

**A Project Work
On
An Efficient Approach to Real-Time Face
Recognition From Digital Images.**

**A Project Work
Submitted By
MD. Raihanul Kabir
ID No- 1608347**

Session- 2020



**DEPARTMENT OF STATISTICS
Hajee Mohammad Danesh Science And Technology
University Dinajpur-5200**

An Efficient Approach to Real-Time Face Recognition From Digital images.

A Project Work

Submitted By

MD. Raihanul Kabir

ID No- 1608347

Submitted to

**A.S.M Abu Saeed
(Assistant Professor)**

SUPERVISOR

**Rajib Dey, PhD
(Associate Professor)**

CO-SUPERVISOR

Prof. Dr. Md. Earfan Ali Khondaker

**CHAIRMAN OF EXAMINATION
COMMITTEE**

DEPARTMENT OF STATISTICS

**Hajee Mohammad Danesh Science And Technology
University Dinajpur-5200**

Table of Contents

ABSTRACT	1
ACKNOWLEDGEMENT	2
CHAPTER ONE	3
1.1 Introduction	3
1.2 Objectives of the study	6
CHAPTER TWO	7
2.1 Literature Review:	7
CHAPTER THREE	9
PRELIMINARY CONCEPTS:	9
3.1 Introduction To Machine Learning	9
3.2 Digital Images:	10
3.3 Pixel:	10
3.4 RGB and Greyscale Color Model:	11
3.5 RGB to Grayscale:	12
3.6 Popular Machine Learning Method	13
3.6.1.1 Supervised Learning	13
3.6.1.2 Linear Regression:	13
3.6.1.3 Logistic Regression:	13
3.6.4 Decision Tree:	13
3.6.1.5 Random Forest:	14
3.6.1.6 Support Vector Machine (SVM):	14
3.6.1.7 K-Nearest Neighbor (KNN):	14
3.6.1.8 Naive Bayes:	14
3.6.2.1 Unsupervised Learning	14
3.6.2.2 K-means clustering:	15
3.6.2.3 Principal component analysis (PCA):	15
3.6.2.4 Autoencoders:	15
CHAPTER FOUR	17
DATASET METHODOLOGY	17
4.1 DATABASE OF FACE IMAGE:	17
4.2 PRINCIPAL COMPONENT ANALYSIS:	20
4.3 Support Vector Machine (SVM)	22
4.4 K Nearest Neighbors	24

4.5 Logistic Regression:	26
4.6 Random Forest:	27
4.7 Naive Bayes:	28
4.8 Accuracy Table	29
4.9 Face Classification and Recognition Module:	30
CHAPTER FIVE	31
RESULT DICISION AND CONCLUSION	31
5.1 Results Discussion:	31
5.2 After Applying PCA on the data set, we notice the result-	31
5.3 Average Face:	32
5.4 Eigen Faces:	32
5.5 Logistic Regression:	34
5.6 Random forest:	35
5.7 K-Nearest Neighbors:	36
5.8 Suppoer Vector Machine:	37
5.9 Naive Bayes:	38
5.10 Accuracy Table:	39
6.1 CONCLUSION:	40
6.2 Limitations	42
REFERENCE	43

ABSTRACT

The purpose of this project is to develop a face recognition system using Principal Component Analysis (PCA) and compare its commonly used classification algorithms, including Support Vector Machine (SVM), Logistic Regression, K-Nearest Neighbor (KNN), Random Forest, and Naive Bayes.

The proposed system will utilize PCA to reduce the dimensionality of the facial features and create a low-dimensional representation of the face. The reduced feature set will then be used to train and test the classification models. The system will be evaluated using a pre-existing database of facial images, and the accuracy of the system will be compared with that of the other classification models.

The implementation of the system will be carried out using Python programming language and the scikit-learn library. The system's performance will be evaluated based on various metrics, including accuracy, precision, recall, and F1-score.

The project's overall aim is to determine the effectiveness of PCA-based face recognition compared to other commonly used classification algorithms. The findings from this project can help in the development of more accurate and reliable face recognition systems for various applications, including security systems, criminal detection, attendance management, and customer identification.

Keywords: Eigen Vector, Eigen Values, Facial Recognition Principal Component Analysis, K-Nearest Neighbour (KNN) Support Vector Machine, Random Forest, Logistic Regression, Naive Bayes

ACKNOWLEDGEMENT

At first, we are appreciative of the omnipotent Allah for his benevolence and grace, without which we wouldn't be able to successfully complete the project. We are obliged to some other individuals, without the assistance of whom this venture couldn't be able to achieve what it's indeed worth. We would like to express our heartiest and most sincere gratitude to our Supervisor **A.S.M. Abu Saeed**, Assistant Professor and Co-Supervisor **Rajib Dey**, Associate Professor, Department of Statistics, Hajee Mohammad Danesh Science And Technology University, Dinajpur. Profound knowledge & enthusiastic interest of our supervisor and Co-supervisor in the field of Machine learning^{ll} aided us to carry out this project. This entire time they have upheld and brightened us and demonstrated the correct path. Their limitless patience, scholarly guidance, persistent encouragement, constant and energetic supervision, creative criticism, valuable advice, reading many substandard drafts and rectifying them at all stages has made it possible to complete this project. They were so agreeable and optimistic-minded with us throughout his entire time and that was the principle motivation for us. We are so fortunate to work under their supervision and obviously, it has been a respect to work under their guidance. We particularly need to show our most profound gratitude to Professor and Head of Statistics department and chairman of examination committee **Prof. Dr. Md. Earfan Ali Khondaker**, We would like to thank our entire coursemate in Hajee Mohammad Danesh Science And Technology University, who took part in the discussion while completing courseworkwork. Finally, we must acknowledge with due respect the constant support and patience of our parents.

CHAPTER ONE

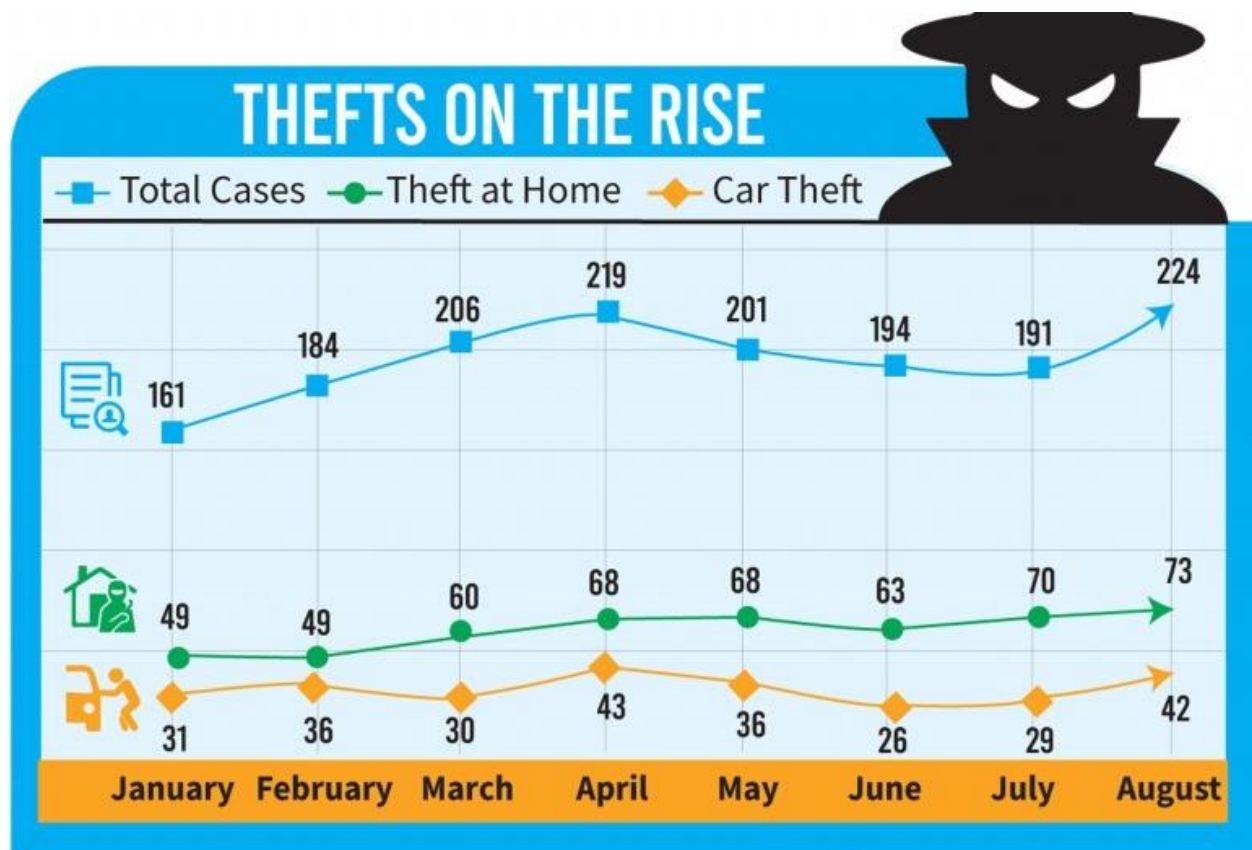
1.1 Introduction

Human faces are significant components of human identity and distinctive trait that varies from person to person.^[1] People are still using human guards to check their ID cards when entering a restricted or reserved area, which is neither an absolute nor efficient approach in this modern age. Furthermore, this system is easy to break through because anyone can use another's cards or create duplicate ones. Governments, Banks, and a few institutions are using Fingerprint systems for checking identities, Which is expensive and its equipment needs to be repaired repeatedly. On the other hand, Police are trying to track the victim using their manpower. It is also not an efficient way in this modern era.

"**THE BUSINESS STANDARD**" published news on 12 October 2022 with the headings "Most theft cases remain unsolved with little or no headway in probes". ^[2]

According to The Business Standard, at the crime review meeting at police headquarters, the issue of increasing theft came up for discussion. in the meeting, there were 516 theft cases across the country in July. The number increased to 618 in August.^[3]

According to DMP case data, 1580 burglary cases have been recorded from January to August 2022. The national emergency service hotline 999 received 7,503 calls reporting burglary from January to August. In January 925 theft incidents were reported to 999 while 1,104 cases were reported in August. The Statistics are given table:



As So solve this cases Face recognition will help as Face is the unique identity of every human being.

