and effectively. The existing system was long time process.

**5.1.1 nlm.m**

1. close all;
2. clc;
3. clear all;
4. % Reading image as input
5. I = imread('lena.png');
6. Img = rgb2gray(I);
7. [m,n] = size(Img);
9. f=2;
10. %neighborhood window size = 2f+1 , ie 5x5
11. t=5;
12. % similarity window size = 2t+1 , ie 11x11
14. % Making gaussian kernel
15. su=1;          %standard deviation of gaussian kernel
16. sm=0;          % sum of all kernel elements (for normalization)
17. ks= 2\*f+1;     % size of kernel (same as neighborhood window size)
18. % Initiating kernel
19. ker = zeros(ks,ks);
20. for x=1:ks
21. for y=1:ks
22. ab = x-f-1;   %horizontal distance of pixel from center(f+1, f+1)
23. cd = y-f-1;   % vertical distance of pixel from center (f+1, f+1)
24. ker(x,y) = 100\*exp(((ab\*ab)+(cd\*cd))/(-2\*(su\*su)));
25. sm = sm + ker(x,y);
26. end
27. end
28. kernel = ker ./ f;
29. kernel = kernel / sm;   % normalization
31. % adding noise into image
32. noisex = imnoise(Img,'gaussian',0,0.002);
33. noisy = double(noisex);
35. % Assign a clear output image
36. cleared = zeros(m,n);
38. %Degree of filtering
39. h=10;
40. % Replicate boundaries of noisy image
41. noisy2 = padarray(noisy,[f,f],'symmetric');
43. % Now we'll calculate ouput for each pixel
44. for i=1:m
45. for j=1:n
46. im = i+f;   % to compensate for shift due to padarray function
47. jn= j+f;
48. % neighborhood of concerned pixel (we called it similarity window)
49. W1 = noisy2(im-f:im+f , jn-f:jn+f);
50. % BOundaries of similarity window for that pixel
51. rmin = max(im-t, f+1);
52. rmax = min(im+t, m+f);
53. smin = max(jn-t, f+1);
54. smax = min(jn+t, n+f);
55. % Calculate weighted average next
56. NL=0;    % same as cleared (i,j) but for simplicity
57. Z =0;    % sum of all s(i,j)
58. % Run loop through all the pixels in similarity window
59. for r=rmin:rmax
60. for s=smin:smax
61. % neighborhood of pixel 'j' being compared for similarity
62. W2 = noisy2(r-f:r+f, s-f:s+f);
63. % square of weighted euclidian distances
64. d2 = sum(sum(kernel.\*(W1-W2).\*(W1-W2)));
65. % weight of similarity between both pixels : s(i,j)
66. sij = exp(-d2/(h\*h));
67. % update Z and NL
68. Z = Z + sij;
69. NL = NL + (sij\*noisy2(r,s));
70. end
71. end
72. % normalization of NL
73. cleared(i,j) = NL/Z;
74. end
75. end
76. % convert cleared to uint8
77. cleared = uint8(cleared);
79. % show results
80. figure(1);
81. set(gcf, 'Position', get(0,'ScreenSize'));
82. subplot(1,2,1),imshow(noisex),title('noisy Image'),colormap(gray);
83. subplot(1,2,2),imshow(cleared),title('output of NL means filter'),colormap(gray);

**5.1.2.Fisherfacecore.m**

function [m\_database V\_PCA V\_Fisher ProjectedImages\_Fisher] = FisherfaceCore(T)

% Use Principle Component Analysis (PCA) and Fisher Linear Discriminant (FLD) to determine the most

% discriminating features between images of faces.

%

% Description: This function gets a 2D matrix, containing all training image vectors

% and returns 4 outputs which are extracted from training database.

% Suppose Ti is a training image, which has been reshaped into a 1D vector.

% Also, P is the total number of MxN training images and C is the number of

% classes. At first, centered Ti is mapped onto a (P-C) linear subspace by V\_PCA

% transfer matrix: Zi = V\_PCA \* (Ti - m\_database).

% Then, Zi is converted to Yi by projecting onto a (C-1) linear subspace, so that

% images of the same class (or person) move closer together and images of difference

% classes move further apart: Yi = V\_Fisher' \* Zi = V\_Fisher' \* V\_PCA' \* (Ti - m\_database)

%

% Argument: T - (M\*NxP) A 2D matrix, containing all 1D image vectors.

% All of 1D column vectors have the same length of M\*N

% and 'T' will be a MNxP 2D matrix.

