Visvesvaraya Technological University, Belagavi.

PROJECT SYNOPSIS

on

"Unique face identification system using machine learning"

Project Synopsis Report submitted in partial fulfillment of the requirement for the award of the degree of

Bachelor of Engineering

in

Electronics and Communication Engineering

For the academic year 2018-19

Submitted by

1CR16EC181	VASHNAVI CHAWDA
1CR16EC183	VANDANA ARYA
1CR16EC208	SHUCHI PANDEY
1CR16EC161	SHRISTI

Under the guidance of

Internal

Prof . Manohar Valleti

Assistant professor

Department of ECE

CMRIT, Bangalore

Department of Electronics and Communication Engineering

CMR Institute of Technology, Bengaluru – 560 037

For college use only

Reviewer Names	Sign	Accepted	Correction Needed	Rejected
1.				
2.				

UNIQUE FACE IDENTIFICATION SYSTEM USING MACHINE LEARNING

ABSTRACT

Human face detection is one domain in computer vision application. There are so many researches in the field of image processing on the face. Some researchers previously conducted research on face recognition, no matter from what tribe, how healthy, how old, they developed research on facial expressions, whether sad, normal or laugh. Some previous research also built on the face detection system focusing on environmental conditions. As the technology advances, the graphic images are fully utilized to promote the welfare of mankind. Unique identification is generally used as a complementary security and is widely used in industries, shops, offices, factories, and even today many housing have been using and applying this technology. The use of security system with face detection is expected to see the actual condition and detect any human presence on the video.

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

CERTIFICATE

This is to certify the Project Report entitled "**Unique face identification system using machine learning**", prepared by **Ms. Vandana Arya**, **Ms. Vaishnavi Chawda, Ms. Shuchi Pandey, Ms. Shristi** bearing **USN 1CR16EC183**, **1CR16EC181, 1CR16EC208, 1CR16EC161,** being bona fide students of **CMR Institute of Technology, Bengaluru** in partial fulfillment of the requirements for the award of **Bachelor of Engineering in Electronics and Communication Engineering** of the V**isvesvaraya Technological University, Belagavi-590018** during the academic year 2019-20.

This is certified that all the corrections and suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The Project report has been approved as it satisfies the academic requirements prescribed for the said degree.

---------------------	----------------------
Signature of Guide	**Signature of HOD**
Mr. Manohar Valleti	**Mr. R Elumalai**
Asst. Professor	**Professor & HoD**
Dept. of ECE, CMRIT	**Dept. of ECE, CMRIT**

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without mentioning the people whose proper guidance and encouragement has served as a beacon and crowned my efforts with success. I take an opportunity to thank all the distinguished personalities for their enormous and precious support and encouragement throughout the duration of this seminar.

I take this opportunity to express my sincere gratitude and respect to **CMR Institute of Technology, Bengaluru** for providing me an opportunity to present my project.

I have a great pleasure in expressing my deep sense of gratitude to **Dr. Sanjay Jain,** Principal, CMRIT, Bangalore, for his constant encouragement.

I would like to thank **Mr. R Elumalai,** HoD, Department of Electronics and Communication Engineering, CMRIT, Bangalore, who shared his opinion and experience through which I received the required information crucial for the project.

I consider it a privilege and honor to express my sincere gratitude to my guide **Mr. Manohar Valleti**, Assistant Professor, Department of Electronics and Communication Engineering, for the valuable guidance throughout the tenure of this review.

I also extend my thanks to the faculties of Electronics and Communication Engineering Department who directly or indirectly encouraged me.

Finally, I would like to thank my parents and friends for all their moral support they have given me during the completion of this work.

Vandana Arya (1CR16EC183)

Vaishnavi Chawda (1CR16EC181)

Shuchi Pandey (1CR16EC208)

Shristi (1CR16EC161)

Table of contents

List of Figures

Chapter 1

INTRODUCTION

Unique Identification system technique originally was created for the purpose of security surveillance to anticipate various functionalities like crime detection, theft, robbery etc. Here, we propose a live video identification system that helps to identify and detect faces from continuous video frames. When a camera starts working, initial frames have to be taken to detect other objects from the frames. The after frames coming from the live camera are to be compared with the initial image to mark the human presence and then detect the faces. The process can be done by image capturing, preparation of dataset, face detection, and pre-processing of the taken image. In the proposed system, face detection and comparison can be done by Haar Cascade and PCA (Principal Component Analysis) and classification is done by SVC (support vector classifier). The performance of an Identification system also depends upon the feature extraction and their classification to get the accurate results. Principal Component Analysis (PCA) was the first algorithm that represents the faces economically. Comparison of video frames can be done by using this technique. Faces can be detected from the frames by the application of Haar cascade to extract features of the human face. However, the system helps to identify and detect faces using the most effective techniques that can be helpful to mankind in the most effective way.