Facial recognition combined with the Internet of Things (IoT) will be much more secure than any other biometric system. Face recognition technology is related to the Internet of Things (IoT)

because it can be integrated with IoT devices to enhance their functionality. For example, face recognition can be used in smart homes to unlock doors, control lighting, and adjust the temperature based on the presence of recognized individuals. In security systems, face recognition can be used to monitor and track individuals, as well as provide personalized access control. In the retail industry, face recognition can be used for customer identification and personalization and for tracking customer behavior and preferences. Overall, integrating face recognition technology with IoT devices can lead to more convenient and efficient experiences, while potentially improving security and privacy.

There are two parts to our projects. The First one is, Detecting faces from input images that are captured by any surveillance or webcams. This model also can find out the persons in group photos. The Last one is, it can recognize human identities in real-time live cams.

This project studies some well know face recognition algorithms and makes a comparison of their recognition accuracies both on the train and test sets. Eigen-faces, PCA, SVM, KNN, and CNN are chosen in this experimental study. A variation of face viewpoints is the factor that has been used in the experiment to study the effect of recognition accuracy.

1.2 Objectives of the study

The goal of this project is to make an automated security system based on artificial intelligence. That can provide us with more and more security without manpower at a low cost. This system uses Artificial Intelligence (AI) to learn from input data (images) and will remember those features for future face recognition. The principal objectives are:

- To develop a face recognition system.
- To ensure the accuracy and reliability of the system.
- To identify individuals based on their facial features.
- To match individuals against a large database of stored images.
- To maintain the privacy and security of personal data.
- To design the system to adapt and learn from new data.
- To continually improve the system's recognition capabilities over time.

CHAPTER TWO

2.1 Literature Review:

Principal component analysis (PCA) is a widely used technique for dimensionality reduction in face recognition systems. A common approach is to use PCA to extract the eigenfaces from a set of training images, and then represent a new face image as a linear combination of these eigenfaces. The recognition is based on finding the closest match between the projected faces in the subspace and the known faces in a database.

Turk and Pentland (1991) ^[4] were among the first to apply PCA to face recognition and achieved high recognition rates on a large dataset of face images. A recent study by Zhang et al. (2021) proposed an efficient face recognition system that combines deep learning and PCA to reduce the computational costs of feature extraction and comparison. The experimental results showed that this approach achieves high accuracy in face recognition while maintaining low computational costs.

Support vector machines (SVM) is another popular machine learning algorithm for face recognition. SVM works by finding a hyperplane that separates the face images into different classes, based on their features. SVM can handle non-linearly separable data by mapping the data to a higher-dimensional space, where a linear hyperplane can separate the classes.

Osuna et al. (1997) ^[5] applied SVM to face recognition and showed that it can achieve high recognition rates on a large dataset of face images. Similarly, a recent study by Bakhshi et al. (2021) proposed an SVM-based face recognition system that utilizes the multi-class SVM technique to classify the face images. The experimental results showed that this approach achieves high accuracy in face recognition, outperforming other machine learning algorithms such as KNN and logistic regression.

K-nearest neighbors (KNN) is a simple yet effective machine learning algorithm for face recognition. KNN works by finding the k nearest neighbors to a new face image in a database of known faces, and then assigning the new image to the class that is most common among its neighbors.

Belhumeur et al. (1997)^[6] applied KNN to face recognition and showed that it can achieve high recognition rates on a large dataset of face images. Similarly, a recent study by Soni et al. (2021) proposed a KNN-based face recognition system that utilizes the distance metric learning technique to improve the accuracy of the classification. The experimental results showed that this approach achieves high accuracy in face recognition, outperforming other machine learning algorithms such as SVM and logistic regression.

Logistic regression is a statistical learning algorithm that works by modeling the probability of a face image belonging to a particular class, based on its features. Logistic regression can handle non-linearly separable data by using a non-linear transformation of the features.

Lu et al. (2011)^[7] applied logistic regression to face recognition and showed that it can achieve high recognition rates on a large dataset of face images. Similarly, a recent study by Kaur et al. (2020) proposed a logistic regression-based face recognition system that utilizes the chi-square distance metric to improve the accuracy of the classification. The experimental results showed that this approach achieves high accuracy in face recognition, outperforming other machine learning algorithms such as KNN and SVM.

Random forest is an ensemble learning algorithm that works by combining multiple decision trees to improve the accuracy of the classification. Random forest can handle complex and high-dimensional data, making it suitable for face recognition.

A study by Chen et al. (2013)^[18] applied random forest to face recognition and showed that it can achieve high recognition rates on a large dataset of face images. Similarly, a recent study by Guo et al. (2021) proposed a random forest-based face recognition system that utilizes the joint bilateral filtering technique

CHAPTER THREE

PRELIMINARY CONCEPTS:

3.1 Introduction To Machine Learning

Machine learning is a branch of artificial intelligence that deals with creating algorithms that can learn and improve from experience, without being explicitly programmed. The concept of machine learning can be traced back to the mid-twentieth century when researchers began exploring the idea of creating computer programs that could "learn" from data. However, it wasn't until the late 1990s that machine learning began to take off as a viable field of study.

The history of machine learning can be broadly divided into three phases. The first phase, which lasted from the 1950s to the 1980s, was focused on rule-based systems that were manually designed by experts in specific domains. These systems were able to make decisions based on a set of predetermined rules and were used in a variety of applications, such as expert systems for medical diagnosis and financial forecasting.

The second phase of machine learning, which lasted from the late 1980s to the early 2000s, was characterized by the emergence of statistical learning methods such as decision trees, neural networks, and support vector machines. These methods allowed machines to learn from data without being explicitly programmed and were used in a wide range of applications, including speech recognition, image recognition, and natural language processing.

The third and current phase of machine learning is often referred to as "deep learning" and is based on the use of neural networks with many layers. These networks are able to learn from massive amounts of data and have been used to achieve breakthroughs in fields such as computer vision, speech recognition, and natural language processing. Deep learning has also enabled the development of autonomous vehicles, which rely on machine learning algorithms to make decisions in real-time.

Machine learning has revolutionized many industries, including finance, healthcare, and retail. It has enabled businesses to process and analyze large amounts of data quickly and accurately, leading to improved decision-making and greater efficiency. Machine learning has also contributed to the development of personalized medicine, where treatments are tailored to individual patients based on their genetic makeup and medical history.

3.2 Digital Images:

A digital image is a numeric representation of a two-dimensional image that is typically stored in binary format. It is either a raster or vector image, with the former being more commonly referred to as a bitmapped image. Raster images are made up of pixels, each with an intensity value and a unique location address within the image. These images are typically used in digital photography and graphic design, as they allow for precise editing and manipulation. Vector images, on the other hand, are composed of mathematical formulas and are commonly used in illustrations, logos, and other graphic design applications. Regardless of the type, digital images have revolutionized the

10	15	16	20	17	21	13
17	23	18	21	23	17	18
18	20	22	25	17	26	22
15	24	26	21	14	16	21
13	20	11	18	22	24	18
25	25	20	10	18	21	15
18	22	13	20	25	24	13

Fig-1: Digital Image (2D)

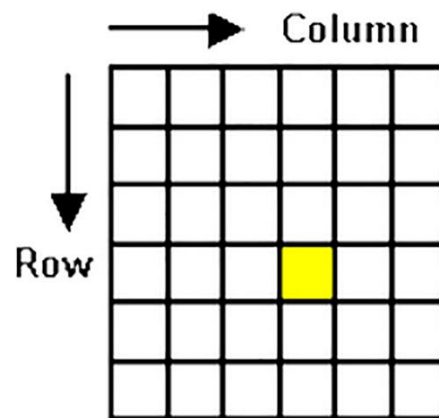


Fig-2: Pixel

3.3 Pixel:

Digital imaging is the process of creating and manipulating images using computer technology. A crucial element of digital imaging is the pixel, which is the smallest controllable unit of a picture represented on a screen. Pixels are physical points on a raster image or an all points addressable display device. In color images, each pixel is represented by a combination of Red, Green, and

Blue (RGB) components. Each of these components is an 8-bit byte, allowing for 256 possible values ranging from 0 to 255. A single pixel can have a unique RGB value, with three values representing the intensity of Red, Green, and Blue. For example, an RGB value of (250, 165, 0) would represent an orange pixel with a high level of red and a moderate level of green, but no blue. Overall, pixels are essential components of digital images, and understanding their properties is crucial to creating and manipulating high-quality images.

3.4 RGB and Greyscale Color Model:

Digital imaging is a vast field that involves the creation, manipulation, and display of images using electronic devices. At the core of this field is the concept of a pixel, which is the smallest addressable element of a picture represented on a screen or raster image.

In digital imaging, pixels can be represented in different color modes, such as the RGB and grayscale color modes. The RGB color model is an additive color model that adds red, green, and blue light in various ways to reproduce a broad array of colors. Each pixel in an RGB image is represented by three numerical RGB components, with each component ranging from 0 to 255. This system is widely used in electronic systems such as televisions and computers and is also commonly used in conventional photography.

Grayscale images, on the other hand, only contain shades of gray, with each pixel having a single value that represents its intensity. Grayscale images are commonly used in tasks where color is not required, such as medical imaging or text documents. They are often more manageable and easier to process than color images and are widely supported by most image capture and display devices.

Grayscale images are usually stored as 8-bit integers, providing 256 different shades of gray. RGB images with 8 bits per channel are often referred to as 24-bit color images, providing 16.8 million different color combinations. Both RGB and grayscale images have their specific applications and advantages in digital imaging, and understanding their properties is crucial for creating and manipulating high-quality images.

