

FACE RECOGNITION USING LAPLACIAN FACES

Submitted in partial fulfillment of the
requirements for the award of
Bachelor of Engineering Degree in Computer Science and Engineering

by

SAURAV YADAV (38110514)
SUNIL KUMAR (38110575)



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
JEPPIAAR NAGAR, RAJIV GANDHI SALAI
CHENNAI – 600119, TAMILNADU

MARCH 2022



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)



(Established under Section 3 of UGC Act, 1956)
Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai -600119
www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **SAURAV YADAV (38110514)** and **SUNIL KUMAR (38110575)** who carried out the project entitled “**FACE RECOGNITION USING LAPLACIAN FACES**” under my supervision from November 2021 to March 2022.

Internal Guide
Dr. S. Murugan M.E., Ph.D.,

Head of the Department
Dr. S. Vigneshwari., M.E., Ph.D.,

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

We **SAURAV YADAV (38110514)** and **SUNIL KUMAR (38110575)** hereby declare that the Project Report entitled "**FACE RECOGNITION USING LAPLACIANFACES**" done by me under the guidance of **Dr. S. Murugan M.E.,Ph.D.**, at Sathyabama Institute of Science and Technology is submitted in partial fulfilment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE:

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. We are grateful to them.

I convey my thanks to **Dr. T. SASIKALA, M.E., Ph.D., Dean, School of Computing** and to **Dr. S. VIGNESHWARI, M.E., Ph.D., and Dr. L. LAKSHMANAN, M.E., Ph.D., Heads of the Department, Department of Computer Science And Engineering** for providing the necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to our Project Guide **Dr. S. Murugan M.E., Ph.D.,** for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of Department of **COMPUTER SCIENCE AND ENGINEERING** who were helpful in many ways for the completion of the project.

ABSTRACT

We propose an facial affirmation system called Lapland affirmation. Using the location Defense projections , face images assigned to subspace of faces to examine. It is not exactly equivalent to the main component The evaluation PCA & the discriminant linear analysis LDA that observe only the structure of facial space. This finds entering the information from the blocked neighborhood and gaining a face subspace that better recognizes the complex structure of the main face. The Laplacian faces are the perfect approximations directed to functions of manager of Laplace in the facial complex. In therefore, the annoying assortments that arise due to changes in lighting, external appearance and posture can be discarded or reduced.

Speculative examination shows that PCA as well as LDA along with LPP can be derived from various models of the graphic. Let's consider proposal Focus laplacian face along with the Eigen face and also Fisher face systems in three assorted educational assortments. The test results suggest that Laplacian face proposed approach offers unmatched representation and gets less failure rates of face affirmation.

TABLE OF CONTENTS

ABSTRACT	v
LIST OF FIGURES	viii

CHAPTER No.	TITLE	PAGE No.
1.	INTRODUCTION	1
	1.1 PROBLEM DEFINITION	2
	1.2 SYSTEM ENVIRONMENT	2
2.	SYSTEM ARCHITECTURE	4
	2.1 EXISTING SYSTEM	4
	2.2 PROPOSED SYSTEM	4
	2.3 FUNCTIONAL REQUIREMENT	6
	2.4 SYSTEM ANALYSIS METHODS	6
3.	SYSTEM DESIGN	9
	3.1 FORM DESIGN	9
	3.1.1 MAIN FORM	9
	3.1.2 VIEW DATABASE FORM	9
	3.1.3 RECOGNITION FORM	9
	3.2 INPUT DESIGN	10
	3.3 MENU DESIGN	10
	3.7 DATABASE DESIGN	10
4.	SOFTWARE DEVELOPMENT METHODOLOGY	11
	4.1 SOFTWARE DEVELOPMENT	11
	4.2 SOFTWARE TESTING	11
	4.3 TEMPLATE / STATISTICAL / NEURAL NETWORKS	14
	4.3.1 PRINCIPAL COMPONENT ANALYSIS	14

4.3.2	DISCRETE COSINE TRANSFORM	14
4.3.3	LINEAR DISCRIMINANT ANALYSIS	15
4.3.4	LOCALITY PRESERVING PROJECTION	15
4.3.5	GABOR WAVELET	15
4.4	NEURAL NETWORKS	16
5.	RESULTS AND DISCUSSION	32
6.	CONCLUSION AND FUTURE WORK	36
	REFERENCES	38
	APPENDIX	
	A. PAPER ACCEPTANCE MAIL	40
	B. PLAGIARISM REPORT	41
	C. JOURNAL PAPER	42
	D. SOURCE CODE	45

LIST OF FIGURES

FIGURE No.	FIGURE NAME	PAGE No.
2.1	ARCHITECTURE DIAGRAM	8
5.1	COMPARISION DIAGRAM	35

CHAPTER 1

INTRODUCTION

A smart environment is one that is able to identify people, interpret their actions, and react appropriately. Thus, one of the most important building blocks of smart environments is a person identification system. Face recognition devices are ideal for such systems, since they have recently become fast, cheap, unobtrusive, and, when combined with voice-recognition, are very robust against changes in the environment. Moreover, since humans primarily recognize each other by their faces and voices, they feel comfortable interacting with an environment that does the same.

Facial recognition systems are built on computer programs that analyze images of human faces for the purpose of identifying them. The programs take a facial image, measure characteristics such as the distance between the eyes, the length of the nose, and the angle of the jaw, and create a unique file called a "template." Using templates, the software then compares that image with another image and produces a score that measures how similar the images are to each other. Typical sources of images for use in facial recognition include video camera signals and pre-existing photos such as those in driver's license databases.

Facial recognition systems are computer based security systems that are able to automatically detect and identify human faces. These systems depend on a recognition algorithm, such as eigenface or the hidden Markov model. The first step for a facial recognition system is to recognize a human face and extract it from the rest of the scene. Next, the system measures nodal points on the face, such as the distance between the eyes, the shape of the cheekbones and other distinguishable features. These nodal points are then compared to the nodal points computed from a database of pictures in order to find a match. Obviously, such a system is limited based on the angle of the face captured and the lighting conditions present. New technologies are currently in development to create three-dimensional models of a person's face based on a digital photograph in order to create more nodal points for comparison. However, such technology is inherently susceptible to error given that the computer is extrapolating a three-dimensional model from a two-dimensional photograph.

Principle Component Analysis is an eigenvector method designed to model linear variation in high-dimensional data. PCA performs dimensionality reduction by projecting the original n -dimensional data onto the $k \ll n$ -dimensional linear subspace spanned by the leading

eigen vectors of the data's covariance matrix. Its goal is to find a set of mutually orthogonal basis functions that capture the directions of maximum variance in the data and for which the coefficients are pairwise decorrelated. For linearly embedded manifolds, PCA is guaranteed to discover the dimensionality of the manifold and produces a compact representation.

Facial Recognition Applications

Facial recognition is deployed in large-scale citizen identification applications, surveillance applications, law enforcement applications such as booking stations, and kiosks

1.1 Problem Definition

Facial recognition systems are computer-based security systems that are able to automatically detect and identify human faces. These systems depend on a recognition algorithm. But the most of the algorithm considers some what global data patterns while recognition process. This will not yield accurate recognition system.

So we propose a face recognition system which can able to recognition with maximum accuracy as possible.

1.2 System Environment

The front end is designed and executed with the J2SDK1.4.0 handling the core java part with User interface Swing component. Java is robust , object oriented , multi-threaded , distributed , secure and platform independent language. It has wide variety of package to implement our requirement and number of classes and methods can be utilized for programming purpose. These features make the programmer's to implement to require concept and algorithm very easier way in Java.

The features of Java are as follows:

Core java contains the concepts like Exception handling, Multithreading, Streams can be well utilized in the project environment. The Exception handling can be done with predefined exception and has provision for writing custom exception for our application. Garbage collection is done automatically, so that it is very secure in memory management. The user interface can be done with the Abstract Window tool Kit and also Swing class. This has variety of classes for components and containers. We can make instance of these classes and this instances denotes particular object that can be utilized in our program.

Event handling can be performed with Delegate Event model. The objects are assigned to the Listener that observe for event, when the event takes place the corresponding methods to handle that event will be called.

This application make use of ActionListener interface and the event click event gets handled by this. The separate method actionPerformed() method contains details about the response of event.

System Requirements

Hardware specifications

Processor	:	Intel Processor IV
RAM	:	128 MB
Hard disk	:	20 GB
CD drive	:	40 x Samsung
Floppy drive	:	1.44 MB
Monitor	:	15' Samtron color
Keyboard	:	108 mercury keyboard

Software Specification

Operating System	:	Windows XP/2000
Language used	:	J2sdk1.

CHAPTER 2

SYSTEM ARCHITECTURE

2.1 Existing System

Facial recognition systems are computer-based security systems that are able to automatically detect and identify human faces. These systems depend on a recognition algorithm. Principal Component Analysis (PCA) is a statistical method under the broad title of factor analysis. The purpose of PCA is to reduce the large dimensionality of the data space (observed variables) to the smaller intrinsic dimensionality of feature space (independent variables), which are needed to describe the data economically. This is the case when there is a strong correlation between observed variables. The jobs which PCA can do are prediction, redundancy removal, feature extraction, data compression, etc. Because PCA is a known powerful technique which can do something in the linear domain, applications having linear models are suitable, such as signal processing, image processing, system and control theory, communications, etc.

The main idea of using PCA for face recognition is to express the large 1-D vector of pixels constructed from 2-D face image into the compact principal components of the feature space. This is called eigenspace projection. Eigenspace is calculated by identifying the eigenvectors of the covariance matrix derived from a set of fingerprint images(vectors).

But the most of the algorithm considers some what global data patterns while recognition process. This will not yield accurate recognition system because

- Less accurate
- Does not deal with manifold structure
- It does not deal with biometric characteristics.

2.2 Proposed System

PCA and LDA aim to preserve the global structure. However, in many real-world applications, the local structure is more important. In this section, we describe a Locality Preserving Projection (LPP) [9], a new algorithm for learning a locality preserving subspace. The complete derivation and theoretical justifications of LPP can be traced back to [9]. LPP seeks to preserve the intrinsic geometry of the data and local structure. The objective function of LPP is as follows.

LPP is a general method for manifold learning. It is obtained by finding the optimal linear approximations to the eigenfunctions of the Laplace Beltrami operator on the manifold [9]. Therefore, though it is still a linear technique, it seems to recover important aspects of the intrinsic nonlinear manifold structure by preserving local structure. Based on LPP, we describe our Laplacianfaces method for face representation in a locality preserving subspace. In the face analysis and recognition problem, one is confronted with the difficulty that the matrix $XDXT$ is sometimes singular.

This stems from the fact that sometimes the number of images in the training set n is much smaller than the number of pixels in each image m . In such a case, the rank of $XDXT$ is at most n , while $XDXT$ is an $m \times m$ matrix,

which implies that $XDXT$ is singular. To overcome the complication of a singular $XDXT$, we first project the image set to a PCA subspace so that the resulting matrix $XDXT$ is nonsingular. Another consideration of using PCA as preprocessing is for noise reduction.

This method, we call Laplacianfaces, can learn an optimal subspace for face representation and recognition. The algorithmic procedure of Laplacianfaces is formally stated below:

1. **PCA projection** : We project the image set x into the PCA subspace by throwing away the smallest principal components. In our experiments, we kept 98 percent information in the sense of reconstruction error. For the sake of simplicity, we still use x to denote the images in the PCA subspace in the following steps. We denote by W_{PCA} the transformation matrix of PCA.

2. Constructing the nearest-neighbor graph. Let G denote a graph with n nodes. The i th node corresponds to the face image x_i . We put an edge between nodes i and j if x_i and x_j are "close," i.e., x_j is among k nearest neighbors of x_i , or x_i is among k nearest neighbors of x_j . The constructed nearest neighbor graph is an approximation of the local manifold structure. Note that here we do not use the ϵ -neighborhood to construct the graph. This is simply because it is often difficult to choose the optimal ϵ in the real-world applications, while k nearest-neighbour graph can be constructed more stably. The disadvantage is that the k

nearest-neighbor search will increase the computational complexity of our algorithm. When the computational complexity is a major concern, one can switch to the k -neighborhood.

3. Choosing the weights. If node i and j are connected, put

column (or row, since S is symmetric) sums of S is the Laplacian matrix. The i th row of matrix X is x_i . be the solutions of (35), ordered according to their eigenvalues.

These eigenvalues are equal to or greater than zero because the matrices $XLXT$ and $XDXT$ are both symmetric and positive semidefinite. Thus, the embedding is as follows:

where y is a k -dimensional vector. W is the transformation matrix. This linear mapping best preserves the manifold's estimated intrinsic geometry in a linear sense. The column vectors of W are the so-called Laplacian faces.