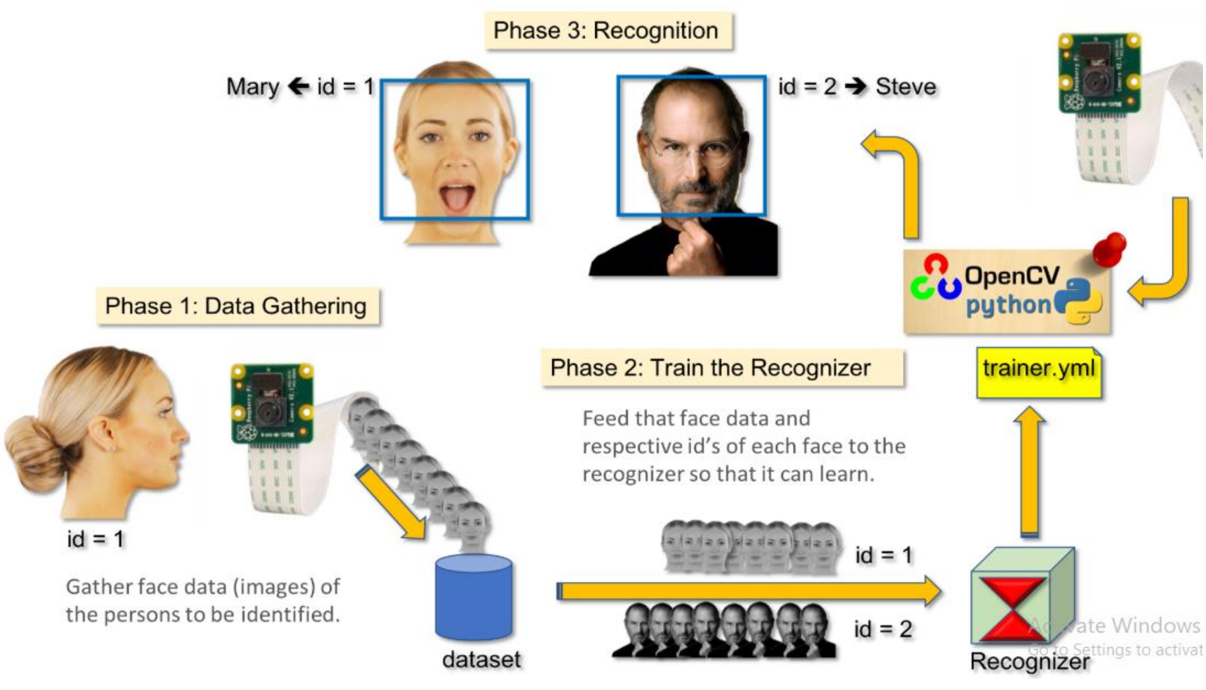


Figure 1. Overview

1.1 EXISTING SYSTEM

The present systems that are used to detect faces from video are in more costly ways. These systems are widely used in different organizations. Traditional method of human detection and identification is a manual method that is very time consuming and needs more effort for people. Although some techniques are used to analyze object detection through live monitoring, person identification cannot be applicable through these systems.

1.1.1 DRAWBACKS

- Time consuming
- Need manual work
- Only object detection is possible
- Cannot process proper face detection

1.2 PROPOSED SYSTEM

The proposed systems are composed of unique identity detection and face detection. The proposed face identity technique is used for multiple purposes and can be applicable in various areas. The technique is initiated when the base frame of video is captured. This initial frame is used to compare other frames. The remaining processes are followed by database development, face detection, pre-processing, feature extraction, and classification stages. By extracting and classifying a face's features, a person can be identified from the previously prepared dataset. The proposed system is very accurate in producing results and is very useful by applying the technique is surveillance cameras.