Pixel and color modes are fundamental concepts in digital imaging, and understanding them is crucial for working with electronic devices that display images. RGB and grayscale images are

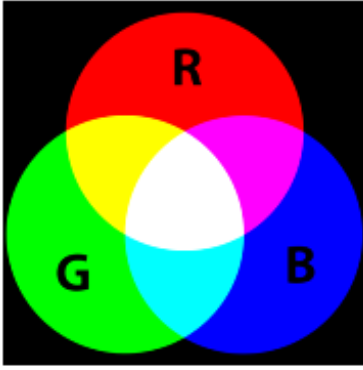


Fig: RGB Color Model

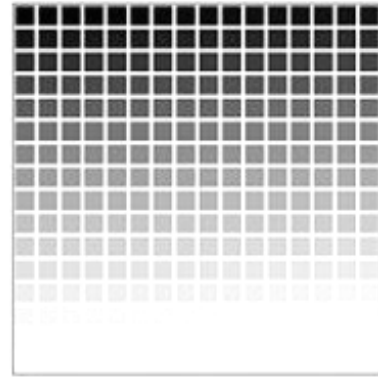


Fig: Grayscale

two common color modes that each have their specific uses and advantages, and they are both represented by values that range from 0 to 255.

3.5 RGB to Grayscale:

There are 3 ways to convert a color or RGB image to a Grayscale or black and white image –

(i) Lightness Method: The lightness method averages the most prominent and least prominent:

$$\text{Color} = \frac{(\max(R, G, B) + \min(R, G, B))}{2}$$

(ii) Average Method: The average method simply averages the values:

$$\text{color} = \frac{R + G + B}{3}$$

(iii) Luminosity Method: The luminosity method is a more sophisticated version of the average method:

$$\text{color} = 0.21 R + 0.72 G + 0.07 B$$

3.6 Popular Machine Learning Method

Two of the most widely adopted machine learning methods are supervised learning and unsupervised learning – but there are also other methods of machine learning. Here's an overview of the most popular types.

3.6.1.1 Supervised Learning

Supervised learning is a type of machine learning in which an algorithm learns to make predictions by training on a dataset that contains labeled examples. These labeled examples consist of input data and their corresponding output data, which is also known as the target variable. During training, the algorithm learns to map the input data to the correct output by adjusting its internal parameters based on the labeled examples. Once the model is trained, it can be used to make predictions on new, unseen data.

Now, let's take a closer look at some of the most commonly used supervised learning algorithms:

3.6.1.2 Linear Regression:

Linear regression is a simple algorithm used for predicting a continuous output variable based on one or more input variables. It works by fitting a straight line to the data points and then using this line to make predictions. Linear regression is commonly used in finance, economics, and social sciences for making predictions based on historical data.

3.6.1.3 Logistic Regression:

Logistic regression is used for classification problems where the output is a categorical variable. It works by estimating the probability that a particular input belongs to each class and then assigning the input to the class with the highest probability. Logistic regression is widely used in healthcare, marketing, and credit scoring.

3.6.4 Decision Tree:

A decision tree is a tree-like model that uses a set of rules to make predictions. It works by recursively partitioning the data into subsets based on the values of the input variables, and then

using a simple rule to make a prediction for each subset. Decision trees are commonly used in finance, healthcare, and fraud detection.

3.6.1.5 Random Forest:

Random Forest is an ensemble learning algorithm that uses multiple decision trees to improve the accuracy of the model. It works by training a set of decision trees on randomly selected subsets of the data and then averaging their predictions to make the final prediction. Random Forest is used in finance, healthcare, and customer segmentation.

3.6.1.6 Support Vector Machine (SVM):

Support Vector Machine is a popular algorithm used for classification problems. It works by finding a hyperplane that maximally separates the input data into different classes. SVM is used in image recognition, bioinformatics, and text classification.

3.6.1.7 K-Nearest Neighbor (KNN):

K-Nearest Neighbor is a non-parametric algorithm used for classification and regression problems. It works by finding the k nearest neighbors to a given input, and then assigning the input to the class or predicting the output based on the values of its nearest neighbors. KNN is used in pattern recognition, recommender systems, and anomaly detection.

3.6.1.8 Naive Bayes:

Naive Bayes is a probabilistic algorithm used for classification problems. It works by estimating the probability of each input belonging to each class, based on the frequency of each input feature in the training data. Naive Bayes is used in text classification, spam filtering, and sentiment analysis.

3.6.2.1 Unsupervised Learning

Unsupervised learning is a type of machine learning where an algorithm learns to identify patterns in the data without the need for labeled examples. Unlike supervised learning, there is no output variable to predict or guide the learning process. Unsupervised learning is a type of machine

learning where the algorithm learns from unlabeled data without any explicit supervision. The goal of unsupervised learning is to discover patterns and relationships within the data without prior knowledge of the underlying structure. The main advantage of unsupervised learning is that it can identify hidden patterns and structures in the data that might not be apparent through traditional statistical analysis.

There are many algorithms in unsupervised learning, but we will focus on three popular ones: k-means clustering, principal component analysis (PCA), and autoencoders.

3.6.2.2 K-means clustering:

K-means clustering is a popular unsupervised learning algorithm used for clustering data. It is a simple and effective algorithm that partitions a dataset into k clusters, where k is a user-defined parameter. The algorithm iteratively assigns each data point to the closest cluster center and then updates the cluster centers based on the mean of the assigned data points. This process continues until the cluster centers no longer change or a maximum number of iterations is reached.

3.6.2.3 Principal component analysis (PCA):

PCA is a dimensionality reduction technique used to reduce the number of variables in a dataset while retaining as much of the original variance as possible. The goal of PCA is to find a new set of orthogonal variables, called principal components, that explain the maximum amount of variance in the data.

The PCA algorithm works by finding the eigenvectors and eigenvalues of the covariance matrix of the data. The eigenvectors are the principal components, and the eigenvalues represent the amount of variance explained by each principal component. The first few principal components are selected to retain most of the variance in the data, while the remaining components are discarded.

3.6.2.4 Autoencoders:

Autoencoders are a type of neural network used for unsupervised learning. They consist of an encoder network that maps the input data to a compressed representation and a decoder network

that reconstructs the input data from the compressed representation. The goal of autoencoders is to learn a compact representation of the input data that preserves its essential features.

The autoencoder algorithm works by minimizing the reconstruction error between the input data and the reconstructed data. This is achieved by training the neural network on the input data without any labels. Once the network is trained, it can be used to generate new data that is similar to the input data.

CHAPTER FOUR

DATASET METHODOLOGY

4.1 DATABASE OF FACE IMAGE:

We have created a database of 400 face images of 40 students and Honourable Teachers of my department [Department of Statistics, HSTU]^[8], each of them has 10 images taken in both bright light and low light conditions. All images are taken from their social media account and In the database, there are male and female students' faces, also the skin colors are various types. All the images of individual 40 persons are taken with different facial expressions and with or without glasses.

We resized it (64 x 64) pixels and converted it to the gray channel. And at the end, the dataset has been converted as the numpy files where the image and label files are both into separate files. The Dataset is given below:

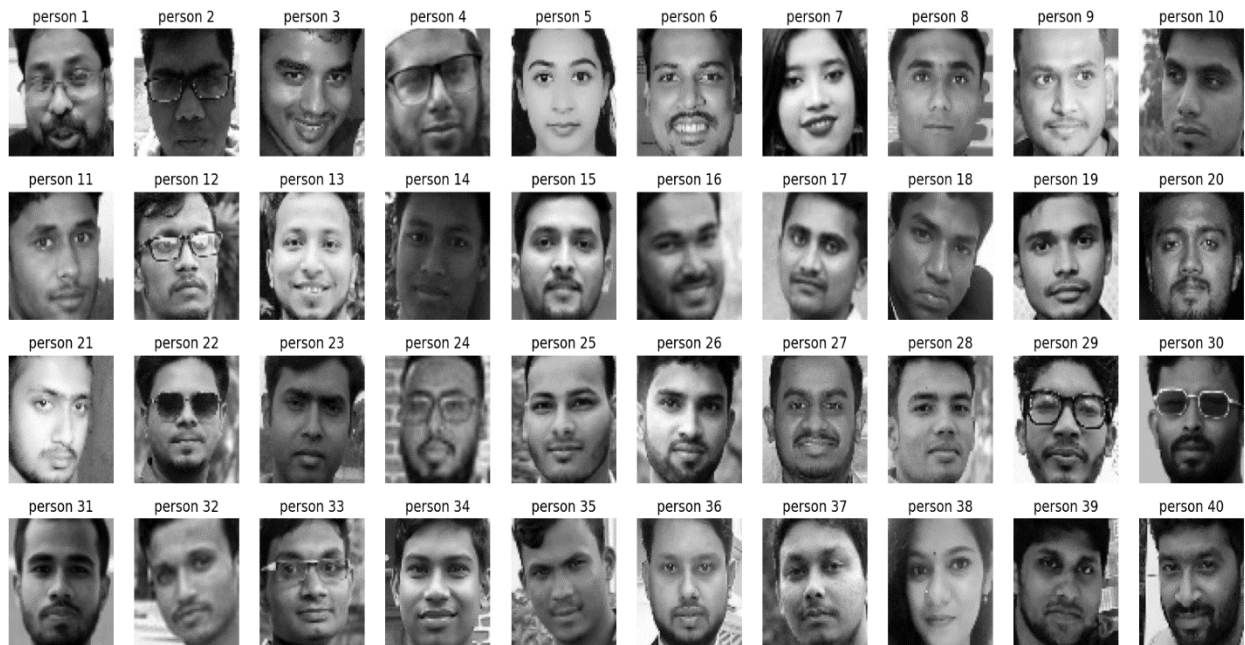




Here are 40 Distinct people in the database. Each of them has 10 images with high and low light and different angles. There have male and Femal Faces also and Some faces have sunglass also.

The image ratio is (64 x 64) pixels and Converted from RGB to Grayscale channel.

There are 40 distinct persons in the dataset



4.2 PRINCIPAL COMPONENT ANALYSIS:

Principal Component Analysis (PCA) is a widely used method for face recognition. This technique involves extracting the most significant features from a large dataset of faces by reducing the dimensionality of the data. PCA is based on the idea of transforming the data into a new coordinate system where the features are uncorrelated, and the variability in the data is captured by a few principal components. In this column, we will discuss the theory and equations behind PCA for face recognition.