This principle is implemented with unsupervised learning concept with training and test data.

2.3 Functional Requirement:

The system must require to implement Principle Component Analysis to reduce image in the dimension less than n and co-variance of the data. The system must be used in Unsupervised learning algorithm. So it must be trained properly with relevant data sets.

Based on this training input data is tested by the application and result is displayed to the user.

2.4 System Analysis Methods

System analysis can be defined, as a method that is determined to use the resources, machine in the best manner and perform tasks to meet the information needs of an organization. It is also a management technique that helps us in designing a new systems or improving an existing system. The four basic elements in the system analysis are

- Output
- Input
- Files
- Process

The above-mentioned are mentioned are the four basis of the System Analysis.

Feasibility Study

Feasibility is the study of whether or not the project is worth doing. The process that follows this determination is called a Feasibility Study. This study is taken in right time constraints and normally culminates in a written and oral feasibility report. This feasibility study is

categorized into seven different types. They are

- Technical Analysis
- Economical Analysis
- Performance Analysis
- Control and Security Analysis
- Efficiency Analysis
- Service Analysis

Technical Analysis

This analysis is concerned with specifying the software that will successfully satisfy the user requirements. The technical needs of a system are to have the facility to produce the outputs in a given time and the response time under certain conditions..

Economic Analysis

Economic Analysis is the most frequently used technique for evaluating the effectiveness of prepared system. This is called Cost/Benefit analysis. It is used to determine the benefits and savings that are expected from a proposed system and compare them with costs. If the benefits overweigh the cost, then the decision is taken to the design phase and implements the system.

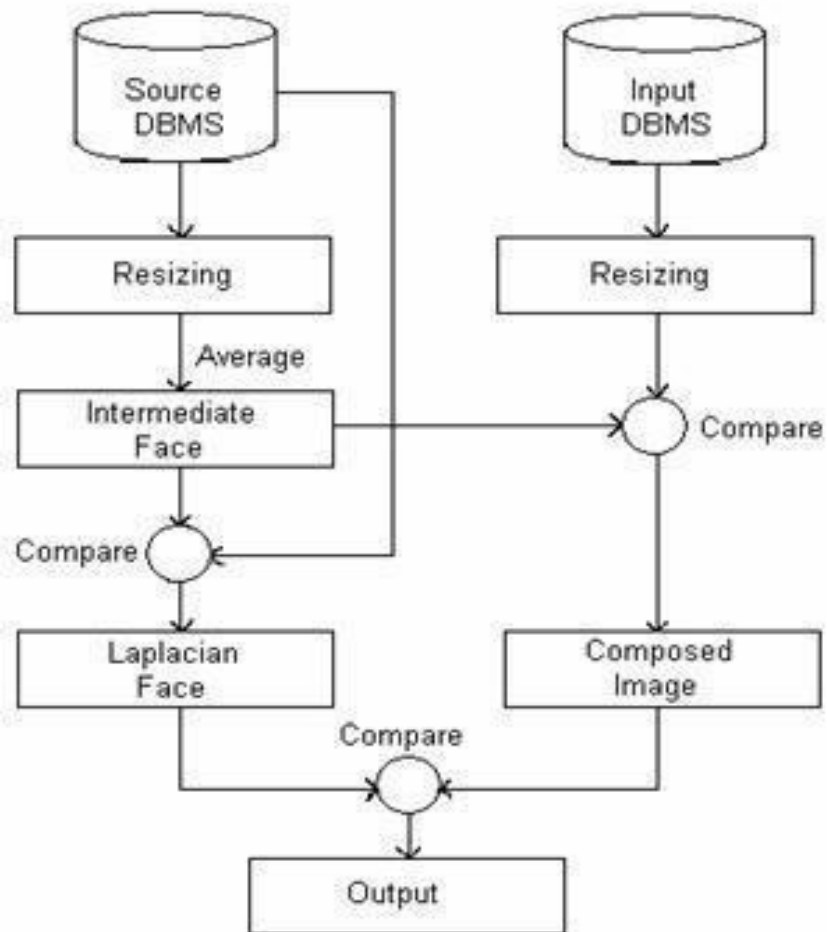
Performance Analysis

The analysis on the performance of a system is also a very important analysis. This analysis analyses about the performance of the system both before and after the proposed system. If the analysis proves to be satisfying from the company's side then this analysis result is moved to the next analysis phase. Performance analysis is nothing but invoking at program execution to pinpoint where bottle necks or other performance problems such as memory leaks might occur. If the problem is spotted out then it can be rectified.

Efficiency Analysis

This analysis mainly deals with the efficiency of the system based on this project. The resources required by the program to perform a particular function are analyzed in this phase. It is also checks how efficient the project is on the system, in spite of any changes in the system. The efficiency of the system should be analyzed in such a way that the user

should not feel any difference in the way of working. Besides, it should be taken into consideration that the project on the system should last for a longer time.



Architecture Diagram

Fig 2.1

CHAPTER 3

SYSTEM DESIGN

Design is concerned with identifying software components specifying relationships among components. Specifying software structure and providing blue print for the document phase. Modularity is one of the desirable properties of large systems. It implies that the system is divided into several parts. In such a manner, the interaction between parts is minimal clearly specified.

Design will explain software components in detail. This will help the implementation of the system. Moreover, this will guide the further changes in the system to satisfy the future requirements.

3.1 Form design

Form is a tool with a message; it is the physical carrier of data or information. It also can constitute authority for actions. In the form design files are used to do each module. The following are list of forms used in this project:

3.1.1 Main Form

Contains option for viewing face from data base. The system retrieves the images stored in the folder called train and test folder ,which is available in bin folder of your application.

3.1.2 View database form

This form retrieves face available in the train folder.It is just for viewing purpose for the user.

3.1.3 Recognition form

This form provides option for loading input image from test folder. Then user has to click Train button which leads the application for training to gain knowledge as it is of the form unsupervised learning algorithm.

Unsupervised learning - this is learning from observation and discovery. The data mining system is supplied with objects but no classes are defined so it has to observe the examples and recognize patterns (i.e. class description) by itself. This system results in a set of class descriptions, one for each class discovered in the environment. Again this is similar to cluster analysis as in statistics.

Then user can click the test button Test button to see the matching for the faces. The matched face will be displayed in the place provided for matched face option. In case of any difference the information will be displayed in place provided in the form.

3.2 Input design

Inaccurate input data is the most common case of errors in data processing. Errors entered by data entry operators can control by input design. Input design is the process of converting user-originated inputs to a computer-based format. Input data are collected and organized into group of similar data.

3.3 Menu Design

The menu in this application is organized into mdiform that organizes viewing of image files from folder. Also it has option for loading image as input , try to perform training method and test whether it recognizes the face or not.

3.4 Database design

A database is a collection of related data. The database has following properties:

Database reflects the changes of the information.A database is logically coherent collection of data with some inherent meaning.This application takes the images form the default folder set for this application train and test folders. The file extension is.jpeg option.

CHAPTER 4

SOFTWARE DEVELOPMENT METHODOLOGY

4.1 Software Development

This software is developed to implement Principle component analysis. Image manipulation in this module designed to view all the faces that are considered in our training case. Principle Component Analysis is an eigenvector method designed to model linear variation in high-dimensional data. PCA performs dimensionality reduction by projecting the original n -dimensional data onto the $k \ll n$ -dimensional linear subspace spanned by the leading eigenvectors of the data's covariance matrix. Its goal is to find a set of mutually orthogonal basis functions that capture the directions of maximum variance in the data and for which the coefficients are pair wise decorrelated. For linearly embedded manifolds, PCA is guaranteed to discover the dimensionality of the manifold and produces a compact representation.

Training module : Unsupervised learning - this is learning from observation and discovery. The data mining system is supplied with objects but no classes are defined so it has to observe the examples and recognize patterns (i.e. class description) by itself. This process requires training data set .This system provides training set has 17 faces and each contains three different poses of faces. It undergoes iterative process stores require detail in face Template two dimension array.

Test module : After training process is over , it process the input image face for eigen face process then can able to say whether it recognizes or not.

4.2 Software Testing

Software Testing is the process of confirming the functionality and correctness of software by running it. Software testing is usually performed for one of two reasons:

1. Defect detection
2. Reliability estimation.

White box testing is concerned only with testing the software product, it cannot guarantee that the complete specification has been implemented. Black box testing is concerned only with testing the specification, it cannot guarantee that all parts of the implementation have

been tested. Thus black box testing is testing against the specification and will discover faults of omission, indicating that part of the specification has not been fulfilled. White box testing is testing against the implementation and will discover faults of commission, indicating that part of the implementation is faulty. In order to fully test a software product both black and white box testing are required.

The problem of applying software testing to defect detection is that software can only suggest the presence of flaws, not their absence (unless the testing is exhaustive). The problem of applying software testing to reliability estimation is that the input distribution used for selecting test cases may be flawed. In both of these cases, the mechanism used to determine whether program output is correct is often impossible to develop. Obviously the benefit of the entire software testing process is highly dependent on many different pieces. If any of these parts is faulty, the entire process is compromised.

Software is now unique unlike other physical processes where inputs are received and outputs are produced. Where software differs is in the manner in which it fails. Most physical systems fail in a fixed (and reasonably small) set of ways. By contrast, software can fail in many bizarre ways. Detecting all of the different failure modes for software is generally infeasible.

The key to software testing is trying to find the myriad of failure modes – something that requires exhaustively testing the code on all possible inputs. For most programs, this is computationally infeasible. It is commonplace to attempt to test as many of the syntactic features of the code as possible (within some set of resource constraints) are called white box software testing technique. Techniques that do not consider the code's structure when test cases are selected are called black box technique.

Functional testing is a testing process that is black box in nature. It is aimed at examine the overall functionality of the product. It usually includes testing of all the interfaces and should therefore involve the clients in the process.

Final stage of the testing process should be System Testing. This type of test involves examination of the whole computer system, all the software components, all the hard ware components and any interfaces.

The whole computer based system is checked not only for validity but also to meet the objectives.

Implementation includes all those activities that take place to convert from the old system to the new. The new system may be totally new, replacing an existing system or it may be major modification to the system currently pu1t2into use. This system "Face Recognition" is a

new system. Implementation as a whole involves all those tasks that we do for successfully replacing the existing or introduce new software to satisfy the requirement.

The entire work can be described as retrieval of faces from database, processed for eigen faces training method and test case are executed and finally result is displayed to the user. The test case has performed in all aspect and the system has given correct result in all the cases.

Face Recognition Methods:-

There are different methods for face recognition, which are as follows-

1. Geometric Based / TemplateBased:-

Face recognition algorithms classified as geometry based or template based algorithms. The template-based methods can be constructed using statistical tools like SVM [Support Vector Machines], PCA [Principal Component Analysis], LDA [Linear Discriminant Analysis], Kernel methods or Trace Transforms. The geometric feature based methods analyse local facial features and their geometric relationship. It is also known as a feature-based method.

2. Piecemeal / Wholistic:-

The relation between the elements or the connection of a function with the whole face not undergone into the amount, many researchers followed this approach, trying to deduce the most relevant characteristics. Some methods attempted to use the eyes, a combination of features and so on. Some Hidden Markov Model methods also fall into this category, and feature processing is very famous in face recognition.

3. Appearance-Based / Model-Based:-

The appearance-based method shows a face regarding several images. An image considered as a high dimensional vector. This technique is usually used to derive a feature space from the image division. The sample image compared to the training set. On the other

hand, the model-based approach tries to model a face. The new sample implemented to the model and the parameters of the model used to recognise the image. The appearance-based method can classify as linear or nonlinear. Ex- PCA, LDA, IDA used in direct approach whereas Kernel PCA used in nonlinear approach. On the other hand, in the model-based method can be classified as 2D or 3D Ex- Elastic Bunch Graph Matching used.

4.3 Template / Statistical / Neural Networks Based:

Template Matching:-

In template matching the patterns are represented by samples, models, pixels, textures, etc. The recognition function is usually a correlation or distance measure.

Statistical Approach:-

In the Statistical approach, the patterns expressed as features. The recognition function in a discriminant function. Each image represented regarding d features. Therefore, the goal is to choose and apply the right statistical tool for extraction and analysis.

There are many statistical tools, which used for face recognition. These analytical tools used in a two or more groups or classification methods. These tools are as follows-

4.3.1. Principal Component Analysis[PCA]:-

One of the most used and cited statistical method is the Principal Component Analysis. A mathematical procedure performs a dimensionality reduction by extracting the principal component of multi-dimensional data. Dimension reduction from 3D to 2D image

4.3.2. Discrete Cosine Transform[DCT]:-

It signifies a series of data points regarding a sum of cosine functions different oscillating

frequencies. The Discrete Cosine Transform is based on Fourier discrete transform and therefore, by compacting the variations it can be used to transform images and allowing an efficient dimensionality reduction.