1.2.1 ADVANTAGES

- Less time consuming
- Cost effective
- Provide accurate results for face identification
- Consumes less manual work

Chapter 2

METHODOLOGY

Live video identification system that helps to identify and detect faces from continuous video frames. When a camera starts working, initial frames have to be taken to detect the objects from the frames. The Haar filters works at the time of image acquisition along with the camera. The frames coming from the live camera are to be compared with the initial image to mark the human presence and then detect the faces. The process can be done by image capturing, preparation of dataset, face detection, and pre-processing of the taken image. In the proposed system, face detection and comparison can be done by Haar Cascade and PCA (Principal Component Analysis) and classification is done by SVC (support vector classifier). The performance of an Identification system also depends upon the feature extraction and their classification to get the accurate results. Principal Component Analysis (PCA) was the first algorithm that represents the faces economically. It extracts the most dominant Eigen faces from the present set of the faces. Comparison of video frames can be done by using this technique. Faces can be detected from the frames by the application of Haar cascade to extract features of the human face. However, the system helps to identify and detect faces using the most effective techniques that can be helpful so mankind in the most effective way. The block diagram for the same is shown.

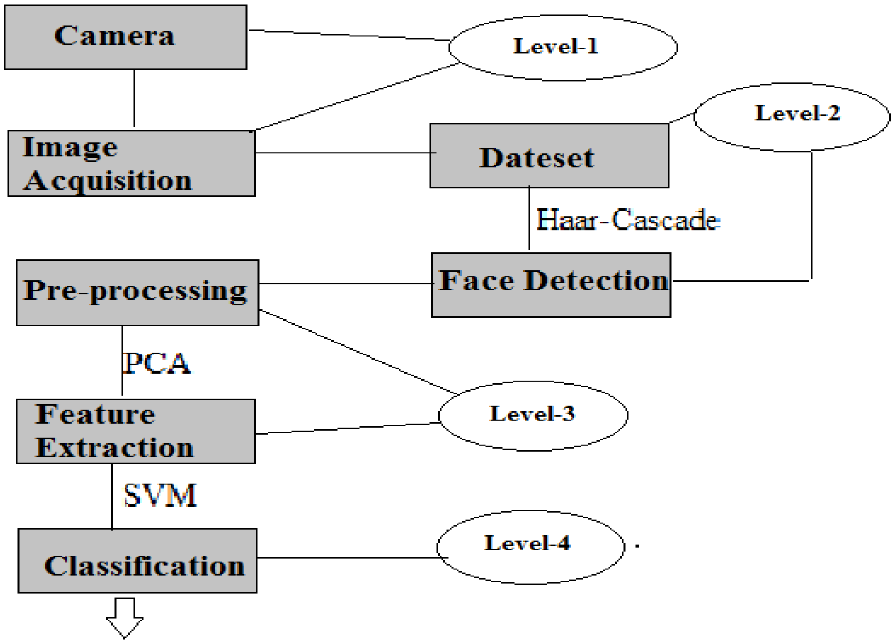


Figure 2. Block Diagram

Chapter 3

PROBLEM STATEMET

Growing numbers of security incidents have installed a sense of insecurity in individuals and business. They are doing what is feasible and affordable to them to feel more secure. Installing security systems at home and offices is considered as a first layer of defense against security threats. The present systems that are used to detect faces from video are in more costly ways. These systems are widely used in different organizations. Traditional method of human detection and identification is a manual method that is very time consuming and needs more effort for people. This project will design and implement a security system based on Machine learning algorithms. Therefore, we propose to design and implement a system that will identify and detect faces using the most effective techniques that can be helpful to mankind in the most effective way.

Chapter 4

LITERATURE SURVEY

4.1 MACHINE LEARNING

Machine Learning is a field of artificial intelligence that uses statistical techniques to give computer systems the ability to "learn" (e.g., progressively improve performance on a specific task) from data, without being explicitly programmed. Machine Learning is an idea to learn from examples and experience, without being explicitly programmed. Inside of writing code, you feed data to the generic algorithm, and it builds logic based on the data given. A Computer program is said to learn from experience E with some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, Improves with experience. Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it learn for themselves.

The process of learning begins with observations or data, such as examples, direct experience, or instruction, in order to look for patterns in data and make better decisions in the future based on the examples that we provide. The primary aim is to allow the computers learn automatically without human intervention or assistance and adjust actions accordingly. Some machine learning methods. Machine learning algorithms are often categorized as supervised or unsupervised.

Supervised machine learning algorithms can apply what has been learned in the past to new data using labeled examples to predict future events. Starting from the analysis of a known training dataset, the learning algorithm produces an inferred function to make predictions about the output values. The system is able to provide targets for any new input after sufficient training. The learning algorithm can also compare its output with the correct, intended output and find errors in order to modify the model accordingly.

In contrast, unsupervised machine learning algorithms are used when the information used to train is neither classified nor labeled. Unsupervised learning studies how systems can infer a function to describe a hidden structure from unlabeled data. The system doesn't figure out the right output, but it explores the data and can draw inferences from datasets to describe hidden structures from unlabeled data.