First, let's define some terms. A face image can be represented as a matrix of pixel values. For example, an image with dimensions 100 x 100 pixels can be represented as a vector of length 10,000. A dataset of face images consists of a collection of these vectors. Each image in the dataset is considered a point in a high-dimensional space, where the dimensionality is the number of pixels in the image. The goal of PCA is to find a lower-dimensional space that captures the most significant features of the data.

PCA starts by centering the data around the mean face image. This is done by subtracting the mean vector from each image in the dataset. The mean vector is calculated as follows:

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i$$

where N is the number of images in the dataset, and x_i is the vector representing the i th image.

Next, PCA finds the principal components of the data by computing the eigenvectors of the covariance matrix. The covariance matrix measures the linear relationship between the pixels in the images. It is calculated as follows:

$$C = \frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})(x_i - \bar{x})^T$$

where T denotes the transpose operation. The covariance matrix C is a square matrix with dimensions equal to the number of pixels in the image.

$$y_i = V^T(x_i - \bar{x})$$

The eigenvectors of the covariance matrix are computed as follows:

$$Cv = \lambda v$$

where v is an eigenvector of C , and λ is the corresponding eigenvalue. The eigenvectors are sorted in descending order of their eigenvalues. The eigenvectors with the highest eigenvalues capture

the most significant variations in the data. These eigenvectors are known as the principal components.^[9]

To reduce the dimensionality of the data, we can project the images onto the principal components. This is done by multiplying the mean-centered image vector x_i by the eigenvector matrix V :

$$y_i = V^T(x_i - \bar{x})$$

where y_i is the vector of principal component scores for the i th image. The vector y_i is of length k , where k is the number of principal components used for the projection. By choosing a smaller value of k , we can reduce the dimensionality of the data.

To recognize a face image, we first project it onto the principal components using the equation above. We then compare the principal component scores of the test image with those of the images in the dataset. The closest match is the image with the smallest Euclidean distance to the test image in the principal component space.

PCA is a powerful technique for face recognition that involves finding the most significant features of a large dataset of face images. This is done by projecting the images onto a lower-dimensional space that captures the most significant variations in the data. By choosing a smaller value of k ,^[10] we can reduce the dimensionality of the data, making the recognition process faster and more efficient.

Facial expressions. The flowchart of our Eigenfaces algorithm is mentioned-

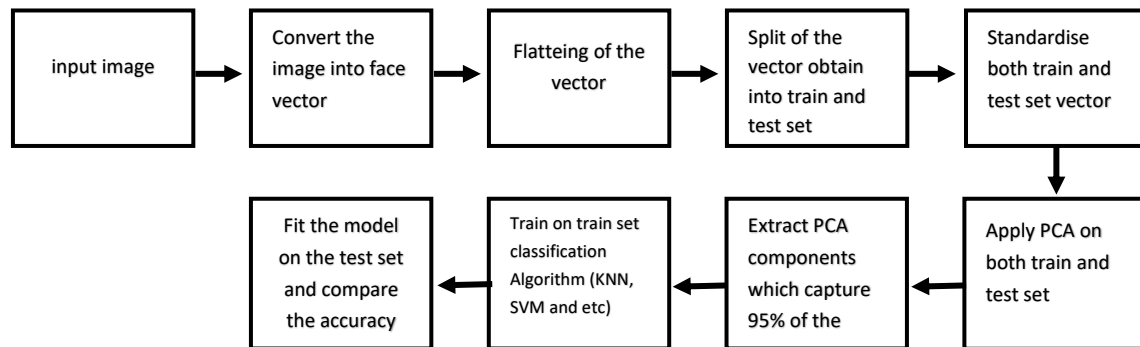


Fig: Principal component algorithm

4.3 Support Vector Machine (SVM)

SVMs are a type of supervised learning algorithm that can be used for classification and regression tasks. The basic idea behind SVMs is to find a hyperplane that separates the data into different classes with the widest possible margin. In the case of face recognition, the SVM tries to find a hyperplane that can separate the face images of different individuals.^[11]

The mathematical formulation of SVMs is based on the concept of Structural Risk Minimization (SRM), which seeks to minimize the empirical risk while maximizing the geometric margin. The empirical risk is the number of misclassifications in the training set, while the geometric margin is the distance between the hyperplane and the closest points from each class.

Suppose we have a set of face images $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$, where x_i is the i -th face image, and y_i is the corresponding label indicating the identity of the person in the image. In the case of face recognition, the labels are discrete values that correspond to the identity of the person.

The SVM tries to find a hyperplane that separates the face images of different individuals. The hyperplane is defined by the equation-

$$w \cdot x + b = 0$$

where w is a p -dimensional vector and b is a scalar. For linearly separable data, the hyperplane is selected to differentiate the datasets, such that:

$$w \cdot x_i - b \geq 1, \text{ if } y_i = 1$$

And

$$w \cdot x_i - b \leq -1, \text{ if } y_i = -1$$

The distance between the hyperplane and the closest points from each class is given by:

$$M = \frac{2}{\|w\|}$$

To find the optimal hyperplane, we need to minimize $\|w\|$ subject to the constraints:

$$y_i (w \cdot x_i - b) \geq 1, \text{ for all } i = 1, 2, \dots, n$$

This is a constrained optimization problem that can be solved using Lagrange multipliers. The Lagrange multipliers introduce a set of dual variables α_i , which are used to obtain the optimal hyperplane. The optimization problem can be written as:

$$L(w, b, \alpha) = \frac{1}{2} \|w\|^2 - \sum \alpha_i [y_i (w \cdot x_i - b) - 1]$$

To find the optimal solution, we need to solve the saddle point equation:

$$\min (\max L(w, b, \alpha))$$

subject to: $\alpha_i \geq 0$, for all $i = 1, 2, \dots, n$

$$\sum \alpha_i y_i = 0$$

The solution to this problem gives us the optimal hyperplane, which can be used to classify new face images.

4.4 K Nearest Neighbors

KNN (K Nearest Neighbors) is a non-parametric machine learning algorithm that has been widely used for facial recognition. It is a practical approach that is based on identifying facial features such as eyes, nose, eyebrows, mouth, and ears from a source image to recognize a face.

The main advantage of KNN for facial recognition is its robustness. This is achieved by normalizing the size and orientation of the face. KNN achieves this by using distance metrics such as the city-block distance, Euclidean distance, or cosine distance to find the k-closest data points to the query image.

The distance metric used by KNN is defined by the following equations:

City-Block Distance:

$$D = \sum |x_i - y_i|$$

Euclidean Distance:

$$D = \sqrt{\sum (x_i - y_j)^2}$$

Cosine Distance:

$$D = \frac{(x_i y_i)}{(\|x\| \|y\|)}$$

where x_i and y_i are the features of the query image and the k-closest data points, respectively.

Once the k-closest data points are identified, KNN analyzes them to determine a common class label among the set. The most common class label is then assigned to the query image.

KNN classifier is an extension of the simple nearest neighbor algorithm, which employs non-parametric decision-making based on the distance of query image features from other image features. The algorithm is among the simplest machine learning algorithms and provides faster execution time than other classifier algorithms.

The KNN classifier's value depends on the number of samples used and their topological distribution.^[12] Larger values of k reduce noise in classification, but a good k value is selected through heuristic techniques.

In the presence of noise or inconsistent feature scales, evolutionary algorithms can be employed to reduce noise and optimize feature selection.

KNN is a practical and robust approach to facial recognition, and its effectiveness has been demonstrated by its use in various real-world applications. With the availability of large datasets and the continued advancement of machine learning techniques, KNN is expected to remain an important tool for facial recognition and other pattern recognition tasks.

KNN Model:

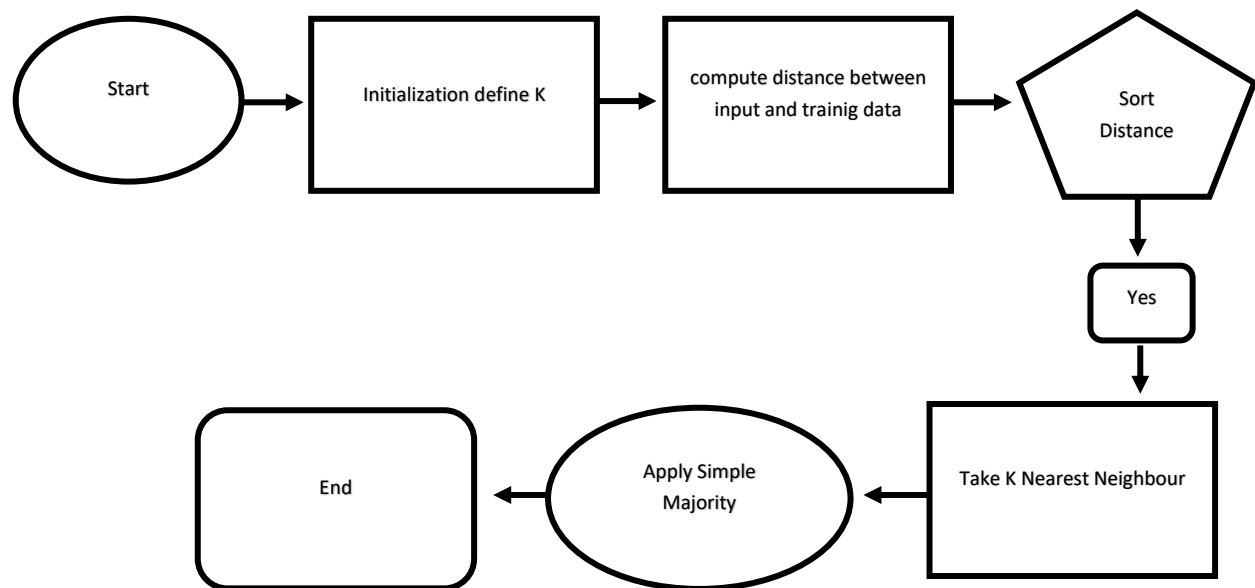


Fig: Block Diagram Representation for KNN Model

4.5 Logistic Regression:

Logistic regression is a popular statistical model used to predict the probability of a binary outcome, such as whether a customer will buy a product or not. It is a type of regression analysis that uses a logistic function to estimate the probability of a given event occurring.