4.3.3. Linear Discriminant Analysis[LDA]:-

LDA is widely used to find the linear combination of features while preserving class separability. Unlike PCA, the LDA tries to model to the difference between levels. For each level the LDA obtains differenced in multiple projection vectors.

4.3.4. Locality Preserving Projections[LPP]:-

HE and NIYOGI introduced The LPP. It is the best alternative of PCA for preserve locality structure and designing. Pattern recognition algorithms usually search for the nearest pattern or neighbours. Therefore, the locality maintaining the quality of LLP can quicken the recognition.

4.3.5. GaborWavelet:-

In this algorithm, it signifies that Neuro-physiological data evidence from the visual cortex of mammalian brains suggests that simple cells in the visual cortex can view as a family of self-similar 2D Gabor wavelets. The Gabor functions proposed by Daugman are local spatial bandpass filters that achieve the theoretical limit for conjoint resolution of information in the 2D spatial and 2D Fourier domains.

Independent Component Analysis [ICA]:-

ICA aims to transform the data as linear combinations of the statistically independent data point. Therefore, its goal is to provide an independent instead that uncorrelated image representation. ICA is an alternative to PCA, which give a more powerful data representation. It is a discriminant analysis criterion, which can be used to enhance PCA.

Kernel PCA:-

Scholkopf et al. introduced the use of Kernel functions for performing nonlinear PCA. Its basic methodology is to apply a nonlinear mapping to the input and then solve a linear PCA in the resulting feature subspace.

4.4. Neural Networks:-

Neural Network has continued to use pattern recognition and classification. Kohonen was the first to show that a neuron network could be used to recognise aligned and normalised faces. There are methods, which perform feature extraction using neural networks. There are many methods, which combined with tools like PCA or LCA and make a hybrid classifier for face recognition. These are like Feed Forward Neural Network with additional bias, Self-Organizing Maps with PCA, and Convolutional Neural Networks with multi-layer perception, etc. These can increase the efficiency of the models.

Neural Networks with Gabor Filters:-

The algorithm achieves face recognition by implementing a multilayer perceptron with a back-propagation algorithm. Firstly, there is a preprocessing step. Each image normalised in phases of contrast and illumination. Then each image is processed through a Gabor filter. The Gabor filter has five orientation parameters and three spatial frequencies, so there are 15 Gabor wavelengths.

Neural Networks and Hidden Markov Models:-

Hidden Markov Models are a statistical tool used in face recognition. They have used in conjunction with neural networks. It generated in a neural network that trains pseudo 2D HMM. The input of this 2D HMM process is the output of the ANN, and It provides the algorithm with the proper dimensionality reduction.

Fuzzy Neural Networks:-

The fuzzy neural networks for face recognition introduce in 2009. In this a face recognition system using a multilayer perceptron. The concept behind this approach is to capture decision surfaces in nonlinear manifolds a task that a simple MLP can hardly complete. The feature vectors are obtained using Gabor wavelength transforms.

PCA

Face recognition is ubiquitous in science fiction: the protagonist looks at a camera, and the camera scans his or her face to recognize the person. More formally, we can formulate facerecognition as a classification task, where the inputs are images and the outputs are people's names. We're going to discuss a popular technique for face recognition called eigenfaces. And at the heart of eigenfaces is an unsupervised dimensionality reduction technique called principal component analysis (PCA), and we will see how we can apply this general technique to our specific task of face recognition.

Before discussing principal component analysis, we should first define our problem. **Face recognition** is the challenge of classifying whose face is in an input image. This is different than **face detection** where the challenge is determining if there is a face in the input image. With face recognition, we need an existing database of faces. Given a new image of a face, we need to report the person's name. A naïve way of accomplishing this is to take the new image, flatten it into a vector, and compute the Euclidean distance between it and all of the other flattened images in our database.

There are several downsides to this approach. First of all, if we have a large database of faces, then doing this comparison for each face will take a while! Imagine that we're building a face recognition system for real-time use! The larger our dataset, the slower our algorithm. But more faces will also produce better results! We want a system that is both fast and accurate. For this, we'll use a neural network! We can train our network on our dataset and use it for our face recognition task.

There's an issue with directly using a neural network: images can be large! If we had a single $m \times n$ image, we would have to flatten it out into a single $mn \times 1$ vector to feed into our neural network as input. For large image sizes, this might hurt speed! This is related to the second problem with using images as-is in our naïve approach: they are high-dimensional! (An $m \times n$ image is really a $mn \times 1$ vector) A new input might have a ton of noise and comparing each and every pixel using matrix subtraction and Euclidean distance might give us a high error and classifications!

These issues are why we don't use the naive method. Instead, we'd like to take our high-dimensional images and boil them down to a smaller dimensionality while retaining the *essence* or *important parts* of the image.

Dimensionality Reduction

The previous section motivates our reason for using a dimensionality reduction technique. Dimensionality reduction is a type of unsupervised learning where we want to take higher-dimensional data, like images, and represent them in a lower-dimensional space.

One technique of dimensionality reduction is called principal component analysis (PCA). The idea behind PCA is that we want to select the hyperplane such that when all the points are projected onto it, they are maximally spread out. In other words, we want the *axis of maximal variance*! Let's consider our example plot above. A potential axis is the x-axis or y-axis, but, in both cases, that's not the best axis. However, if we pick a line that cuts through our data diagonally, that is the axis where the data would be most spread. If we were to project our points onto this axis, they would be maximally spread! But how do we figure out this axis? We can borrow a term from linear algebra called eigenvectors! This is where eigenfaces gets its name! Essentially, we compute the covariance matrix of our data and consider that covariance matrix's largest eigenvectors. Those are our principal axes and the axes that we project our data onto to reduce dimensions. Using this approach, we can take high-dimensional data and reduce it down to a lower dimension by selecting the largest eigenvectors of the covariance matrix and projecting onto those eigenvectors.

Since we're computing the axes of maximum spread, we're retaining the most important aspects of our data. It's easier for our classifier to separate faces when our data are spread out as opposed to bunched together.

There are other dimensionality techniques, such as Linear Discriminant Analysis, that use supervised learning and are also used in face recognition, but PCA works really well!

How does this relate to our challenge of face recognition? We can conceptualize our $m \times n$ images as points in-dimensional space. Then, we can use PCA to reduce our space from into something much smaller. This will help speed up our computations and

$$\max_{\mathbf{w}} \sum_{i=1}^n (y_i - \bar{y})^2$$

$$\bar{y} = \frac{1}{n} \sum_{i=1}^n y_i$$

The output set of principal vectors $w_1, w_2, \dots, w_k$ are an orthonormal set of vectors representing the eigenvectors of the sample covariance matrix associated with the $k < d$ largest eigenvalues. While PCA seeks directions that are efficient for representation, Linear Discriminant Analysis seeks directions that are efficient for discrimination. Suppose we have a set of n d -dimensional samples $x_1, x_2, \dots, x_n$, belonging to l classes of faces. The objective function is as follow

$$S_B = \sum_{i=1}^l n_i (\mathbf{m}^{(i)} - \mathbf{m})(\mathbf{m}^{(i)} - \mathbf{m})^T$$

$$S_W = \sum_{i=1}^l \left(\sum_{j=1}^{n_i} (\mathbf{x}_j^{(i)} - \mathbf{m}^{(i)})(\mathbf{x}_j^{(i)} - \mathbf{m}^{(i)})^T \right)$$

where $\bar{\mathbf{x}}$ is the total sample mean vector, n_i is the number of samples in the i th class, $\bar{\mathbf{x}}_i$ is the average vector of the i th class, and $\mathbf{x}_{ij}$ is the j th sample in the i th class. We call $\mathbf{S}_W$ the within-class scatter matrix and $\mathbf{S}_B$ the between-class scatter matrix. PCA and LDA aim to preserve the global structure. However, in many real world applications, the local structure is more important. In this section, we describe Locality Preserving Projection (LPP) [9], a new algorithm for learning a locality preserving subspace. The complete derivation and theoretical justifications of LPP can be traced back to [9]. LPP seeks to preserve the intrinsic geometry of the data and local structure. The objective function of LPP is as follows:

$$\min \sum_{ij} (y_i - y_j)^2 S_{ij}$$

where y_i is the one-dimensional representation of $\mathbf{x}_i$ and the matrix $\mathbf{S}$ is a similarity matrix. A possible way of defining $\mathbf{S}$ is as follows:

$$S_{ij} = \begin{cases} \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / t), & \|\mathbf{x}_i - \mathbf{x}_j\|^2 < \varepsilon \\ 0 & \text{otherwise.} \end{cases}$$

or

$$S_{ij} = \begin{cases} \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / t), & \text{if } \mathbf{x}_i \text{ is among } k \text{ nearest neighbors of } \mathbf{x}_j \\ & \text{or } \mathbf{x}_j \text{ is among } k \text{ nearest neighbors of } \mathbf{x}_i \\ 0 & \text{otherwise.} \end{cases}$$

otherwise. or $\mathbf{x}_i$ is among nearest neighbors of $\mathbf{x}_j$, if $\mathbf{x}_j$ is among nearest neighbors of $\mathbf{x}_i$ where ε is sufficiently small, and $\varepsilon > 0$. Here, ε defines the radius of the local neighborhood. In other words, ε defines the “locality”. The objective function with our choice of symmetric weights S_{ij} ($S_{ij} = S_{ji}$) incurs a heavy penalty if neighboring points $\mathbf{x}_i$ and $\mathbf{x}_j$ are mapped far apart, i.e. if $(y_i - y_j)^2$ is large. Therefore, minimizing it is an attempt to ensure that if $\mathbf{x}_i$ and $\mathbf{x}_j$ are “close” then y_i

and y_j are close as well. Following some simple algebraic steps, we see that

$$\begin{aligned}
& \frac{1}{2} \sum_{ij} (y_i - y_j)^2 S_{ij} \\
&= \frac{1}{2} \sum_{ij} (\mathbf{w}^T \mathbf{x}_i - \mathbf{w}^T \mathbf{x}_j)^2 S_{ij} \\
&= \sum_{ij} \mathbf{w}^T \mathbf{x}_i S_{ij} \mathbf{x}_i^T \mathbf{w} - \sum_{ij} \mathbf{w}^T \mathbf{x}_i S_{ij} \mathbf{x}_j^T \mathbf{w} \\
&= \sum_i \mathbf{w}^T \mathbf{x}_i D_{ii} \mathbf{x}_i^T \mathbf{w} - \mathbf{w}^T X S X^T \mathbf{w} \\
&= \mathbf{w}^T X D X^T \mathbf{w} - \mathbf{w}^T X S X^T \mathbf{w} \\
&= \mathbf{w}^T X (D - S) X^T \mathbf{w} \\
&= \mathbf{w}^T X L X^T \mathbf{w}
\end{aligned}$$

where $X = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n]$, and D is a diagonal matrix; its entries are column (or row, since S is symmetric) sums of S , $D_{ii} = \sum_j S_{ji}$. $L = D - S$ is the Laplacian matrix [6]. Matrix D provides a natural measure on the data points. The bigger the value D_{ii} (corresponding to y_i) is, the more “important” is y_i . Therefore, we impose a constraint as follows:

$$\begin{aligned}
& \mathbf{y}^T D \mathbf{y} = 1 \\
\Rightarrow & \mathbf{w}^T X D X^T \mathbf{w} = 1
\end{aligned}$$

Finally, the minimization problem reduces to finding:

$$\begin{aligned}
& \arg \min_{\mathbf{w}} \mathbf{w}^T X L X^T \mathbf{w} \\
& \mathbf{w}^T X D X^T \mathbf{w} = 1
\end{aligned}$$

The transformation vector $\mathbf{w}$ that minimizes the objective function is given by the minimum eigenvalue solution to the generalized eigenvalue problem.

$$X L X^T \mathbf{w} = \lambda X D X^T \mathbf{w}$$

Note that the two matrices are both symmetric and positive semi-definite, since the Laplacian matrix L and the diagonal matrix D are both symmetric and positive semi-definite. The Laplacian matrix for finite graph is analogous to the Laplace Beltrami operator on compact Riemannian manifolds. While the Laplace Beltrami operator for a manifold is generated by the Riemannian metric, for a graph it comes from the adjacency relation. Belkin and Niyogi [3] showed that the optimal map preserving locality can be found by solving the following optimization problem on the manifold:

$$\min_{\|f\|_{L^2(M)}=1} \int_M \|\nabla f\|^2$$

$$\min_{\|f\|_{L^2(M)}=1} \int_M \mathcal{L}(f)f$$

where L is the Laplace Beltrami operator on the manifold, i.e. $L(f) = -\operatorname{div}\nabla(f)$. Thus, the optimal f has to be an eigenfunction of L . If we assume f to be linear, we have $f(\mathbf{x}) = \mathbf{w}^T \mathbf{x}$. By spectral graph theory, the integral can be discretely approximated by $\mathbf{w}^T X L X^T \mathbf{w}$ and the L^2 norm of f can be discretely approximated by $\mathbf{w}^T X D X^T \mathbf{w}$, which will ultimately lead to the following eigenvalue problem:

$$X L X^T \mathbf{w} = \lambda X D X^T \mathbf{w}$$

The derivation reflects the intrinsic geometric structure of the manifold. For details, see [3][6][9].