4.2 NEED FOR MACHINE LEARING

Machine Learning is a field which is raised out of Artificial Intelligence (AI). Applying artificial Intelligence, we wanted to build better and intelligent machines. But expect for few tasks such as finding the shortest path between two points, it was unable to program more complex and constantly evolving challenges. There was a realization that the only way to be able to achieve this task was to let machine learn from itself. This sounds similar to child learning from itself. So, machine learning was developed as a capability for computers. And Now Machine Learning is present in so many segments of technology, that we don’t even realize it while using it. Finding patterns into data on planet earth is possible only for human brains. The data being very massive, the time taken to compute is increased, and this is where machine learning comes into action, to help people with large data in minimum time. The techniques we use for data mining have been around for many years, but they were not effective as they did not have the competitive power to run the algorithms. If you run deep learning with access to better data, the output we get will lead to dramatic breakthroughs which is Machine Learning.

Facial recognition is a biometric software application capable of uniquely identifying or verifying a person by comparing and analyzing patterns based on the person’s facial contours. Facial recognition is mostly used for security purposes, though there is increasing interest in other areas of use. In fact, facial recognition technology has received significant attention as it has potential for a wide range of application related to law enforcement as well as other enterprises.

There are different facial recognition techniques in use, such as the generalized matching face detection method and the adaptive regional blend matching method. Most facial recognition systems function based on the different nodal points on a human face. The values measured against the variable associated with points of a person’s face help in uniquely identifying or verifying the person. With this technique, applications can use data captured from faces and can accurately and quickly identify target individuals. Facial recognition techniques are quickly evolving with new approaches such as 3-D modeling, helping to overcome issues with existing techniques.

There are many advantages associated with facial recognition. Compared to other biometric techniques, facial recognition is of a non-contact nature. Face images can be captured from a distance and can be analyzed without ever requiring any interaction with the user/person. As a result, no user can successfully imitate another person. Facial recognition can serve as an excellent security measure for time tracking and attendance. Facial recognition is also cheap technology as there is less processing involved, like in other biometric techniques.

Machine learning has solved many problems by choosing one machine learning algorithm, feeding in data, and getting the result. We do not need to build our own neural network. We have access to a trained model that can be used. It does exactly what we need it to do outputs a bunch of numbers (face encodings) when we pass in the image of someone's face; comparing face encodings of faces from different images will tell us if someone's face matches with anyone we have images of.

4.3 EXISTING SYSTEM

In the Existing System "Live Camera Monitoring". Vehicle detection is an important component in many related applications, such as self-guided vehicles, driver assistance systems, intelligent parking systems, or measurement of traffic parameters, including vehicle count, speed, and flow. A recent trend is to apply vision-based techniques to analyze vehicles. However, vision-based vehicle detection is a challenging topic due to the huge within-class variability. For example, vehicles may vary in shape, size, and color. Moreover, vehicle appearance depends on its pose and may be affected by nearby objects. . It needed to process acquired images in real-time to save more time for driver reaction. Template-based methods need to use thousands of predefined patterns of the vehicle class and perform correlation between the test image and the template, which makes them time-consuming. In addition, template-based methods are sensitive to the varying background (e.g. buildings, bridges and guardrails). Therefore, appearance-based methods are more common in the vision-based vehicle detection literature. Appearance-based methods, which rely on the machine learning, learn the characteristics of the vehicle class from a set of training images which capture the variability of the vehicle appearance. Usually, the variability of the non-vehicle class is also modeled to improve the performance. Firstly, each training image is represented by a set of local or global features. Then, the decision boundary between the vehicle and nonvehicle classes is learned by training a classifier. There are at least two fundamental challenges faced by appearance-based validation methods: the accuracy and the processing time. In this paper, we focus on the investigation of the processing time of the machine learning methods, specifically, seeking solutions to speed up the training and incremental learning processes based on AdaBoost. We first design a Haar-like feature extraction method to represent a vehicle's edges and structures, and then propose a rapid feature selection algorithm by using AdaBoost due to the large pool of Haar-like features. Finally, we design an incremental learning algorithmto improve the classification performance significantly. Experimental results demonstrate that the proposed approaches not only speed up both the training process and the incremental learning process of AdaBoost, but also achieve competitive classification accuracies compared with several state-of-the-art methods. We present an algorithm for computing Haar-like features. In recent years, there has been a transition from complex image features, such as edges and symmetry, to general and robust feature sets for vehicle detection. Haar-like features are extremely well represented and demonstrated by their good performance in the object detection. There are two motivations for the employment of the Haar-like features rather than raw pixel values. The first reason is that the Haar-like features can encode ad hoc domain knowledge, which is difficult to describe using a finite quantity of training data. The second motivation is that a Haar-like feature-based system can operate much faster than a pixel-based system. Therefore, we choose Haar-like features as the vehicle feature representation.