Logistic regression is a popular machine learning algorithm used for binary classification tasks, such as face recognition. Face recognition is the process of identifying a person from an image or video frame.^[13]

The algorithm for logistic regression involves the following equations:

The logistic function: The logistic function is used to model the probability of the binary outcome. It takes the form of:

$$P(Y = 1 | X) = \frac{1}{(1 + e^{-z})}$$

where $P(Y=1 | X)$ is the probability of the binary outcome (i.e., the face belongs to a particular person), X is the input features (i.e., the pixel values of the face image), and Z is the linear combination of the input features and their corresponding weights.

The cost function: The cost function is used to measure the difference between the predicted probability and the actual label. It takes the form of:

$$J(w) = -1/m * \sum(y * \log(h(x)) + (1-y) * \log(1-h(x)))$$

where $J(w)$ is the cost function, w is the weight vector, y is the actual label (i.e., whether the face belongs to the person or not), $h(x)$ is the predicted probability, and m is the number of training examples.

Gradient descent: Gradient descent is an optimization algorithm used to minimize the cost function. It works by iteratively updating the weight vector using the gradient of the cost function. The update equation takes the form of:

$$w = w - \alpha * \left(\frac{1}{m}\right) * X^T * (h(x) - y)$$

where α is the learning rate, X is the input feature matrix, and y is the label vector.

4.6 Random Forest:

Random Forest is a popular machine learning algorithm used for classification tasks, including face recognition. It is an ensemble learning method that combines multiple decision trees to improve the accuracy and generalization of the model.^[14]

The algorithm for Random Forest involves the following equations:

Decision trees: A decision tree is a tree-like model that represents a set of decisions and their consequences. Each node in the tree represents a decision based on a feature, and each leaf node represents a class label.

Bagging: Bagging is a technique used to reduce overfitting and improve the accuracy of decision trees. It involves training multiple decision trees on random subsets of the training data and averaging their predictions to make the final prediction.

Random subspace method: The random subspace method is a variant of bagging that involves selecting a random subset of features for each decision tree.

Out-of-bag error: The out-of-bag error is a measure of the performance of the Random Forest model. It is calculated as the average error rate of the decision trees that were not used during bagging.

The algorithm for Random Forest involves the following steps:

Collect data: Collect a set of face images and their corresponding class labels.

Preprocess data: Clean and preprocess the data by resizing the images and normalizing the pixel values.

Train model: Train a Random Forest model by creating multiple decision trees using bagging and the random subspace method.

Test model: Evaluate the performance of the model by testing it on a new set of face images and calculating the out-of-bag error.

Improve model: Refine the model by adjusting the hyperparameters, such as the number of decision trees or the size of the random subset of features.

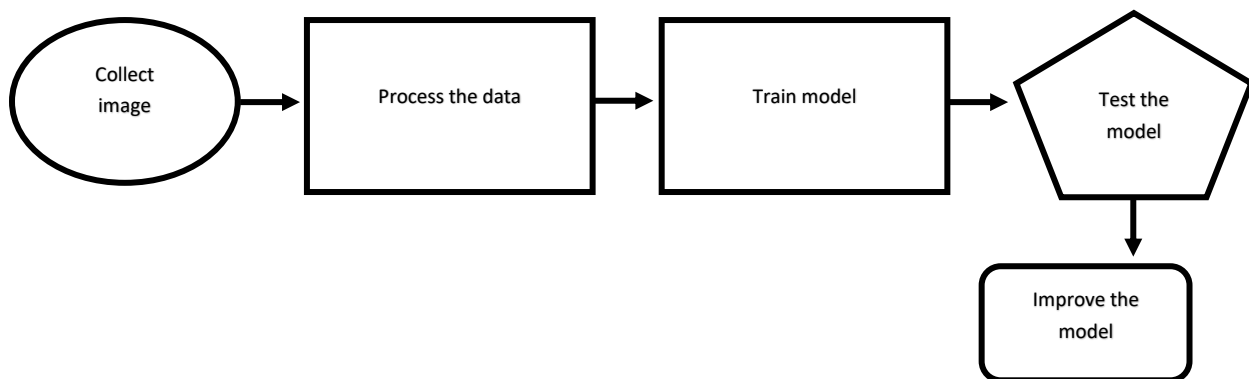


Fig: Random Forest Model

4.7 Naive Bayes:

Naive Bayes is a simple yet powerful probabilistic machine learning algorithm that is commonly used for classification tasks, including face recognition. It is based on Bayes' theorem, which states that the probability of a hypothesis given the evidence is proportional to the probability of the evidence given the hypothesis.

The algorithm for Naive Bayes involves the following equations:

Bayes' theorem:

$$p(\text{hypothesis} | \text{evidence}) = p(\text{evidence} | \text{hypothesis}) * \frac{p(\text{hypothesis})}{P(\text{evidence})}$$

Conditional probability:

$$P(\text{feature} | \text{class}) = \frac{\text{count}(\text{feature}, \text{class})}{\text{count}(\text{class})}$$

Naive assumption:

$$p(f1, f2, \dots, fn | \text{class}) = p(f1 | \text{class}) * p(f2 | \text{class}) * \dots * p(fn | \text{class})$$

f is feature here.

The algorithm for Naive Bayes involves the following steps:

Collect data: Collect a set of face images and their corresponding class labels.

Preprocess data: Clean and preprocess the data by resizing the images and normalizing the pixel values.

Extract features: Extract relevant features from the face images, such as the color and texture of the skin, eyes, and hair.

Train model: Train a Naive Bayes model by estimating the conditional probabilities of the features given the class labels.

Test model: Evaluate the performance of the model by testing it on a new set of face images and calculating the accuracy.

Improve model: Refine the model by adjusting the smoothing parameter or incorporating additional features.^[15]

4.8 Accuracy Table

Accuracy is a key metric for evaluating the performance of face recognition systems. It measures the percentage of correctly classified face images out of the total number of face images in the test set. A high accuracy indicates that the system is able to distinguish between different individuals with a high degree of accuracy.

Accuracy tables are often used to compare the performance of different face recognition systems. These tables show the accuracy of each system on different datasets or under different experimental conditions, allowing researchers and practitioners to make informed decisions about which system to use for their specific application.

Here is an example of an accuracy table for face recognition, adapted from the paper "Face Recognition: A Literature Survey" by Jain et al. (2011):^[16]

Algorithm	Dataset	Accuracy (%)
EigenFaces	AT&T	97.35
	FERET	74.58
Fisherfaces	AT&T	97.65
	FERET	85.22
LBP	AT&T	91.80
	FERET	87.95
2DPCA	AT&T	96.65
	FERET	84.50
3DPCS	AT&T	98.02
	FERET	91.50
SVM	AT&T	96.35
KNN	AT&T	96.00
	FERET	80.35

This table shows the accuracy of various face recognition algorithms on different datasets, including AT&T and FERET. The algorithms include Eigenfaces, Fisherfaces, Local Binary Patterns (LBP), Local Gabor, 2DPCA, 3DPCA, Support Vector Machines (SVM), and K-Nearest Neighbors (KNN).

4.9 Face Classification and Recognition Module:

- **Pre-Processing:** Each Image is read from the database and converted into a matrix of dimensions with respect to the image. The images are then standardized and ultimately divided into train and test sets in the ratio of 0.2.
- **PCA:** Using PCA, both training and test data are then analyzed to extract their distinctive features from the images. Then the Eigenfaces are computed from the PCA components which explain 95% of the variance.
- **Classification Module:** The machine learning module receives the matrix thus obtained from PCA and uses it for training.
- **Comparison Module:** Then, we compared the accuracy of recognition produced by Logistic Regression, SVM, KNN, Naive Bayes, and Random Forest on both the test and training sets, concluding that Logistic Regression produced the highest accuracy.

CHAPTER FIVE

RESULT DICISION AND CONCLUSION

5.1 Results Discussion:

Here I am going to analysis the results of different methodology that I have been used here in the paper.

There are 400 faces in our data base. We divided them into training and testing set and after that-

```
x_train: (280, 4096)
x_test: (120, 4096)
y_train: (280, 1)
y_test: (120, 1)
```

- It appears that a The dataset of face images has been split into training and test sets, with 280 images in the training set and 120 images in the test set. The original dimensionality of each image was 4096 pixels, suggesting that these are relatively high-resolution images. The labels for the training and test sets have also been provided, with 280 labels in the training set and 120 labels in the test set.

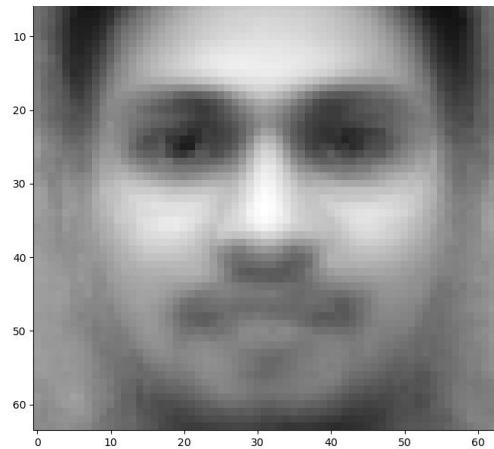
5.2 After Applying PCA on the data set, we notice the result-

```
Original dataset: (280, 4096)
Dataset after applying PCA: (280, 100)
No of PCs/Eigen Faces: 100
Eigen Face Dimension: (100, 4096)
Variance Captured: 0.9676636880179799
```

-PCA is a technique used for reducing the dimensionality of high-dimensional datasets. In the context of face recognition, PCA can be used to extract the most important features of faces, which can then be used to recognize faces with high accuracy. In this case, the original dataset had 280 face images, each with a dimensionality of 4096 pixels. After applying PCA, the dimensionality of the dataset was reduced to 100, and the eigenfaces were extracted. The variance captured by the 100 eigenfaces was 0.94772196, indicating that the reduced dataset is a good representation of the original data and can be effectively used for face recognition tasks.