LDA

As highly structured two-dimensional patterns, human face images can be analyzed in the spatial and the frequency domains. These patterns are composed of components that are easily recognized at high levels but are loosely defined at low levels of our visual system.^{2,27} Each of the facial components (features) has a different discrimination power for identifying a person or the person's gender, race, and age.

There have been many studies of the significance of such features that used subjective psycho visual experiments. Using objective measures, in this section we propose a computational scheme for evaluating the significance of different facial attributes in terms of their discrimination potential. The results of this analysis can be supported by subjective psycho visual findings. To analyze any rep, where V can be the original image, its spatial segments, or transformed images, we provide the following framework. First, we need a training set composed of a relatively large group of subjects with diverse facial characteristics.

The appropriate selection of the training set directly determines the validity of the final results. The database should contain several examples of face images for each subject in the training set and at least one example in the test set. These examples should represent different frontal views of subjects with minor variations in view angle. They should also include different facial expressions, different lighting and background conditions, and examples with and without glasses. It is assumed that all images are already normalized to $m \times n$ arrays and that they contain only the face regions and not much of the subjects' bodies. Second, for each image and subimage, starting with the two-dimensional $m \times n$ array of intensity values, we construct the lexicographic vector expansion $f \in \mathbb{R}^{m \times n}$. This vector corresponds to the initial representation of the face. Thus the set of all faces in the feature space is treated as a high-dimensional vector space. Third, by defining all instances of the same person's face as being in one class and the faces of different subjects as being in different classes for all subjects in the training set, we establish a framework for performing a cluster separation analysis in the feature space. Also, having labeled all instances in the training set and having defined all the classes, we compute the within- and between-class scatter matrices as follows. Here S_w is the within-class scatter matrix showing the average scatter of the sample vectors (V) of different classes C_i around their respective mean, vectors $\bar{V}_i$. Similarly, S_b is the between-class scatter matrix, representing the scatter of the conditional mean, vectors ($\bar{V}_i$) around the overall mean

vector m . PrC_i is the probability of the i th class. The discriminatory power of a representation can be quantified by using various measures. In this paper we use the separation matrix, which shows the combination of within- and between-class scatters of the feature points in the representation space. The class separation matrix and a measure of separability can be computed as J_V is our measure of the discrimination power (DP) of a given representation V .

As mentioned above, the representation may correspond to the data in its original form (e.g., a gray-scale image), or it can be based on a set of abstract features computed for a specific task. For example, through this analysis we are able to compare the DP's of different spatial segments (components) of a face. We can apply the analysis to segments of the face images such as the areas around the eyes, mouth, hair, and chin or combinations of them. Figure 1 shows a separation analysis for horizontal segments of the face images in the database. The results show that the DP's of all segments are comparable and that the area between the nose and the mouth has more identification information than other parts shows that the DP of the whole image is significantly larger than the DP's of its parts. Using wavelet transforms^{28–30} as multiscale orthogonal representations of face images, we can also perform a comparative analysis of the DP's of sub images in the wavelet domain. Different components of a wavelet decomposition capture different visual aspects of a grayscale image. At each level of decomposition there are four orthogonal sub images corresponding to the smoothed, low-frequency variations sharp changes in the horizontal direction, i.e., vertical edges. • HL: sharp changes in the vertical direction, i.e., horizontal edges. • HH: sharp changes in non horizontal, non vertical directions, i.e., other edges. We applied the LDA to each sub image of the wavelet transform of the face and estimated the DP of each image.

LPP

Many face recognition techniques have been developed over the past few decades. One of the most successful and well-studied techniques to face recognition is the appearance-based method . When using appearance-based methods, we usually represent an image of size $n \times m$ pixels by a vector in an $n \times m$ dimensional space. In practice, however, these $n \times m$ dimensional spaces are too large to allow robust and fast face recognition. A common way to attempt to resolve this problem is to use dimensionality reduction techniques. Two of the most popular techniques for this purpose are Principal Component Analysis (PCA) and Linear Discriminant

Analysis. PCA is an eigenvector method designed to model linear variation in high-dimensional data. PCA performs dimensionality reduction by projecting the original n -dimensional data onto the k -dimensional linear subspace spanned by the leading eigenvectors of the data's covariance matrix. Its goal is to find a set of mutually orthogonal basis functions that capture the directions of maximum variance in the data and for which the coefficients are pairwise decorrelated. For linearly embedded manifolds, PCA is guaranteed to discover the dimensionality of the manifold and produces a compact representation. Turk and Pentland [28] use Principal Component Analysis to describe face images in terms of a set of basis functions, or "eigenfaces".

LDA is a supervised learning algorithm. LDA searches for the project axes on which the data points of different classes are far from each other while requiring data points of the same class to be close to each other. Unlike PCA which encodes information in an orthogonal linear space, LDA encodes discriminating information in a linear separable space using bases are not necessarily orthogonal. It is generally believed that algorithms based on LDA are superior to those based on PCA. However, some recent work shows that, when the training dataset is small, PCA can outperform LDA, and also that PCA is less sensitive to different training datasets. 3 Recently, a number of research efforts have shown that the face images possibly reside on a nonlinear submanifold. However, both PCA and LDA effectively see only the Euclidean structure. They fail to discover the underlying structure, if the face images lie on a nonlinear submanifold hidden in the image space. Some nonlinear techniques have been proposed to discover the nonlinear structure of the manifold, e.g. Isomap, LLE, and Laplacian Eigenmap.

These nonlinear methods do yield impressive results on some benchmark artificial data sets. However, they yield maps that are defined only on the training data points and how to evaluate the maps on novel test data points remains unclear. Therefore, these nonlinear manifold learning techniques might not be suitable for some computer vision tasks, such as face recognition. In the meantime, there has been some interest in the problem of developing low dimensional representations through kernel based techniques for face recognition. These methods can discover the nonlinear structure of the face images. However, they are computationally expensive. Moreover, none of them explicitly considers the structure of the manifold on which the face images possibly reside.

In this paper, we propose a new approach to face analysis (representation and recognition), which explicitly considers the manifold structure. To be specific, the manifold structure is modeled by a nearest-neighbor graph which preserves the local structure of the image space. A face subspace is obtained by Locality Preserving Projections (LPP) [9]. Each face image in the image space is mapped to a lowdimensional face subspace, which is characterized by a set of feature images, called Laplacianfaces.

LPP can also be obtained from statistical viewpoint. Suppose the data points follow some underlying distribution. Let $\mathbf{x}$ and $\mathbf{y}$ be two random variables. We define that a linear mapping $\mathbf{x} \rightarrow \mathbf{w}^T \mathbf{x}$ best preserves the local structure of the underlying distribution in the L^2 sense if it minimizes the L^2 distances between $\mathbf{w}^T \mathbf{x}$ and $\mathbf{w}^T \mathbf{y}$ provided that $\|\mathbf{x} - \mathbf{y}\| < \varepsilon$ Namely:

$$\min E(|\mathbf{w}^T \mathbf{x} - \mathbf{w}^T \mathbf{y}|^2 \mid \|\mathbf{x} - \mathbf{y}\| < \varepsilon)$$

where ε is sufficiently small, and $\varepsilon > 0$. Here, ε defines the “locality”. Define $\mathbf{z} = \mathbf{x} - \mathbf{y}$, then we have the following objective function:

$$\min E(|\mathbf{w}^T \mathbf{z}|^2 \mid \|\mathbf{z}\| < \varepsilon)$$

It follows that,

$$\begin{aligned} & E(|\mathbf{w}^T \mathbf{z}|^2 \mid \|\mathbf{z}\| < \varepsilon) \\ &= E(\mathbf{w}^T \mathbf{z} \mathbf{z}^T \mathbf{w} \mid \|\mathbf{z}\| < \varepsilon) \\ &= \mathbf{w}^T E(\mathbf{z} \mathbf{z}^T \mid \|\mathbf{z}\| < \varepsilon) \mathbf{w} \end{aligned}$$

Given a set of sample points $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$, we first define an indicator function S_{ij} as follows:

$$S_{ij} = \begin{cases} 1, & \|\mathbf{x}_i - \mathbf{x}_j\|^2 < \varepsilon \\ 0 & \text{otherwise.} \end{cases}$$

Let d be the number of non-zero S_{ij} , and D be a diagonal matrix whose entries are column (or row, since S is symmetric) sums of S , $D_{ii} = \sum_j S_{ji}$. By the Strong Law of Large Numbers, $E(\mathbf{z}\mathbf{z}^T | \|\mathbf{z}\| < \epsilon)$ can be estimated from the sample points as follows:

$$\begin{aligned}
& E(\mathbf{z}\mathbf{z}^T | \|\mathbf{z}\| < \epsilon) \\
& \approx \frac{1}{d} \sum_{\|\mathbf{z}\| < \epsilon} \mathbf{z}\mathbf{z}^T \\
& = \frac{1}{d} \sum_{\|\mathbf{x}_i - \mathbf{x}_j\| < \epsilon} (\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T \\
& = \frac{1}{d} \sum_{i,j} (\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T S_{ij} \\
& = \frac{1}{d} \left(\sum_{i,j} \mathbf{x}_i \mathbf{x}_i^T S_{ij} + \sum_{i,j} \mathbf{x}_j \mathbf{x}_j^T S_{ij} - \sum_{i,j} \mathbf{x}_i \mathbf{x}_j^T S_{ij} - \sum_{i,j} \mathbf{x}_j \mathbf{x}_i^T S_{ij} \right) \\
& = \frac{2}{d} \left(\sum_i \mathbf{x}_i \mathbf{x}_i^T D_{ii} - \sum_{i,j} \mathbf{x}_i \mathbf{x}_j^T S_{ij} \right) \\
& = \frac{2}{d} (XDX^T - XSX^T) \\
& = \frac{2}{d} LX^T
\end{aligned}$$

where $L = D - S$ is the Laplacian matrix. The i th column of matrix X is $\mathbf{x}_i$. By imposing the same constraint, we finally get the same minimization problem described in . In this section, we present a theoretical analysis of LPP and its connections to PCA and LDA.