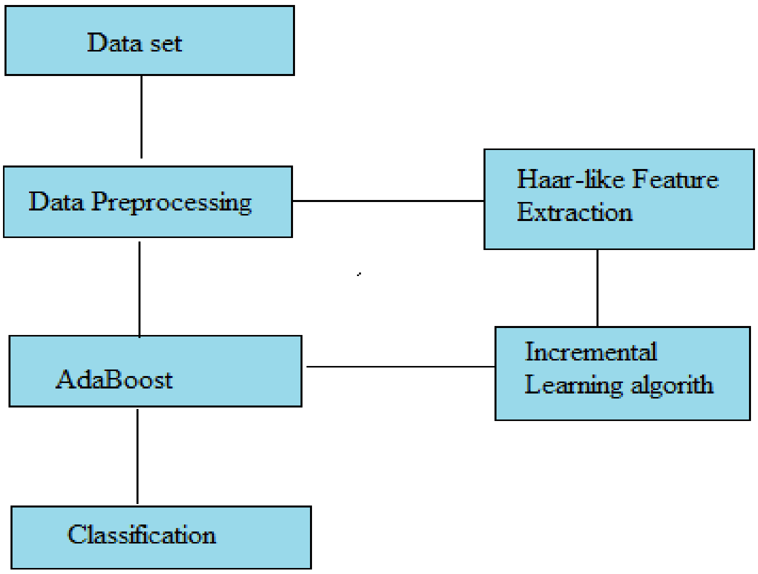


Figure 3. System architecture of Existing System

Figure 3 describes the system architecture of the existing system. The system had been classified into - Data set, Image Pre-Processing , Feature Extraction and AdaBoost algorithm ,Test Image Pre-Processing , Classification

4.3.1 DISADVANTAGES

1. Haar Features have to be determined manually, there is a certain limit to the types of things it can detect. If you give classifier (a network, or any algorithm that detects faces) edge and line features, then it will only be able to detect objects with clear edges and lines. Even as a face detector, if we manipulate the face a bit (say, cover up the eyes with sunglasses, or tilt the head to a side), a Haar-based classifier may not be able to recognize the face.

2. Complexity of the classification increases by using Adaboost algorithm.The power of ensembling is such that we can still build powerful ensemble models even when the individual models in the ensembles are extremely simple.

3. Decision Stumps are the simplest model we could construct would just guess the same label for every new example, no matter what it looked like. The accuracy of such a model would be best if we guess whichever answer, 1 or 0, is most common in the data. If, say, 60% of the examples are 1s, then we'll get 60% accuracy just by guessing 1 every time.

4. Decision stumps improve upon this by splitting the examples into two subsets based on the value of one feature. Each stump chooses a feature, say X2, and a threshold, T, and then splits the examples into the two groups on either side of the threshold.

5. To find the decision stump that best fits the examples, we can try every feature of the input along with every possible threshold and see which one gives the best accuracy. While it naively seems like there are an infinite number of choices for the threshold, two different thresholds are only meaningfully different if they put some examples on different sides of the split. To try every possibility, then, we can sort the examples by the feature in question and try one threshold falling between each adjacent pair of examples.

6. The algorithm just described can be improved further, but even this simple version is extremely fast in comparison to other ML algorithms.

4.4. PROPOSED SYSTEM

The proposed systems composed of live camera monitoring and face detection. The proposed face recognition technique is used for multiple purposes and can be applicable in various areas. The technique is initiated when the base frame of video is captured. The remaining processes are followed by database development, face detection, pre- processing, feature extraction, and classification stages. By extracting and classifying a face's features, person can be identified from the previously prepared dataset. In this proposed system we are using Haar cascade method for face detection where we get ROI (region of interest). Haar Cascade is basically a classifier which is used to detect the object for which it has been trained for, from the source. The Haar Cascade is trained by superimposing the positive image over a set of negative images. We are using PCA (principal component analysis) algorithm for feature extraction. PCA is predominantly used as a dimensionality reduction technique in domains like facial recognition, computer vision and image compression. Finally, SVM (support vector machine) for classification of data. A support vector machine (SVM) is machine learning algorithm that analyzes data for classification analysis.