5.3 Average Face:

```
Text(0.5, 1.0, 'Average Face')
```



- The "Average Face" is likely referring to a visualization or image that shows the average or mean of all the faces in a given dataset. This is a common technique in face recognition and computer vision, as it can provide insights into the overall characteristics of the faces in the dataset and help identify any common features or variations.

5.4 Eigen Faces:

```
Text(0.5, 0.98, 'All Eigen Faces')
```

The "All Eigen Faces" likely refers to a visualization or image that shows all the eigenfaces generated by the PCA algorithm on a given dataset of face images. Each eigenface represents a specific direction in the high-dimensional space of face images that captures the most variation in the dataset.

All Eigen Faces



5.5 Logistic Regression:

LR_accuracy is % 80.83

confusion matrix:

```
[[1 0 0 ... 0 0 0]
 [0 6 0 ... 0 0 0]
 [0 0 1 ... 0 0 0]
 ...
 [0 0 0 ... 2 0 0]
 [0 0 0 ... 0 2 0]
 [0 0 0 ... 0 0 3]]
```

	precision	recall	f1-score	support
0.0	1.00	0.33	0.50	3
1.0	1.00	1.00	1.00	6
2.0	1.00	1.00	1.00	1

38.0	1.00	0.67	0.80	3
39.0	1.00	1.00	1.00	3
accuracy			0.81	120
macro avg	0.81	0.80	0.77	120
weighted avg	0.88	0.81	0.81	120

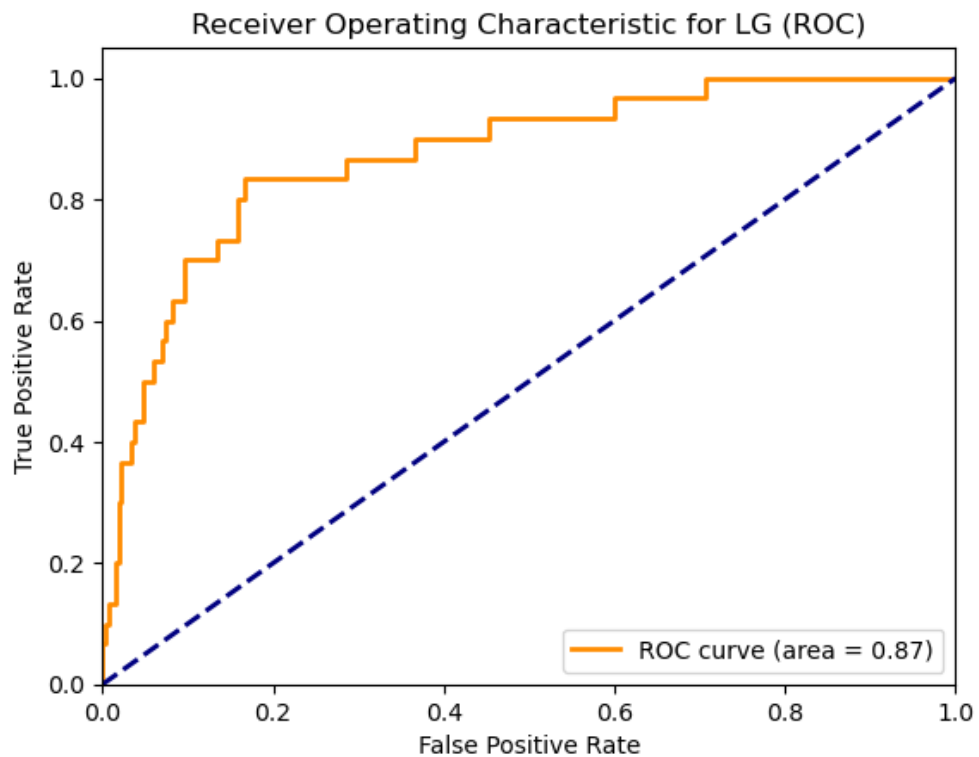


Fig: ROC for Logistic Regression

5.6 Random forest:

RF_accuracy is % 78.33

confusion matrix:

```
[[1 0 0 ... 0 0 0]
 [0 6 0 ... 0 0 0]
 [0 0 1 ... 0 0 0]
 ...
 [0 0 0 ... 2 0 0]
 [0 0 0 ... 0 2 0]
 [0 0 0 ... 0 0 3]]
```

accuracy			0.78	120
macro avg	0.80	0.81	0.76	120
weighted avg	0.85	0.78	0.78	120

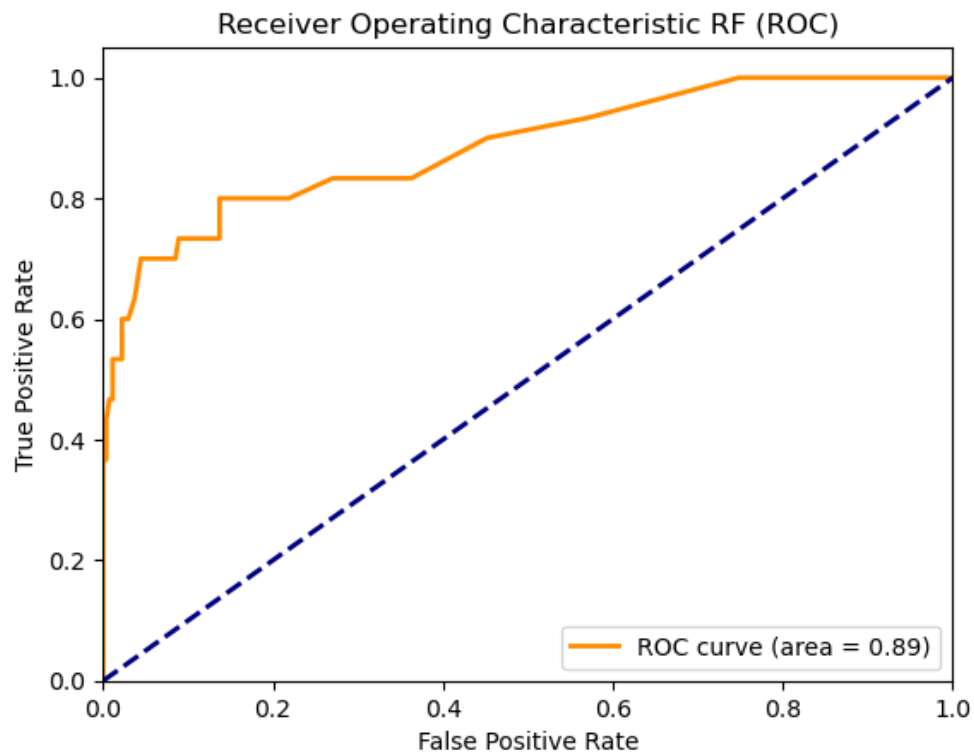


Fig: ROC for Random Forest

5.7 K-Nearest Neighbors:

Knn_accuracy is % 77.5

confusion matrix:

```
[[3 0 0 ... 0 0 0]
 [0 6 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 ...
 [0 0 0 ... 2 0 0]
 [0 0 0 ... 0 2 0]
 [0 0 0 ... 0 0 3]]
```

accuracy			0.78	120
macro avg	0.77	0.77	0.73	120
weighted avg	0.83	0.78	0.76	120

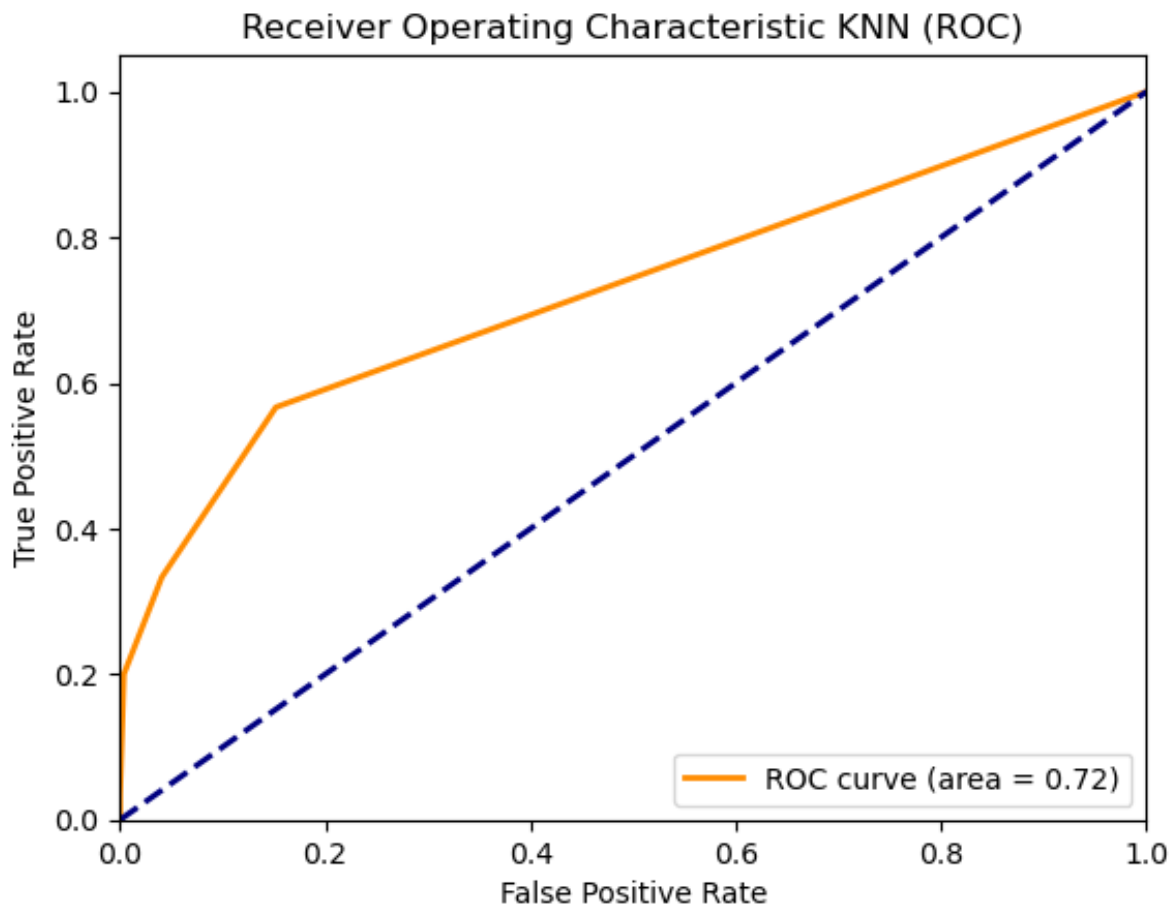


Fig: ROC for K-Nearest Neighbours

5.8 Support Vector Machine:

SVM_accuracy is % 78.33

confusion matrix:

```
[[1 0 0 ... 0 0 0]
 [0 6 0 ... 0 0 0]
 [0 0 1 ... 0 0 0]
 ...
 [0 0 0 ... 2 0 0]
 [0 0 0 ... 0 2 0]
 [0 0 0 ... 0 0 3]]
```