5.1 Connections to PCA

We first discuss the connections between LPP and PCA. It is worthwhile to point out that XLX^T is the data covariance matrix, if the Laplacian matrix $\frac{1}{n}I - \frac{1}{n^2}\mathbf{e}\mathbf{e}^T$ where n is the number of data points, I is the identity matrix and $\mathbf{e}$ is a column vector taking 1 at each entry. In fact, the Laplacian matrix here has the effect of removing the sample mean from the sample vectors. In this case, the weight matrix S takes $1/n^2$ at each entry, i.e. $S_{ij} = 1/n^2, \forall i, j. D_{ii} = 1/n$.

i.e. $\mathbf{m} = 1/n \sum_i \mathbf{x}_i$. We have:

$$\begin{aligned}
XLX^T &= \frac{1}{n} X(I - \frac{1}{n} \mathbf{e}\mathbf{e}^T) X^T \\
&= \frac{1}{n} XX^T - \frac{1}{n^2} (X\mathbf{e})(X\mathbf{e})^T \\
&= \frac{1}{n} \sum_i \mathbf{x}_i \mathbf{x}_i^T - \frac{1}{n^2} (n\mathbf{m})(n\mathbf{m})^T \\
&= \frac{1}{n} \sum_i (\mathbf{x}_i - \mathbf{m})(\mathbf{x}_i - \mathbf{m})^T + \frac{1}{n} \sum_i \mathbf{x}_i \mathbf{m}^T + \frac{1}{n} \sum_i \mathbf{m} \mathbf{x}_i^T - \frac{1}{n} \sum_i \mathbf{m} \mathbf{m}^T - \mathbf{m} \mathbf{m}^T \\
&= E[(\mathbf{x} - \mathbf{m})(\mathbf{x} - \mathbf{m})^T] + 2\mathbf{m}\mathbf{m}^T - 2\mathbf{m}\mathbf{m}^T \\
&= E[(\mathbf{x} - \mathbf{m})(\mathbf{x} - \mathbf{m})^T]
\end{aligned}$$

where $E[(\mathbf{x} - \mathbf{m})(\mathbf{x} - \mathbf{m})^T]$ is just the covariance matrix of the data set. The above analysis shows that the weight matrix S plays a key role in the LPP algorithm. When we aim at preserving the global structure, we take ϵ (or k) to be infinity and choose the eigenvectors (of the matrix XLX^T) associated with the largest eigenvalues. Hence the data points are projected along the directions of maximal variance. When we aim at preserving the local structure, we take ϵ to be sufficiently small and choose the eigenvectors (of the matrix XLX^T) associated with the smallest eigenvalues. Hence the data points are projected along the directions preserving locality. It is important to note that, 11 when ϵ (or k) is sufficiently small, the Laplacian matrix is no longer the data covariance matrix, and hence the directions preserving locality are not the directions of minimal variance. In fact, the directions preserving locality are those minimizing local variance. 5.2 Connections to LDA LDA seeks directions that are efficient for discrimination. The projection is found by solving the generalized eigenvalue problem

$$\bullet B \mathbf{w} = \lambda SW \mathbf{w}$$

where S and W are defined in section 2. Suppose there are l classes. The i th class contains n_i sample points. Let $\mathbf{m}(i)$ denote the average vector of the i th class. Let $\mathbf{x}(i)$ denote the random vector associated to the i th class and $(i)j \mathbf{x}$ denote the j th sample point in the i th class. We can rewrite the matrix SW as follows:

$$\begin{aligned}
S_W &= \sum_{i=1}^I \left(\sum_{j=1}^{n_i} (\mathbf{x}_j^{(i)} - \mathbf{m}^{(i)}) (\mathbf{x}_j^{(i)} - \mathbf{m}^{(i)})^T \right) \\
&= \sum_{i=1}^I \left(\sum_{j=1}^{n_i} (\mathbf{x}_j^{(i)} (\mathbf{x}_j^{(i)})^T - \mathbf{m}^{(i)} (\mathbf{x}_j^{(i)})^T - \mathbf{x}_j^{(i)} (\mathbf{m}^{(i)})^T + \mathbf{m}^{(i)} (\mathbf{m}^{(i)})^T) \right) \\
&= \sum_{i=1}^I \left(\sum_{j=1}^{n_i} \mathbf{x}_j^{(i)} (\mathbf{x}_j^{(i)})^T - n_i \mathbf{m}^{(i)} (\mathbf{m}^{(i)})^T \right) \\
&= \sum_{i=1}^I \left(X_i X_i^T - \frac{1}{n_i} (\mathbf{x}_1^{(i)} + \dots + \mathbf{x}_{n_i}^{(i)}) (\mathbf{x}_1^{(i)} + \dots + \mathbf{x}_{n_i}^{(i)})^T \right) \\
&= \sum_{i=1}^I \left(X_i X_i^T - \frac{1}{n_i} X_i (\mathbf{e}_i \mathbf{e}_i^T) X_i^T \right) \\
&= \sum_{i=1}^I X_i L_i X_i^T
\end{aligned}$$

where $X_i L_i X_i^T$ is the data covariance matrix of $X_i = [\mathbf{x}_1^{(i)}, \mathbf{x}_2^{(i)}, \dots, \mathbf{x}_{n_i}^{(i)}]$ is a $d \times n_i$ matrix where I is the identity matrix and $\mathbf{e}_i = (1, 1, \dots, 1)^T$ is an n_i dimensional vector. To further simplify the above equation, we define:

$$W_{ij} = \begin{cases} 1/n_k & \text{if } \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ both belong to the } k^{th} \text{ class} \\ 0 & \text{otherwise} \end{cases}$$

$$L = I - W$$

Thus, we get:

$$S_W = X L X^T$$

otherwise 1/ if and both belong to the class Thus, It is interesting to note that we could regard the matrix W as the weight matrix of a graph with data points as its nodes. Specifically, W_{ij} is the weight of the edge (x_i, x_j) . W reflects the class relationships of the data points. The matrix L is thus called graph Laplacian, which plays key role in LPP [9]. Similarly, we can compute the matrix S_B as follows:

$$\begin{aligned}
S_B &= \sum_{i=1}^l n_i (\mathbf{m}^{(i)} - \mathbf{m})(\mathbf{m}^{(i)} - \mathbf{m})^T \\
&= \left(\sum_{i=1}^l n_i \mathbf{m}^{(i)} (\mathbf{m}^{(i)})^T \right) - 2\mathbf{m} \left(\sum_{i=1}^l n_i \mathbf{m}^{(i)} \right) + \left(\sum_{i=1}^l n_i \right) \mathbf{m} \mathbf{m}^T \\
&= \left(\sum_{i=1}^l \frac{1}{n_i} (\mathbf{x}_1^{(i)} + \dots + \mathbf{x}_{n_i}^{(i)}) (\mathbf{x}_1^{(i)} + \dots + \mathbf{x}_{n_i}^{(i)})^T \right) - 2n\mathbf{m} \mathbf{m}^T + n\mathbf{m} \mathbf{m}^T \\
&= \left(\sum_{i=1}^l \sum_{j,k=1}^{n_i} \frac{1}{n_i} \mathbf{x}_j^{(i)} (\mathbf{x}_k^{(i)})^T \right) - 2n\mathbf{m} \mathbf{m}^T + n\mathbf{m} \mathbf{m}^T \\
&= XWX^T - 2n\mathbf{m} \mathbf{m}^T + n\mathbf{m} \mathbf{m}^T \\
&= XWX^T - n\mathbf{m} \mathbf{m}^T \\
&= XWX^T - X \left(\frac{1}{n} \mathbf{e} \mathbf{e}^T \right) X^T \\
&= X \left(W - \frac{1}{n} \mathbf{e} \mathbf{e}^T \right) X^T \\
&= X \left(W - I + I - \frac{1}{n} \mathbf{e} \mathbf{e}^T \right) X^T \\
&= -XLX^T + X \left(I - \frac{1}{n} \mathbf{e} \mathbf{e}^T \right) X^T \\
&= -XLX^T + C
\end{aligned}$$

Thus, the generalized eigenvector problem of LDA can be written as follows:

$$\begin{aligned}
S_B \mathbf{w} &= \lambda S_W \mathbf{w} \\
\Rightarrow (C - XLX^T) \mathbf{w} &= \lambda XLX^T \mathbf{w} \\
\Rightarrow C \mathbf{w} &= (1 + \lambda) XLX^T \mathbf{w} \\
\Rightarrow XLX^T \mathbf{w} &= \frac{1}{1 + \lambda} C \mathbf{w}
\end{aligned}$$

Thus, the projections of LDA can be obtained by solving the following generalized eigenvalue

$$XLX^T \mathbf{w} = \lambda C \mathbf{w}$$

The optimal projections correspond to the eigenvectors associated with the smallest eigenvalues. If the sample mean of the data set is zero, the covariance matrix is simply which is close to the matrix $XDXT$ in the LPP algorithm. Our analysis shows that LDA actually aims to preserve discriminating information and global geometrical structure. Moreover, LDA has a similar form to LPP. However, LDA is supervised while LPP can be performed in either

supervised or unsupervised manner. Proposition 1: The rank of L is $n - c$. Proof: Without loss of generality, we assume that the data points are ordered according to which class they are in, so that $\{x_1, \dots, x_{n_1}\}$ are in the first class, $\{x_{n_1+1}, \dots, x_{n_1+n_2}\}$ are in the second class, etc. Thus, we can write L as follows:

$$L = \begin{bmatrix} L_1 & & & \\ & L_2 & & \\ & & \ddots & \\ & & & L_c \end{bmatrix}$$

where L_i is a square matrix,

By adding all but the first column vectors to the first column vector and then subtracting the first row vector from any other row vectors, we get the following matrix whose rank is $n_i - 1$. Therefore, the rank of and hence the rank of L . Proposition 1 tells us that the rank of $XLXT$ is at most $n - c$. However, in many cases in appearance-based face recognition, the number of pixels in an image (or, the dimensionality of the image space) is larger than $n - c$, i.e. $d > n - c$. Thus, $XLXT$ is singular. In order to overcome the complication of a singular $XLXT$, Belhumeur et. al. [1] proposed the Fisherface approach that the face images are projected from the original image space to a subspace with dimensionality $n - c$ and then LDA is performed in this subspace.

CHAPTER 5

RESULTS AND DISCUSSION

Once the Laplacianfaces are created, face recognition [2][14][28][29] becomes a pattern classification task. In this section, we investigate the performance of our proposed Laplacianfaces method for face recognition. The system performance is compared with the Eigenfaces method [28] and the Fisherfaces method [2], two of the most popular linear methods in face recognition. In this study, three face databases were tested. The first one is the PIE (pose, illumination, and expression) database from CMU [25], the second one is the Yale database [30], and the third one is the MSRA database collected at the Microsoft Research Asia. In all the experiments, preprocessing to locate the faces was applied. Original images were normalized (in scale and orientation) such that the two eyes were aligned at the same position. Then, the facial areas were cropped into the final images for matching. The size of each cropped image in all the experiments is 32×32 pixels, with 256 grey levels per pixel. Thus, each image is represented by a 1024-dimensional vector in image space. No further preprocessing is done. Figure 5 shows an example of the original face image and the cropped image. Different pattern classifiers have been applied for face recognition, including nearest-neighbor [2], Bayesian [15], Support Vector Machine [17], etc. In this paper, we apply the nearest-neighbor classifier for its simplicity. In short, the recognition process has three steps. First, we calculate the Laplacianfaces from the training set of face images; then the new face image to be identified is projected into the face subspace spanned by the Laplacianfaces finally, the new face image is identified by a nearest-neighbor classifier. Figure 5. The original face image and the cropped image. The Yale face database [30] was constructed at the Yale Center for Computational Vision and Control. It contains 165 grayscale images of 15 individuals. The images demonstrate variations in lighting condition (left-light, center-light, right-light), facial expression (normal, happy, sad, sleepy, surprised, and wink), and with/without glasses. A random subset with six images per individual (hence 90 images in total) was taken with labels to form the training set. The rest of the database was considered to be the testing set. The training samples were used to learn the Laplacianfaces. The testing samples were then projected into the lowdimensional representation subspace. Recognition was performed using a nearest neighbor classifier. We averaged the results over 20 random splits. Figure 6. Recognition accuracy vs. dimensionality reduction on Yale database.

Approach	Dims	Error Rate
Eigenfaces	33	25.3%
Fisherfaces	14	20.0%
Laplacianfaces	28	11.3%

In the above subsections, we have shown that our Laplacianfaces approach outperforms the Eigen-faces and Fisherfaces approaches on three different face databases. In this section, we evaluate the per-formances of these three approaches with different number of training samples. We expect that a good algorithm should be able to take advantage of more training samples.

We used the MSRA database for this experiment. Let *MSRA-S1* denote the image set taken in the first session, and *MSRA-S2* denote the image set taken in the second session. *MSRA-S2* was exclusively used for testing. 10%, 40%, 70%, and 100% face images were randomly selected from *MSRA-S1* for training. For each of the first three cases, we repeated 20 times and computed the error rate on average. In Figure 11 and Table 4 we show the experimental results of the improvement of the performance of our algorithm with the number of training samples. As can be seen, our algorithm takes advantage of more training samples. The performance improves significantly as the number of training sample in-creases. The error rate of recognition reduces from 26.04% with 10% training samples to 8.17% with 100% training samples. This is due to the fact that, our algorithm aims at discovering the underlying face manifold embedded in the ambient space and more training samples help to recover the manifold structure. Eigenfaces also takes advantage of more training samples to some extent. The error rate of Eigenfaces reduces from 46.48% with 10% training samples to 35.42% with 100% training samples. But its performance starts to converge at 40% (training samples) and little improvement is obtained if more training samples are used. For Fisherfaces, there is nconvincing evidence that it can take advantage of more training samples.

Discussions

Three experiments on three databases have been systematically performed. These experiments re-veal a number of interesting points:

All these three approaches performed better in the optimal face subspace than in the original image space.