Comparison of video frames can be done by using this technique. Faces can be detected from the frames by the application of Haar cascade to extract features of human face. Design and implement a security system based on Raspberry Pi microcomputer where images get accumulated as a data after pre-processing . However, the system helps to identify and detect faces using the most effective techniques that can be helpful to mankind in the most effective way.

4.5 FACE DETECTION USING HAAR CASCADE

Object Detection using Haar feature-based cascade classifiers is an effective method proposed by Paul Viola and Michael Jones in the 2001 paper, "Rapid Object Detection using a Boosted Cascade of Simple Features". It is a machine learning based approach in which a cascade function is trained from a lot of positive and negative images. It is then used to detect objects in other images.

Here we will work with face detection. Initially, the algorithm needs a lot of positive images (images of faces) and negative images (images without faces) to train the classifier. Then we need to extract features from it. For this, Haar features shown in below image are used. They are just like our convolutional kernel. Each feature is a single value obtained by subtracting the sum of pixels under the white rectangle from the sum of pixels under the black rectangle.

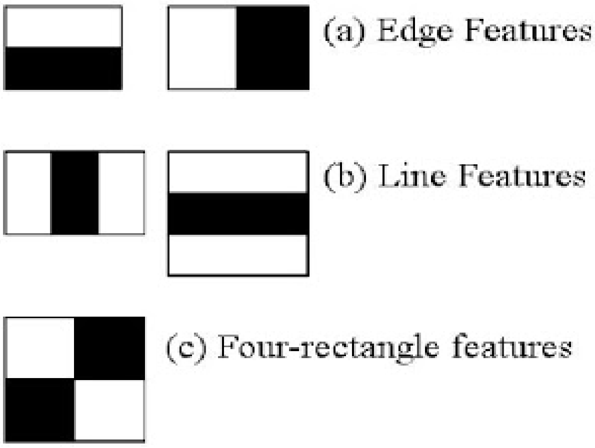


Figure 4. Haar cascade filters

Now all possible sizes and locations of each kernel are used to calculate plenty of features. For each feature calculation, we need to find the sum of the pixels under the white and black rectangles. To solve this, they introduced the integral images. It simplifies calculation of the sum of the pixels, how large may be the number of pixels, to an operation involving just four pixels.

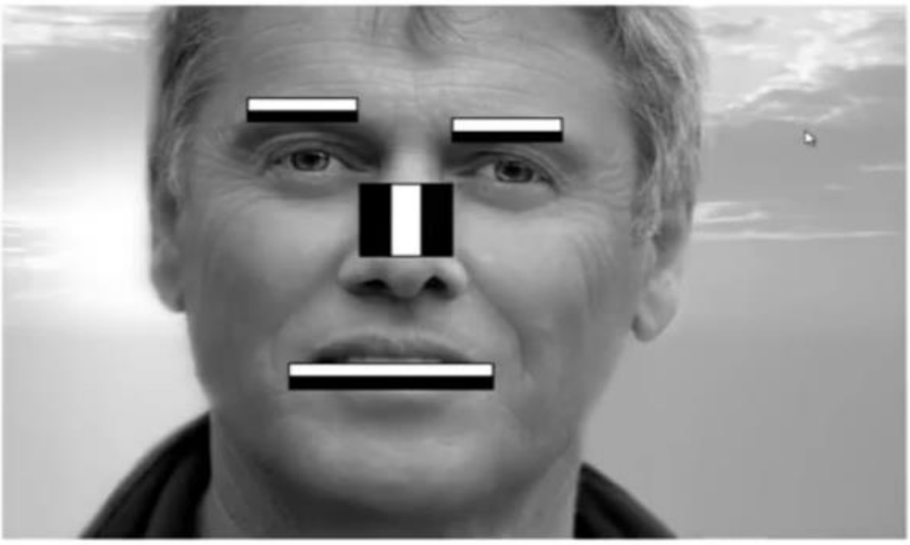


Figure 5. Representing more relevant features for haar cascade

For example, consider the above image. The eyebrows of the person is detected by the horizontal line feature filters. The area around the eyebrows are lighter. The eyebrow is detected because the its color is darker and the area around it is brighter. Similarly, for nose, the nose length contains brighter pixels and areas around it contains darker pixels. Hence it is detected by vertical line feature filters. And similarly mouth is detected, by horizontal line detector filter because of the color of the pixels in the area.