%

% Returns: m\_database - (M\*Nx1) Mean of the training database

% V\_PCA - (M\*Nx(P-C)) Eigen vectors of the covariance matrix of the

% training database

% V\_Fisher - ((P-C)x(C-1)) Largest (C-1) eigen vectors of matrix J = inv(Sw) \* Sb

% ProjectedImages\_Fisher - ((C-1)xP) Training images, which are projected onto Fisher linear space

%

% See also: EIG

% Original version by Amir Hossein Omidvarnia, October 2007

% Email: aomidvar@ece.ut.ac.ir

Class\_number = ( size(T,2) )/2; % Number of classes (or persons)

Class\_population = 2; % Number of images in each class

P = Class\_population \* Class\_number; % Total number of training images

m\_database = mean(T,2);

A = T - repmat(m\_database,1,P);

L = A'\*A; % L is the surrogate of covariance matrix C=A\*A'.

[V D] = eig(L); % Diagonal elements of D are the eigenvalues for both L=A'\*A and C=A\*A'.

L\_eig\_vec = [];

for i = 1 : P-Class\_number

L\_eig\_vec = [L\_eig\_vec V(:,i)];

end

V\_PCA = A \* L\_eig\_vec; % A: centered image vectors

Zi = V\_PCA' \* (Ti-m\_database)

ProjectedImages\_PCA = [];

for i = 1 : P

temp = V\_PCA'\*A(:,i);

ProjectedImages\_PCA = [ProjectedImages\_PCA temp];

end

m\_PCA = mean(ProjectedImages\_PCA,2); % Total mean in eigenspace

m = zeros(P-Class\_number,Class\_number);

Sw = zeros(P-Class\_number,P-Class\_number); % Initialization os Within Scatter Matrix

Sb = zeros(P-Class\_number,P-Class\_number); % Initialization of Between Scatter Matrix

for i = 1 : Class\_number

m(:,i) = mean( ( ProjectedImages\_PCA(:,((i-1)\*Class\_population+1):i\*Class\_population) ), 2 )';

S = zeros(P-Class\_number,P-Class\_number);

for j = ( (i-1)\*Class\_population+1 ) : ( i\*Class\_population )

S = S + (ProjectedImages\_PCA(:,j)-m(:,i))\*(ProjectedImages\_PCA(:,j)-m(:,i))';

end

Sw = Sw + S; % Within Scatter Matrix

Sb = Sb + (m(:,i)-m\_PCA) \* (m(:,i)-m\_PCA)'; % Between Scatter Matrix

end

[J\_eig\_vec, J\_eig\_val] = eig(Sb,Sw); % Cost function J = inv(Sw) \* Sb

J\_eig\_vec = fliplr(J\_eig\_vec);

for i = 1 : Class\_number-1

V\_Fisher(:,i) = J\_eig\_vec(:,i); % Largest (C-1) eigen vectors of matrix J

end

Yi = V\_Fisher' \* V\_PCA' \* (Ti - m\_database)

for i = 1 : Class\_number\*Class\_population

ProjectedImages\_Fisher(:,i) = V\_Fisher' \* ProjectedImages\_PCA(:,i);

End

**5.1.3 Example.m**

clear all

clc

close all

TrainDatabasePath = uigetdir(strcat(matlabroot,'\work'), 'Select training database path' );

TestDatabasePath = uigetdir(strcat(matlabroot,'\work'), 'Select test database path');

prompt = {'Enter test image name (a number between 1 to 10):'};

dlg\_title = 'Input of FLD-Based Face Recognition System';

num\_lines= 1;

def = {'1'};

TestImage = inputdlg(prompt,dlg\_title,num\_lines,def);

TestImage = strcat(TestDatabasePath,'\',char(TestImage),'.jpg');

im = imread(TestImage);

T = CreateDatabase(TrainDatabasePath);

[m, V\_PCA, V\_Fisher, ProjectedImages\_Fisher] = FisherfaceCore(T);

OutputName = Recognition(TestImage, m, V\_PCA, V\_Fisher, ProjectedImages\_Fisher);

SelectedImage = strcat(TrainDatabasePath,'\',OutputName);

SelectedImage = imread(SelectedImage);

imshow(im)

title('Test Image');

figure,imshow(SelectedImage);

title('Equivalent Image');

str = strcat('Matched image is :',OutputName);disp(str

**CHAPTER 6**

**CONCLUSION**

Linear Discriminant Analysis(LDA) is supervised learning technique that relies on class labels and is well suited for distributed classes in small datasets.Different distance measures or classifiers may be used for finding the distance between trainee image and database images such as Euclidean distance. Rather than using a single distance classifier for finding the distance between images, some combination of the standard distance measures might outperform the individual distance measures.The simplest mechanism for combining distance measures is to add them. Thus an Enhanced framework for human face recognition using LDA is Implemented.Human faces with variability of facial expressions can be easily recognized.Thus it provides a higher accuracy and reduce the computation time.