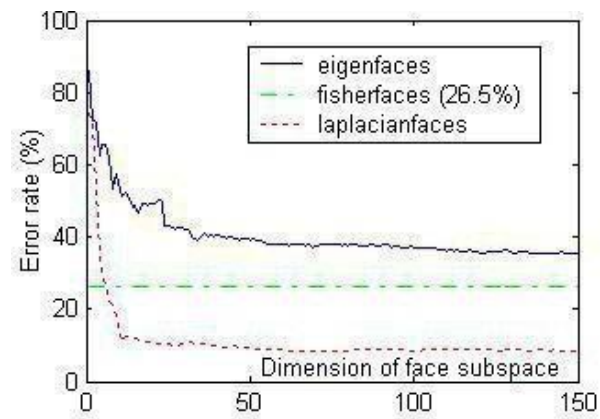
In all the three experiments, Laplacianfaces consistently performs better than Eigenfaces and Fisherfaces. Especially, it significantly outperforms Fisherfaces and Eigenfaces on Yale database and MSRA database. These experiments also show that our algorithm is especially suitable for frontal face images. Moreover, our algorithm takes advantage of more training samples, which is important to the real world face recognition systems.

Comparing to the Eigenfaces method, the Laplacianfaces method encodes more discriminating information in the low-dimensional face subspace by preserving local structure which is more important than the global structure for classification, especially when nearest neighbor like classifiers are used. In fact, if there is a reason to believe that Euclidean distances ($\|x_i - x_j\|$) are meaningful only if they are small (local), then our algorithm finds a projection that respects such a belief. An-other reason is that, as we show in Figure 1, the face images probably reside on a nonlinear manifold. Therefore, an efficient and effective subspace representation of face images should be capable of charactering the nonlinear manifold structure, while the Laplacianfaces are exactly derived by finding the optimal linear approximations to the eigenfunctions of the Laplace Betrami operator on the face manifold. By discovering the face manifold structure, Laplacianfaces can identify the person with different expressions, poses, and lighting conditions.

	Percentage of training samples in <i>MSRA-S1</i>			
	10%	40%	70%	100%

Laplacianfaces	26.04%	14.71%	13.15%	8.17%
Fisherfaces	26.96%	20.05%	29.43%	26.43%
Eigenfaces	46.48%	37.76%	38.54%	35.42%

Fig 5.1



CHAPTER 6

CONCLUSION AND FUTURE WORK

Our system is proposed to use Locality Preserving Projection in Face Recognition which eliminates the flaws in the existing system. This system make the faces to reduce into lower dimensions and algorithm for LPP is performed for recognition. The application is developed successfully and implemented as mentioned above.

This system seems to be working fine and successfully. This system can able to provide the proper training set of data and test input for recognition. The face matched or not is given in the form of picture image if matched and text message in case of any difference.

The manifold ways of face analysis (representation and recognition) is introduced in this paper in order to detect the underlying nonlinear manifold structure in the manner of linear subspace learning. To the best of our knowledge, this is the first devoted work on face representation and recognition which explicitly considers the manifold structure. The manifold structure is approximated by the adjacency graph computed from the data points. Using the notion of the Laplacian of the graph, we then compute a transformation matrix which maps the face images into a face subspace. We call this the Laplacianfaces approach. The Laplacianfaces are obtained by finding the optimal linear approximations to the eigen-functions of the Laplace Beltrami operator on the face manifold. This linear transformation optimally preserves local manifold structure. Theoretical analysis of the LPP algorithm and its connections to PCA and LDA are provided. Experimental results on PIE, Yale and MSRA databases show the effectiveness of our method.

One of the central problems in face manifold learning is to estimate the intrinsic dimensionality of the *nonlinear* face manifold, or, degrees of freedom. We know that the dimensionality of the manifold is equal to the dimensionality of the local tangent space. Some previous works [33][34] show that the local tangent space can be approximated using points in a neighbor set. Therefore, one possibility is to estimate the dimensionality of the tangent space.

Another possible extension of our work is to consider the use of the unlabeled samples. It is important to note that the work presented here is a general method for face analysis (face

representation and recognition) by discovering the underlying face manifold structure. Learning the face manifold (learning Laplacianfaces) is essentially an unsupervised learning process. And in many practical cases, one finds a wealth of easily available unlabeled samples. These samples might help to discover the face manifold. For example, in [4], it is shown how unlabeled samples are used for discovering the manifold structure and hence improving the classification accuracy. Since the face images are believed to reside on a sub-manifold embedded in a high-dimensional ambient space, we believe that the unlabeled samples are of great value. We are currently exploring these problems in theory and practice.

REFERENCES

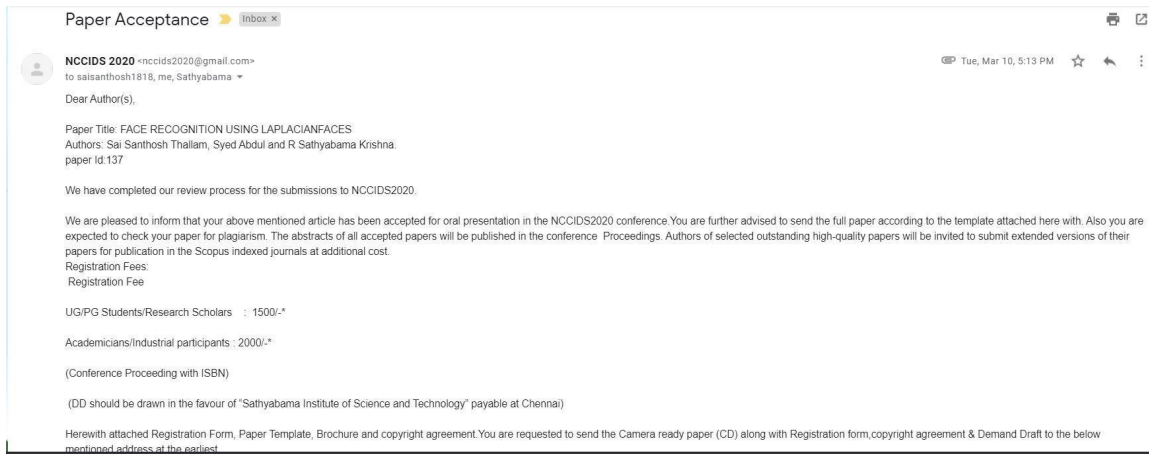
- [1] A. U. Batur and M. H. Hayes, "Linear Subspace for Illumination Robust Face Recognition" *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, Hawaii, Dec. 11-13, 2001.
- [2] P. N. Belhumeur, J. P. Hespanha and D. J. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, No. 7, 1997, pp. 711-720.
- [3] M. Belkin and P. Niyogi, "Laplacian Eigenmaps and Spectral Techniques for Embedding and Clustering", *Advances in Neural Information Processing System 15*, Vancouver, British Columbia, Canada, 2001.
- [4] M. Belkin and P. Niyogi, "Using Manifold Structure for Partially Labeled Classification", *Advances in Neural Information Processing System 15*, Vancouver, British Columbia, Canada, 2002.
- [5] M. Brand, "Charting a Manifold", *Advances in Neural Information Processing Systems*, Vancouver, Canada, 2002.
- [6] Fan R. K. Chung, Spectral Graph Theory, *Regional Conferences Series Mathematics*, number 92, 1997.
- [7] Y. Chang, C. Hu and M. Turk, "Manifold of Facial Expression", *Proc. IEEE International Workshop on Analysis and Modeling of Faces and Gestures*, Nice, France, Oct2003.
- [8] R. Gross, J. Shi, and J. Cohn, "Where to go with Face Recognition", *Third Workshop on Empirical Evaluation Methods in Computer Vision*, Kauai, Hawaii, December 11-13, 2001.
- [9] Xiaofei He and Partha Niyogi, "Locality Preserving Projections", *Advances in Neural Information Processing Systems*, Vancouver, Canada, 2003.
- [10] K.-C. Lee, J. Ho, M.-H. Yang, and D. Kriegman, "Video-Based Face Recognition Using Probabilistic Appearance Manifolds", *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 313-320, vol. 1, 2003.
- [11] A. Levin and A. Shashua, "Principal Component Analysis Over Continuous Subspaces and Intersection of Half-Spaces", *Proc. of the European Conference on Computer Vision (ECCV)*, Copenhagen, Denmark, May 2002
- [12] S. Z. Li, X. W. Hou, H. J. Zhang, Q. S. Cheng, "Learning Spatially Localized, Parts-Based Representation", *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, Hawaii, Dec. 11-13, 2001.

- [13] Q. Liu, R. Huang, H. Lu, and S. Ma, "Face Recognition Using Kernel Based Fisher Discriminant Analysis", in *Proc. of the fifth Int. Conf. on Automatic Face and Gesture Recognition*, Washington D. C., May 2002.
- [14] A. M. Martinez and A. C. Kak, "PCA versus LDA", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23(2):228-233, 2001.
- [15] B. Moghaddam and A. Pentland, "Probabilistic Visual Learning for Object Representation", *IEEE Trans. On Pattern Analysis and Machine Intelligence*, vol 19, pp. 696-710, 1997.
- [16] H. Murase and S. K. Nayar, "Visual Learning and Recognition of 3-D objects from appearance", *International Journal of Computer Vision*, 14:5-24, 1995.
- [17] P. J. Phillips, "Support Vector Machines Applied to Face Recognition". In *Advances in Neural Information Processing Systems* 11, pp. 803-809, 1998.
- [18] Sam T. Roweis, and Lawrence K. Saul, "Nonlinear Dimensionality Reduction by Locally Linear Embedding", *Science*, vol 290, 22 December 2000.
- [19] Sam Roweis, Lawrence Saul and Geoff Hinton, "Global Coordination of Local Linear Models", *Advances in Neural Information Processing System* 14, 2001.
- [20] Lawrence K. Saul and Sam T. Roweis, "Think Globally, Fit Locally: Unsupervised Learning of Low Dimensional Manifolds", *Journal of Machine Learning Research*, v4, pp. 119-155, 2003.

APPENDIX

A. PAPER ACCEPTANCE MAIL

FACE RECOGNITION BY LAPLACIANFACES



B. PLAGIARISM REPORT

Turnitin

https://www.turnitin.com/newreport_printview.asp?eq=1&eb=1&esm...

Turnitin Originality Report

Processed on: 21-Feb-2020 12:23 AM CST
ID: 1261340981
Word Count: 1754
Submitted: 1

C By C C

Similarity Index

5%

Similarity by Source

Internet Sources:	0%
Publications:	3%
Student Papers:	2%

3% match (publications)

[Haiyang Zhang. "Grid-Based Multi-scale PCA Method for Face Recognition in the Large Face Database", Lecture Notes in Computer Science, 2006](#)

C. JOURNAL PAPER

FACE RECOGNITION USING LAPLACIANFACES

Syed Abdul¹, Thallam Sai Santhosh², Ms. R. Sathyabama Krishna³

[1][2]Student, Dept. of CSE, Sathyabama Institute of Science and Technology,
Chennai, India

[3]Professor, Dept. of CSE, Sathyabama Institute of Science and Technology,
Chennai, India

Author email id: abdulsyedabd@gmail.com,
saisanthosh1818@gmail.com, sathyabamar.cse@sathyabama.ac.in

Abstract: We propose an facial recognition system called LaplacianFace recognition. Using the location preserving projections, images assigned to subspace of faces are examine. It is not exactly equivalent to the main component The evaluation PCA & the linear discriminant analysis, LDA observe only the structure of facial space. This finds entering of information gaining from a face subspace that recognizes the better complex structure of the main face. The LaplacianFaces are the perfect approximations directed to functions of manager of Laplace in the facial complex. Therefore, the annoying faces that arise due to changes in lighting, external appearance and posture can be detected. Speculative examination shows that PCA as well as LDA along with LPP can be derived from various models of the graphic. The test results suggest that Laplacian face proposed approach offers unmatched representation and gets less failure rates of face affirmation.

Keywords: Principal Component Analysis(PCA), Linear Discriminant Analysis(LDA), Locality Preserving Projection(LPP), LaplacianFaces.

1. Introduction

A LaplacianFace is one that can recognize individuals and respond suitably. In this way, one of the most significant structure squares of shrewd situations is an individual ID framework. Face acknowledgment gadgets are perfect for such frameworks, since they have as of late become quick.

Face recognition that are based on the PC security frameworks which can ordinarily perceive, with visiting the appearance of humans. They rely on an insistence check like eigen face or secured model like Markov. The fundamental development of facial attestation framework to see the face of human and concentrate it the remainder of it. The structure calculates nodal bases upon face, for example, in segment in-between eyes, the state of cheekbones along with different prominent highlights. The nodal focuses are by at that point showed up contrastingly comparable to the nodal focuses arranged from a base of pictures so as to find the match. Indisputably, such as structure that bound dependent upon reason for a face that got conditioning of light present. Advances at beginning at now in movement to make 3D models of an individual's face subject to a mechanized photo so as to make more

nodal communities for relationship. In any case, such improvement is typically vulnerable to mess up given that the PC is extrapolating a three-dimensional model from a 2D photo.