For this, we apply each and every feature on all the training images. For each feature, it finds the best threshold which will classify the faces to positive and negative. But obviously, there will be errors or misclassifications. We select the features with minimum error rate, which means they are the features that best classifies the face and non-face images. (The process is not as simple as this. Each image is given an equal weight in the beginning. After each classification, weights of misclassified images are increased. Then again same process is done. New error rates are calculated. Also new weights. The process is continued until required accuracy or error rate is achieved or required number of features are found).

Final classifier is a weighted sum of these weak classifiers. It is called weak because it alone can't classify the image, but together with others forms a strong classifier. It says even 200 features provide detection with 95% accuracy. Their final setup had around 6000 features.

In an image, most of the image region is non-face region. So it is a better idea to have a simple method to check if a window is not a face region. If it is not, discard it in a single shot. Don't process it again. Instead focus on region where there can be a face. This way, we can find more time to check a possible face region. For this they introduced the concept of Cascade of Classifiers. Instead of applying all the 6000 features on a window, group the features into different stages of classifiers and apply one-by-one. (Normally first few stages will contain very less number of features). If a window fails the first stage, discard it. We don't consider remaining features on it. If it passes, apply the second stage of features and continue the process. According to authors, on an average, 10 features out of 6000+ are evaluated per sub-window. So this is a simple intuitive explanation of how Viola-Jones face detection works. Historically, working with only image intensities (i.e., the RGB pixel values at each and every pixel of image) made the task of feature calculation computationally expensive. A publication by Papageorgiou et al discussed working with an alternate feature set based on Haar wavelets instead of the usual image intensities. Paul Viola and Michael Jones adapted the idea of using Haar wavelets and developed the so-called Haar-like features. A Haar-like feature considers adjacent rectangular regions at a specific location in a detection window, sums up the pixel intensities in each region and calculates the difference between these sums. This difference is then used to categorize subsections of an image. For example, with a human face, it is a common observation that among all faces the region of the eyes is darker than the region of the cheeks. Therefore, a common Haar feature for face detection is a set of two adjacent rectangles that lie above the eye and the cheek region. The position of these rectangles is defined relative to a detection window that acts like a bounding box to the target object (the face in this case).

4.6 PRINCIPLE COMPONENT ANALYSIS.

This algorithm was introduced to overcome the problem of overfitting, which occurs when a model tries to predict a trend in data that is too busy. It is a result of an overly complex model with too many parameters. A model that is over fitted is not accurate because it does not reflect the reality of the data. A model that has learned the noise instead of the signal is considered "over fit" because it fits the training dataset but has poor fit with new datasets.

The figure below shows an example of overfitting where the black line fits the data well but the green line is overfit.

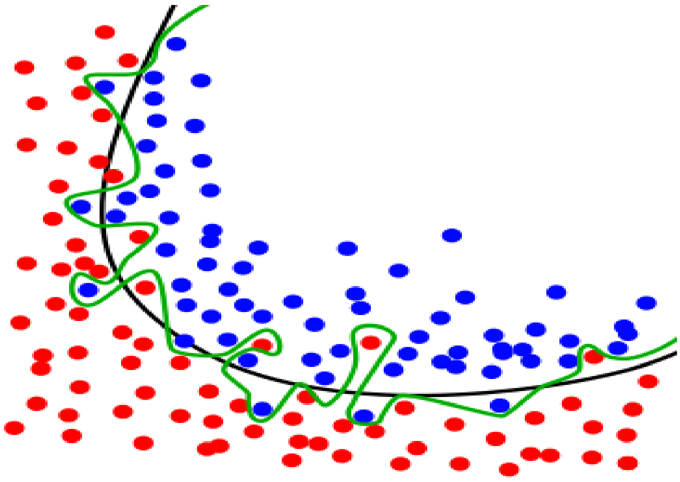


Figure 6. Example of over fitting

Usually, having a good amount of data lets us build a better predictive model since we have more data to train the machine with. It turns out that in large dimensional datasets, there might be lots of inconsistencies in the features or lots of redundant features in the dataset, which will only increase the computation time and make data processing and EDA more convoluted. To get rid of the curse of dimensionality, a process called dimensionality reduction was introduced. Dimensionality reduction techniques can be used to filter only a limited number of significant features needed for training and this is where PCA comes in. The main idea behind PCA is to figure out patterns and correlations among various features in the dataset. On finding a strong correlation between different variables, a final decision is made about reducing the dimensions of the data in such a way that the significant data is still retained. Such a process is very essential in solving complex data-driven problems that involve the use of high-dimensional data sets.