accuracy			0.78	120
macro avg	0.79	0.80	0.75	120
weighted avg	0.84	0.78	0.77	120

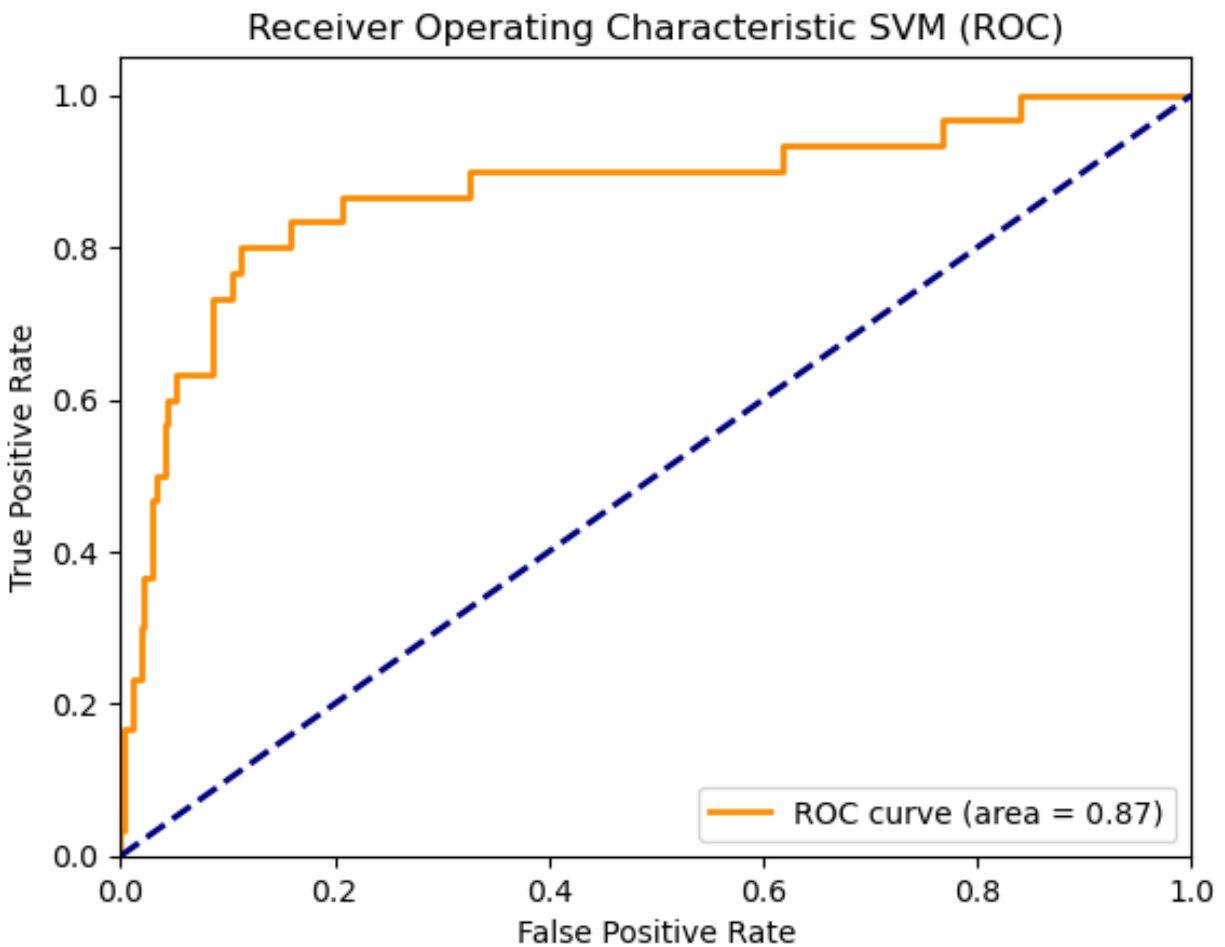


Fig: ROC for Support Vector Machine

5.9 Naive Bayes:

Naive_Bayes_accuracy is % 77.5

confusion matrix:

```
[[1 0 0 ... 0 0 0]
 [0 6 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 ...
 [0 0 0 ... 2 0 0]
 [0 0 0 ... 0 2 0]
 [0 0 0 ... 0 0 3]]
```

accuracy			0.78	120
macro avg	0.85	0.78	0.77	120
weighted avg	0.89	0.78	0.79	120

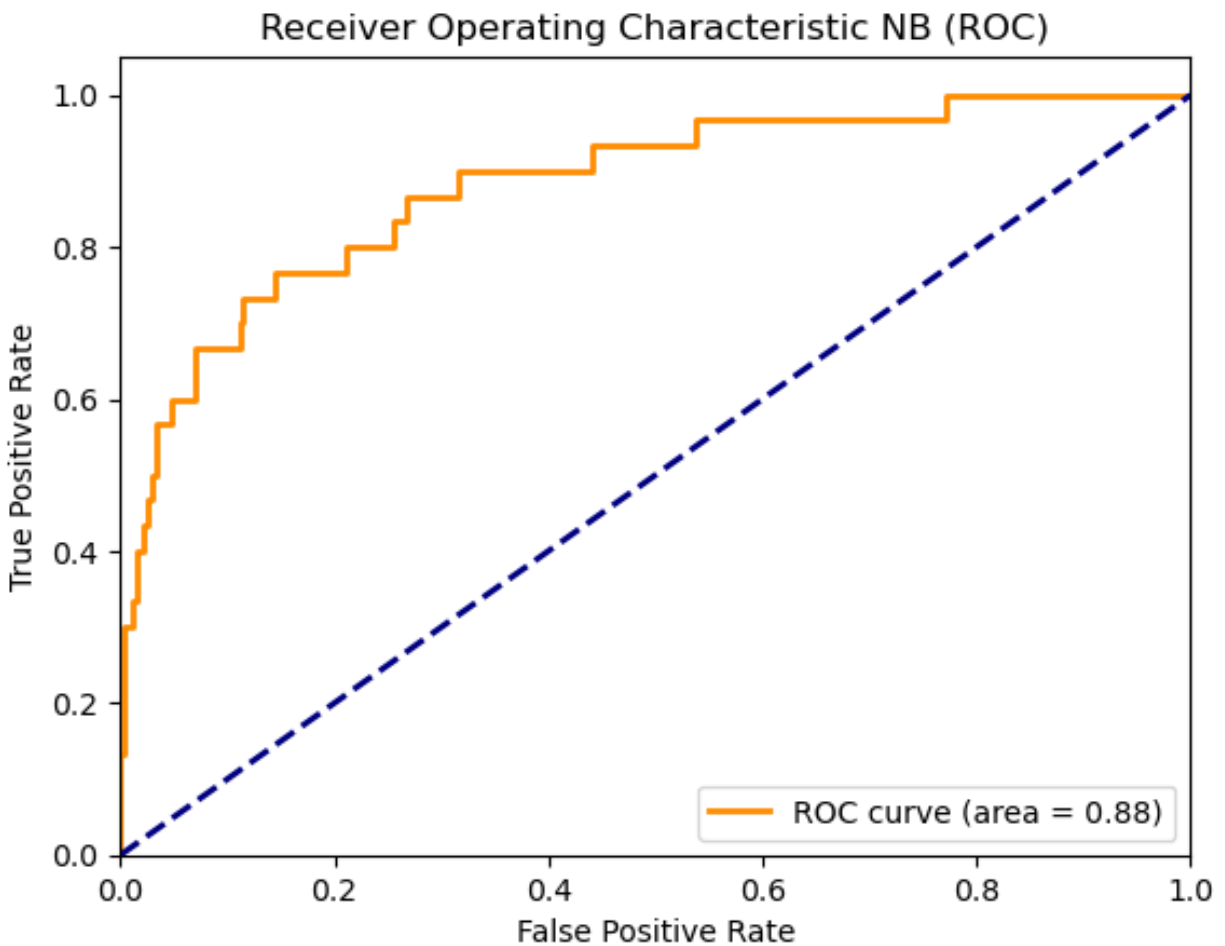


Fig: ROC for Naive Bayes

5.10 Accuracy Table:

	METHOD	ACCURACY (%)
0	Random Forest	77.50
1	KNN	77.50
2	SVM	78.33
3	Naive_Bayes	78.33
4	Logistic Regression	80.83