2. Methods

Principal Component Analysis(PCA) and Linear Discriminant Analysis(LDA)

The only way deal with oversee acclimating to issue of absurd dimension of the picture space is for diminish the dimension by joining highlights. Straight mixes are express, captivating considering how they are definitely not hard to select additionally, proficiently tractable. Appropriately, direct methodology paper the large information into a small dimensional subspace.

Considering the issue of tending to the aggregate of the vectors in a huge amount of large dimensional models $x_1; x_2; \dots; x_n$, with zero mean, by a solitary vector $y = \{y_1; y_2; \dots; y_n\}$ with the end goal that y_i tends to x_i . In particular, we locate a prompt mapping from the d -dimensional space to a line. Without loss of complete declaration, we mean the change vector by w . That is, $w^T x_i = y_i$. In fact, the hugeness of w is of no genuine critical ness since it just scales y_i . In face insistence, every vector x_i shows a face picture.

principal component analysis (PCA) is a real framework that uses a balanced change to change over the view of a set maybe related elements from set of estimations of straightly different elements called head sections. Head sections isn't actually or then again equal to the amount of one of a kind elements. This change is portrayed with the goal that the principle head portion has the greatest possible change (that is, accounts for anyway a great part of the irregularity in the data as could sensibly be normal), and each succeeding section in this manner has the most critical change can be from the basic that it is balanced to(i.e., uncorrelated with) the past parts. PCA is delicate to the general scaling of the exceptional segments.

LDA can be likewise identified along PCA to examine both of that search for direct mixes of factors to get the best data.LDA endeavors to explain the contrast among information classes. PCA then again will not take into account any distinct in class and factor investigation constructs the include blends which depends on and opposed to similitude's. Discriminant examine is likewise not the same as calculate investigation that it's anything but an association system

2.2 Locality Preserving

Projection(LPP)

PCA along with LDA intend to save the worldwide structures. Be that as it may, in some certifiable applications, the neighborhood structure is increasingly significant. Right now, depict Locality Protecting Projection (LPP) , another calculation for learning a territory saving subspace

LPP are direct Projectile maps that emerge when a difference is settled Issue that ideally saves the area Structure of the information record. BVG should resemble a Decision as opposed to head part investigation (PCA) - A Standard straight procedure that reports information along the Course of greatest difference. Exactly when the stop The dimensional information are found in a low measurement different Consolidated into the ecological space, the protection of the region The projections are acquired by finding the ideal direct Ways to deal with the segments of Laplace Beltrami administrator in the authority

3. Learning LaplacianFaces

LPP is a general system of complex learning. We get by finding the ideal straight approximate to the eigen parts of the Laplace official of the complex. At the present time, it is as of recently a straight framework, it appears to

recoup important bits of the trademark nonlinear complex structure by saving near to structure.

In context on faces, we portray our Laplacian faces strategy face delineation for safeguarding subspace. The face assessment and certification issue, one is gone looking with the trouble that the grid XDXT for every once in a while solitary. This stems from the way that now and then the measure of pictures in the course of action set is considerably more little for the measure of data in each picture. In this case, the position of XDXT is everything seen as n , while XDXT is an network, which suggests that XDXT is single. To conquer the difficulty of a particular XDXT, we first paper the picture set to a PCA subspace with the target that the following lattice XDXT is non singular. Another thought of utilizing PCA has made it before is for commotion decay. This strategy, we call it as Laplacian faces, can get to know an ideal subspace for face portrayal and attestation.

4. Visual Analysis

By and large, face pictures perhaps envisioned as bases drawn on a low-dimensional complex hid in a high-dimensional wrapping space. Staggeringly, we can think about that as a sheet of adaptable is fell into a

small ball. The goal of an mapping is to make less dimension. In this event that the paper is torn simultaneously, the mapping is topology guaranteeing. In addition, if the flexible isn't expanded or obviously squeezed, the mapping jam the estimation structure of the essential space. Right now, the main goal is to find the face complex by a locally topology-saving mapping for face appraisal.

5. Face complex investigation

Consider a fundamental instance of picture vacillation. Assume that a great data of face pictures is created while the human face turns progressively. Naturally, the face pictures identify with a predictable curve in the picture space because there is only a solitary degree of chance. The sacred emissary of rotate. From this can you say that the plan of face pictures are normally one dimensional. Right now, can say that the game plan of face pictures are normally one dimensional. Various continuous works have demonstrated that the face pictures do live on a low dimensional sub complex introduced in a high-dimensional including space (picture space). Thus, a convincing subspace learning figuring should have the choice to recognize

the nonlinear complex structure. The standard figures , for instance, PCA and LDA, model the face pictures in Euclidean space. They reasonably watch only the Euclidean structure. Thus , they disregard to recognize the normal low- dimension. With its neighborhood securing character, the Laplacian faces seem to have the alternative to get the inborn face complex structure to a greater degree. This shows a model that the face pictures with various stance and presence of an individual are mapped into two-dimensional subspace.

The picture of face educational record used here is identical to that used in. This data set contains several face pictures taken from progressive housings of a little video. The size of each image is in the form of pixels, with dull levels for every pixel. In this manner, each face picture is addressed for an in the enveloping space. Regardless, these photos are acknowledged to start from a sub complex with hardly any degree of chance.

6. Face Representation

The face portrayed already, and the picture can be spoken to as a point in picture space . Be that as it may, because of the undesirable varieties

coming about because of changes in light, outward appearance and the picture space may not be an ideal space for seeing.

In the 3rd sec , we had talked about the use to gain proficiency with certain region saving face subspace which is inhumane toward exception and clamor. The pictures of appearances in the preparation set are utilized to learn subspace.

7. Conclusion

This Paper is by all accounts working fine and effectively. This framework can ready to give the correct preparing set of information and test contribution for acknowledgment. The face coordinated or not is given as picture if coordinated and instant message if there should be an occurrence of any distinction. Face acknowledgment innovation has progressed significantly in twenty years. Today, machines are consequently confirm character data for secure exchanges, for reconnaissance and security errands, and for get to control of structures and so forth.

These is used for the most of the work in certain conditions and to know calculations that can change the ecological limitations for acquire good quality . But it was , cutting the edge

face data frameworks will have far reaching application in certain situations - where all are like collaborators.

8. References

1. U. Batur and M. H. Hayes, "Linear Subspace for Illumination Robust Face Recognition" IEEE Int. Conf. on Computer Vision and Pattern Recognition, Hawaii, Dec. 11-13, 2001.
2. P. N. Belhumeur, J. P. Hespanha and D. J. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection", IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 19, No. 7, 1997, pp. 711-720.
3. M. Belkin and P. Niyogi, "Laplacian Eigenmaps and Spectral Techniques for Embedding and Clustering", Advances in Neural Information Processing System 15, Vancouver, British Columbia, Canada, 2001.
4. M. Belkin and P. Niyogi, "Using Manifold Structure for Partially Labeled Classification", Advances in Neural Information Processing System 15, Vancouver, British Columbia, Canada, 2002.
5. M. Brand, "Charting a Manifold", Advances in Neural Information Processing Systems, Vancouver, Canada, 2002.
6. Fan R. K. Chung, Spectral Graph Theory, Regional Conferences Series Mathematics, number 92, 1997.
7. Y. Chang, C. Hu and M. Turk, "Manifold of Facial Expression", Proc. IEEE International Workshop on Analysis and Modeling of Faces and Gestures, Nice, France, Oct 2003.
8. R. Gross, J. Shi, and J. Cohn, "Where to go with Face Recognition", Third Workshop on Empirical Evaluation Methods in Computer Vision, Kauai, Hawaii, December 11-13, 2001.
9. Xiaofei He and Partha Niyogi, "Locality Preserving Projections", Advances in Neural Information Processing Systems, Vancouver, Canada, 2003.
10. K.-C. Lee, J. Ho, M.-H. Yang, and D. Kriegman, "Video-Based Face Recognition Using Probabilistic Appearance Manifolds", Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp. 313-320, vol. 1, 2003.

D. SOURCE CODE

```
//LaplacianFaceRecog.java
import java.lang.*;

import java.io.*;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

class LaplacianFaceRecog extends JFrame implements ActionListener
{
    JFrame frmMain=new JFrame("Face Recognition Using
LaplacianFaces");
    JLabel lblTestPath=new JLabel("Input Test Image:");
    JTextField txtTestPath=new JTextField("images\\test\\face16d.pgm");
    JButton btRecognize=new JButton("Recognize");
    JLabel lblResult=new JLabel("Result:");
    JTextArea txtResult=new JTextArea("");
    JScrollPane spResult=new JScrollPane(txtResult);
    JFrame frmImage1=new JFrame("Input Image");
    JFrame frmImage2=new JFrame("Matched Image");

    //system declarations
    int MaxFaceIndex=17;
    double DifferenceThreshold=15000.0;
    int NumFaces;
    int MaxFaces=100;
    int FaceTemplate[][];
    int Faces[][][];
    int LaplacianFaces[][][];
    String FaceFileNames[];
```

```
String tResult="";
```

```
//constructor
```

```
public LaplacianFaceRecog()
```

```
{
```

```
    frmMain.setDefaultLookAndFeelDecorated(true);  
    frmMain.setResizable(false);  
    frmMain.setBounds(100,100,315,250);  
    frmMain.getContentPane().setLayout(null);  
    lblTestPath.setBounds(17,15,100,20);  
    frmMain.getContentPane().add(lblTestPath);  
    txtTestPath.setBounds(15,35,170,20);  
    frmMain.getContentPane().add(txtTestPath);  
    lblResult.setBounds(17,65,100,20);  
    frmMain.getContentPane().add(lblResult);  
    spResult.setBounds(15,85,280,120);  
    frmMain.getContentPane().add(spResult);  
    txtResult.setEditable(false);  
    btRecognize.setBounds(193,35,100,20);  
    btRecognize.addActionListener(this);  
    frmMain.getContentPane().add(btRecognize);  
    frmImage1.setDefaultLookAndFeelDecorated(true);  
    frmImage1.setResizable(false);  
    frmImage1.setBounds(450,100,200,150);  
    frmImage2.setDefaultLookAndFeelDecorated(true);
```

```

        frmImage2.setResizable(false);
        frmImage2.setBounds(670,100,200,150);

        frmImage1.setVisible(true);
        frmImage2.setVisible(true);
        frmMain.setVisible(true);
    }

    //events

    public void actionPerformed(ActionEvent evt)
    {
        if(evt.getSource()==btRecognize)
        {
            if(new File(txtTestPath.getText()).exists()==false)
            {
                JOptionPane.showMessageDialog(null,"Test
Image, not found.");
                return;
            }
        }
    }

```

```

drawImage(frmlImage1.getGraphics(),txtTestPath.getText());

        tResult="";
        txtResult.setText(tResult);
        train();
        test();

    }

}

//internal methods

void drawImage(Graphics g,String tPath)
{

    PGM tpgm=new PGM();
    tpgm.setFilePath(tPath);
    tpgm.readImage();

    g.clearRect(0,0,200,150);
    for(int r=0;r<tpgm.getRows();r++)
    {

        for(int c=0;c<tpgm.getCols();c++)
        {

            int intensity=tpgm.getPixel(r,c);

            Color color=new Color(intensity,intensity,intensity);
            g.setColor(color);
            g.fillRect(c,r+30,1,1);
        }
    }
}

```

```

        }

    }

}

//methods

public void addResultText(String tStr)

{

    tResult=tResult+tStr;
    txtResult.setText(tResult);
}

public void train()

{

    int xBase,yBase,xSub,ySub;
    int xLow,xHigh,yLow,yHigh;
    int GrayLevel;
    int CellSum,CellAvg;
    int i,j;
    int xDiv,yDiv;

    int BlockWidth,BlockHeight;
    int StartX,StartY;
    int SizeX,SizeY;

    //set system parameters
    SizeX=80;

```

```

SizeY=80;
xDiv=20;
yDiv=20;
BlockWidth=SizeX/xDiv;
BlockHeight=SizeY/yDiv;

FaceTemplate=new int[xDiv][yDiv];
Faces=new int[xDiv][yDiv][MaxFaces];
LaplacianFaces=new int[xDiv][yDiv][MaxFaces];
FaceFileNames=new String[MaxFaces];
NumFaces=0;

addResultText("Training...");
PGM  pgm1=new  PGM();
for(i=0;i<=MaxFaceIndex;i++)
{
    for(j=97;j<=98;j++)/'a' to 'b'
    {
        NumFaces=NumFaces+1;

        FaceFileNames[NumFaces]="images\\train\\face"+i+(char)
j+".pgm";

        PGM_ImageFilteringFilter=newPGM_ImageFilter();

        imgFilter.set_inFilePath(FaceFileNames[NumFaces]);
    }
}