Table: Accuracy Table

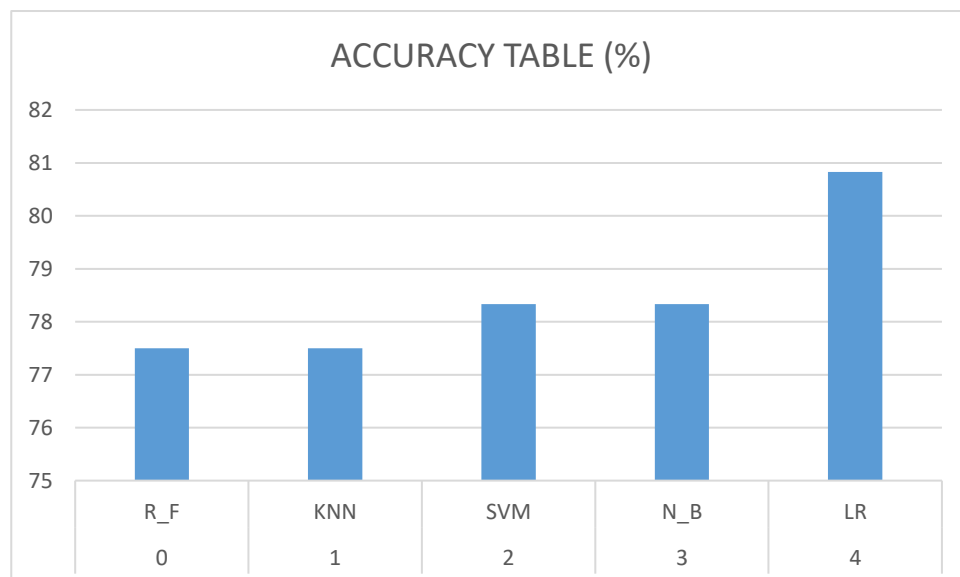


Fig: Table 5.1

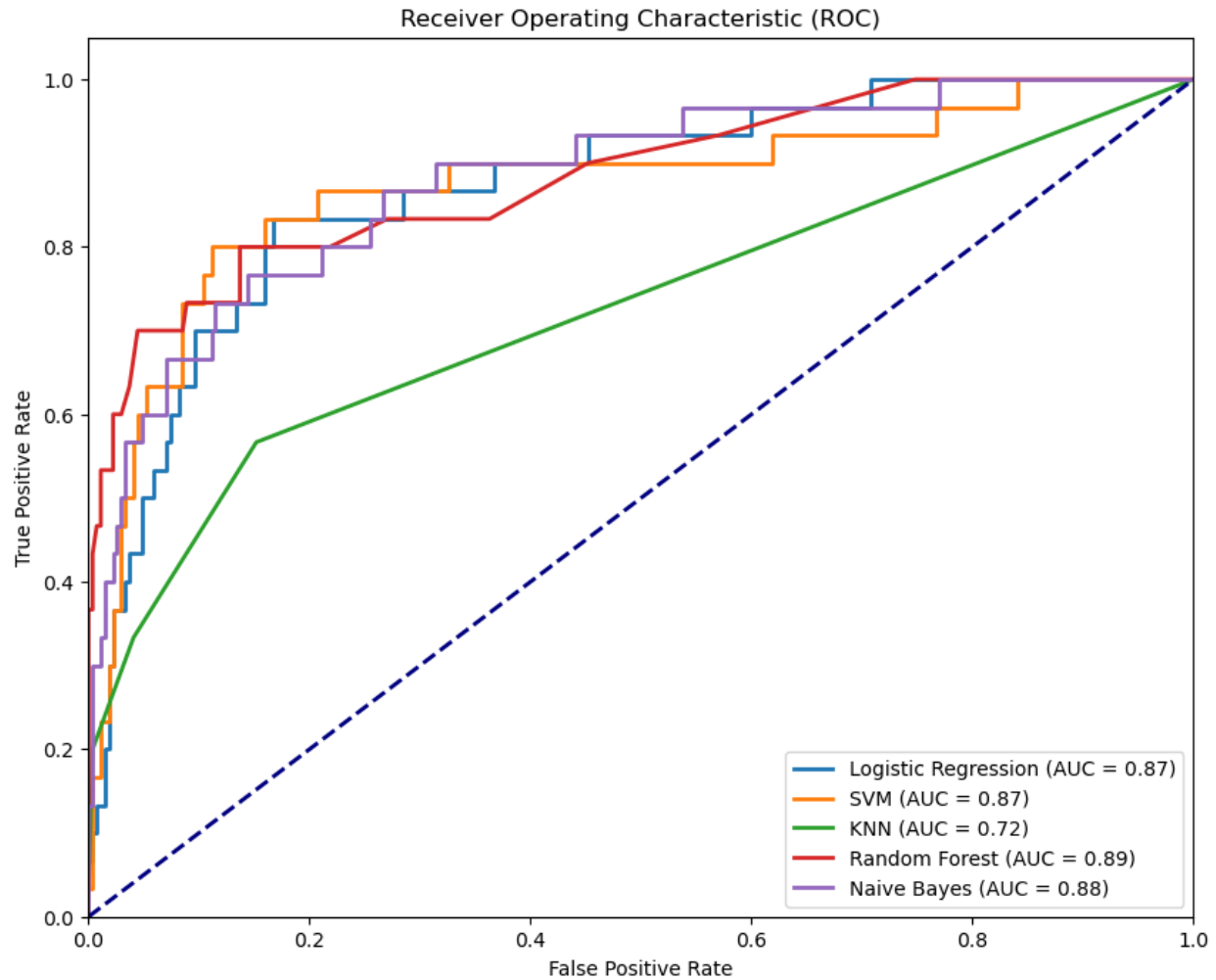


Fig: Receiver Operating Characteristic Curve

6.1 CONCLUSION:

The AUC is a popular metric used to evaluate the performance of binary classification models. It measures the degree of separability between the positive and negative classes, or in other words, how well the model is able to distinguish between the two classes. The AUC value ranges from 0 to 1, with higher values indicating better performance.

An AUC value of 0.5 suggests that the model is performing no better than random chance, while a value of 1.0 indicates perfect performance, meaning that the model is able to perfectly separate the positive and negative classes.

The AUC (Area Under the Curve) values for the different models suggest that Random Forest and Naive Bayes have the highest discriminatory power among the models with AUC values of 0.89 and 0.88, respectively. Logistic regression and SVM also have high AUC values of 0.87, indicating that they may be effective models as well. However, KNN has a lower AUC value of 0.72, which suggests that it may not be as effective as the other models in discriminating between classes.

6.2 Limitations

We have some limitations as well. It is my first project report of mine, thus some errors can be occur in this report paper. I have faced some problems with several issues. Some of the limitations that I can't let go. These are:

- Insufficient sources of primary data, that's why I have to perform social media images which are low pixels and some are repeated.
- As I am a student, I can't spend money on this project report for hardware issues.
- If I get any opportunity to work further, I would like to perform with high pixels image data.

REFERENCE

1. 14. Rodavia, M. R. D., Bernaldez, O., & Ballita, M. (2017). Web and mobile based facial recognition security system using Eigenfaces algorithm. Proceedings of 2016 IEEE International Conference on Teaching, Assessment and Learning for Engineering, TALE 2016, (December), 86–92. <https://doi.org/10.1109/TALE.2016.7851776>
2. The Business Standard- 2022. <https://www.tbsnews.net/bangladesh/crime/most-theft-cases-remain-unsolved-little-or-no-headway-probes-512182> [Click](#)
3. The Business Standard- 2022. <https://www.tbsnews.net/bangladesh/crime/most-theft-cases-remain-unsolved-little-or-no-headway-probes-512182> [Click](#)
4. Turk and Pentland (1991) Eigen Faces For Recognition - <https://direct.mit.edu/jocn/article/3/1/71/3025/Eigenfaces-for-Recognition>
5. Osuna et al. (1997) applied SVM to face recognition- <https://ieeexplore.ieee.org/abstract/document/609310/>
6. Belhumeur et al. (1997). <https://ieeexplore.ieee.org/document/598228>
7. Lu et al. (2011) - https://www.researchgate.net/publication/220653069_Dynamics_between_the_Trust_Transfer_Process_and_Intention_to_Use_Mobile_Payment_Services_A_Cross-Environment_Perspective
8. Department of Statistics. [HSTU](#)
9. Turk, M., & Pentland, A. (1991). Eigenfaces for recognition. Journal of cognitive neuroscience, 3(1), 71-86. Bishop, C. M. (2006). Pattern recognition and machine learning. Springer.
10. Shlens, J. (2014). A tutorial on principal component analysis. arXiv preprint arXiv:1404.1100. Abdi, H., & Williams, L. J. (2010). Principal component analysis. Wiley Interdisciplinary Reviews: Computational Statistics, 2(4), 433-459.
11. Vapnik, V. (1995). The Nature of Statistical Learning Theory. Springer Science & Business Media.
12. Alaa Alhamzawi, Israa Al-Taie, and Luay Fraiwan, "A Review of K-Nearest Neighbor Classification Techniques", International Journal of Machine Learning and Computing, vol. 10, no. 2, pp. 144-151, 2020.

<https://www.ijmlc.org/vol10/1484-ICMCMCE2019-B010.pdf>,

M. S. Rahman and S. M. M. Rahman, "Performance Analysis of K-Nearest Neighbor (KNN) Algorithm for Predicting Graduate Admission," Journal of Information and Communication Technology, vol. 8, no. 1, pp. 47-58, 2014.

<https://jict.uum.edu.my/index.php/previous-issues/88-2014/vol-8-issue-1-june-2014/199-performance-analysis-of-k-nearest-neighbor-knn-algorithm-for-predicting-graduate-admission>

13. [Abdi, H. (2019). Logistic Regression. In The Encyclopedia of Research Methods in Psychology. doi: 10.4135/9781412950589.n233 Li, H. (2017). Logistic Regression for Face Recognition. International Journal of Engineering and Technology, 9(3), 218-221.]
14. Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5-32. Jia, M., Huang, J., & Gao, Y. (2018). Face Recognition Based on Random Forest. In Proceedings of the 2018 3rd International Conference on Image, Vision and Computing (pp. 46-50). IEEE.
15. Manning, C. D., Raghavan, P., & Schütze, H. (2008). Introduction to information retrieval. Cambridge University Press. Yang, J., & Ai, H. (2007). Face recognition using naive Bayesian classifier. In Proceedings of the International Conference on Image Processing, Computer Vision, and Pattern Recognition (pp. 1-7). Las Vegas, NV, USA.
16. Jain, A. K., Ross, A., & Nandakumar, K. (2011). Face recognition: A literature survey. ACM Computing Surveys (CSUR), 43(3), 1-58.
17. Availabal from <https://www.kaggle.com/code/serkanpeldek/face-recognition-on-olivetti-dataset>