```

```

imgFilter.set_outFilePath("temp.pgm");
imgFilter.resize(SizeX,SizeY);
pgm1.setFilePath("temp.pgm");
pgm1.readImage();

for(xBase=0;xBase<=xDiv-1;xBase++)
{
    for(yBase=0;yBase<=yDiv-1;yBase++)
    {
        StartX=xBase*BlockWidth;
        StartY=yBase*BlockHeight;
        xLow=StartX;
        xHigh=StartX+BlockWidth-1;
        yLow=StartY;
        yHigh=StartY+BlockHeight-1;

        CellSum=0;

        for(xSub=xLow;xSub<=xHigh;xSub++)
        {
            for(ySub=yLow;ySub<=yHigh;ySub++)
            {
                GrayLevel=pgm1.getPixel(xSub,ySub);
            }
        }
    }
}

```

```

CellSum=CellSum+GrayLevel;

}

}

CellAvg=CellSum/(BlockWidth*BlockHeight);

Faces[xBase][yBase][NumFaces]=CellAvg;

}

}

}

}

        CellSum=0;
        for(i=1;i<=NumFaces;i++)
        {
                CellSum=CellSum+Faces[xBase][yBase][i];

        }

        CellAvg=CellSum/NumFaces;
        FaceTemplate[xBase][yBase]=CellAvg;
    }

}

for(xBase=0;xBase<=xDiv-1;xBase++)

```

```

{
    for(yBase=0;yBase<=yDiv-1;yBase++)
    {
        for(i=1;i<=NumFaces;i++)
        {
            LaplacianFaces[xBase][yBase][i]=Faces[xBase][yBase][
i]- FaceTemplate[xBase][yBase];
        }
    }
}

PGM    pgm2=new    PGM();
pgm2.setFilePath("template.pgm");
pgm2.setType("P5");
pgm2.setComment("");
pgm2.setDimension(SizeX,SizeY);
pgm2.setMaxGray(255);

```

```

        StartX=xBase*BlockWidth;
        StartY=yBase*BlockHeight;
        xLow=StartX;
        xHigh=StartX+BlockWidth-1;
        yLow=StartY;
        yHigh=StartY+BlockHeight-1;

        for(xSub=xLow;xSub<=xHigh;xSub++)
        {
            for(ySub=yLow;ySub<=yHigh;ySub++)
            {
                GrayLevel=FaceTemplate[xBase][yBase];
                pgm2.setPixel(xSub,ySub,GrayLevel);
            }
        }

        pgm2.writeImage();
        addResultText("done.");
    }

    public void test()
    {
        int xBase,yBase,xSub,ySub;
        int xLow,xHigh,yLow,yHigh;

```

```

int GrayLevel;
int CellSum,CellAvg;
int i,j;
int xDiv,yDiv;

int BlockWidth,BlockHeight;
int StartX,StartY;
int SizeX,SizeY;

//set system parameters
SizeX=80;
SizeY=80;
xDiv=20;
yDiv=20;
BlockWidth=SizeX/xDiv;
BlockHeight=SizeY/yDiv;

int TestFace[][]=new int[xDiv][yDiv];

int TestLaplacianFace[][]=new int[xDiv][yDiv];
int LaplacianDiff;
int MinLaplacianIndex;

double TotalLaplacianDiff,MinLaplacianDiff;

addResultText("\nTesting...");
PGM pgm1=new PGM();
pgm1.setFilePath(txtTestPath.getText());
pgm1.readImage();

for(xBase=0;xBase<=xDiv-1;xBase++)
{
    for(yBase=0;yBase<=yDiv-1;yBase++)
    {

```

```

        StartX=xBase*BlockWidth;
        StartY=yBase*BlockHeight;
        xLow=StartX;
        xHigh=StartX+BlockWidth-1;
        yLow=StartY;
        yHigh=StartY+BlockHeight-1;

        CellSum=0;
        for(xSub=xLow;xSub<=xHigh;xSub++)
        {
            for(ySub=yLow;ySub<=yHigh;ySub++)
            {
                GrayLevel=pgm1.getPixel(xSub,ySub);
                CellSum=CellSum+GrayLevel;
            }
        }

        CellAvg=CellSum/(BlockWidth*BlockHeight);
        TestFace[xBase][yBase]=CellAvg;
    }
}

PGM    pgm2=new    PGM();
pgm2.setFilePath("diff.pgm");
pgm2.setType("P5");
pgm2.setComment("");
pgm2.setDimension(SizeX,SizeY);
pgm2.setMaxGray(255);

```

```

for(xBase=0;xBase<=xDiv-1;xBase++)
{
    for(yBase=0;yBase<=yDiv-1;yBase++)
    {
        StartX=xBase*BlockWidth;
        StartY=yBase*BlockHeight;
        xLow=StartX;
        xHigh=StartX+BlockWidth-1;
        yLow=StartY;
        yHigh=StartY+BlockHeight-1;

        for(xSub=xLow;xSub<=xHigh;xSub++)
        {
            for(ySub=yLow;ySub<=yHigh;ySub++)
            {
                GrayLevel=TestFace[xBase][yBase];
                pgm2.setPixel(xSub,ySub,GrayLevel);
            }
        }
    }
}

```

```

for(xBase=0;xBase<=xDiv-1;xBase++)
{
    for(yBase=0;yBase<=yDiv-1;yBase++)
    {

```

```

        TestLaplacianFace[xBase][yBase]=TestFace[xBase][yBase]-
        FaceTemplate[xBase][yBase];
    }

}

MinLaplacianDiff=2147483647; //2^32
MinLaplacianIndex=-1;
for(i=1;i<=NumFaces;i++)
{
    TotalLaplacianDiff=0;
    for(xBase=0;xBase<=xDiv-1;xBase++)
    {
        for(yBase=0;yBase<=yDiv-1;yBase++)
        {
            TotalLaplacianDiff=TotalLaplacianDiff+java.lang.Math.abs(TestLaplacia
nFace[xBase][yBase]-LaplacianFaces[xBase][yBase][i]);
        }
    }

    if(MinLaplacianDiff>TotalLaplacianDiff)
    {
        MinLaplacianDiff=TotalLaplacianDiff;
    }
}

```

```

        MinLaplacianIndex=i;
    }

    }

    pgm2.writeImage();

    PGM pgmMatched=new PGM();
    pgmMatched.setFilePath(FaceFileNames[MinLaplacianIndex]);
    pgmMatched.readImage();
    pgmMatched.setFilePath("matched.pgm");
    pgmMatched.writeImage();
    drawImage(frmlImage2.getGraphics(),"matched.pgm");
    addResultText("done.");
    addResultText("\n\nMatched:
"+FaceFileNames[MinLaplacianIndex]);
    }

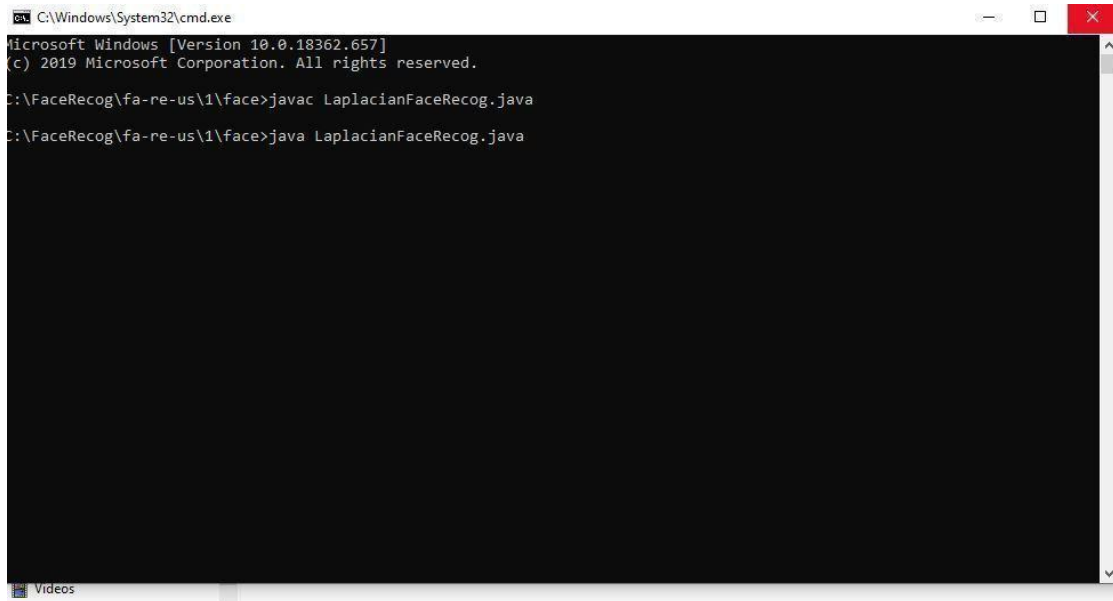
}

```

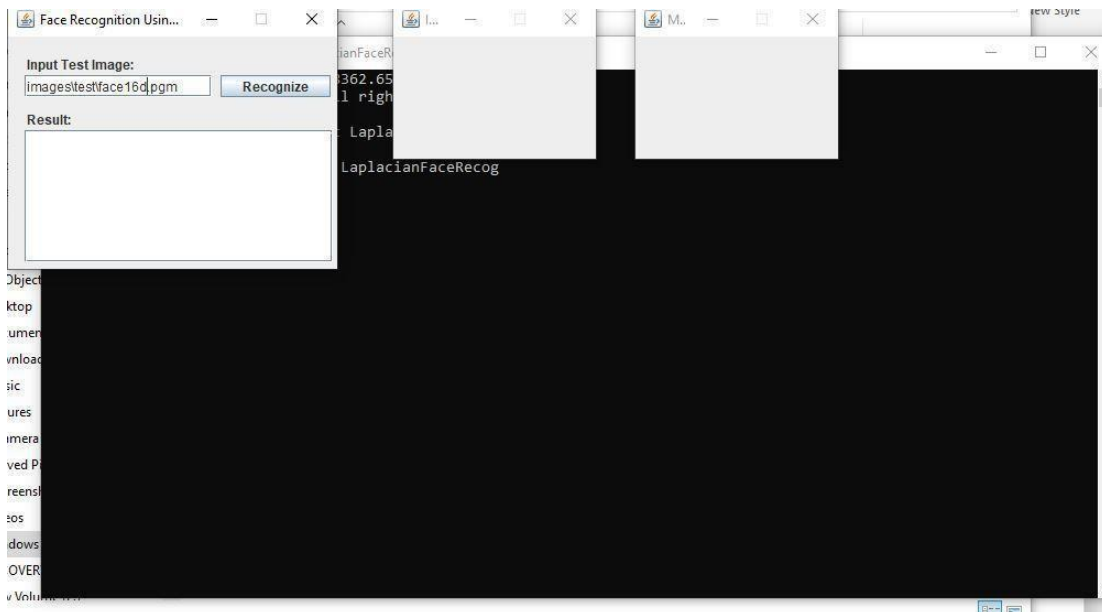
```
public static void main(String args[])
{
    new LaplacianFaceRecog();
}
}
```

E. SCREEN SHOTS

TO RUN THE CODE



Start the application as above.



In this the data base consists of train and test pictures data .In this the pictures stored in data base are train data .The pictures which are used for testing are stored in test data. Therefore the data which we used for testing is called test data.

F. PUBLICATION

FACE RECOGNITION BY LAPLACIANFACES

Paper Acceptance 

 **NCCIDS 2020** <nccids2020@gmail.com>
to saisanthosh1818, me, Sathyabama ▾

Tue, Mar 10, 5:13 PM ☆ ↶ ⋮

Dear Author(s),

Paper Title: FACE RECOGNITION USING LAPLACIANFACES
Authors: Sai Santhosh Thallam, Syed Abdul and R Sathyabama Krishna.
paper Id:137

We have completed our review process for the submissions to NCCIDS2020.

We are pleased to inform that your above mentioned article has been accepted for oral presentation in the NCCIDS2020 conference. You are further advised to send the full paper according to the template attached here with. Also you are expected to check your paper for plagiarism. The abstracts of all accepted papers will be published in the conference Proceedings. Authors of selected outstanding high-quality papers will be invited to submit extended versions of their papers for publication in the Scopus indexed journals at additional cost.

Registration Fees:
Registration Fee

UG/PG Students/Research Scholars : 1500/-*
Academicians/Industrial participants : 2000/-*

(Conference Proceeding with ISBN)

(DD should be drawn in the favour of "Sathyabama Institute of Science and Technology" payable at Chennai)

Herewith attached Registration Form, Paper Template, Brochure and copyright agreement. You are requested to send the Camera ready paper (CD) along with Registration form, copyright agreement & Demand Draft to the below mentioned address at the earliest.