4.6.1 STEP BY STEP PROCESS OF PCA ALGORITHM.

Step 1: We start with the data set ‘A’ which is in the form of matrix of dimension m x n, where m rows represent the variables whereas n columns represent the samples i.e. observations. We will now linearly transform this matrix into another matrix ‘B’ of the same dimension m x n, so that for some matrix Z given by equation (1).

$$B = Z * A \tag{1}$$

Where, Z is another matrix of dimension m x m.

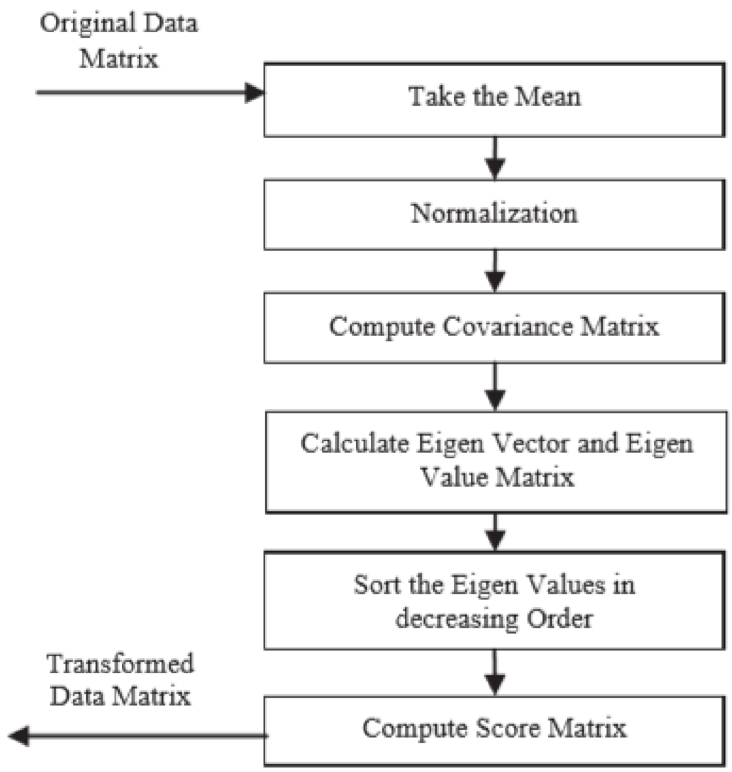


Figure 7. Flow chart for PCA algorithm

Step 2: Standardization of the Data

Normalization, also called standardization is all about scaling your data in such a way that all the variables and their values lie within a similar range. Consider an example, let's say that we have 2 variables in our data set, one has values ranging between 10-100 and the other has values between 1000-5000. In such a scenario, the output calculated by using these predictor variables is going to be biased since the variable with a larger range will have a more obvious impact on the outcome. Therefore, standardizing the data into a comparable range is very important. Standardization is carried out by subtracting each value in the data from the mean and dividing it by the overall deviation in the data set.

Normalization is the important part of the algorithm in which we need to calculate the mean of the original data matrix and subtract off the mean for finding principal components.

$$Mean\ (m) = 1/N \sum_{n=1}^{N} A[m, n] \qquad (2)$$

Step 3: Computing the Co-variance matrix

As mentioned earlier, PCA helps to identify the correlation and dependencies among the features in a data set. A covariance matrix expresses the correlation between the different variables in the data set. It is essential to identify heavily dependent variables because they contain biased and redundant information which reduces the overall performance of the model. Mathematically, a covariance matrix is a p × p matrix, where p represents the dimensions of the data set. Each entry in the matrix represents the covariance of the corresponding variables. Consider a case where we have a 2-Dimensional data set with variables a and b, the covariance matrix is a 2×2 matrix as shown below:

Cov(a, a) represents the covariance of a variable with itself, which is nothing but the variance of the variable 'a' Cov(a, b) represents the covariance of the variable 'a' with respect to the variable 'b'. And since covariance is commutative, Cov(a, b) = Cov(b, a) Here are the key takeaways from the covariance matrix:

The covariance value denotes how co-dependent two variables are with respect to each other. If the covariance value is negative, it denotes the respective variables are indirectly proportional to each other. A positive covariance denotes that the respective variables are directly proportional to each other.

Step 4: Calculating the Eigenvectors and Eigenvalues

Eigenvectors and eigenvalues are the mathematical constructs that must be computed from the covariance matrix in order to determine the principal components of the data set.

What are Principal Components?

Principal components are the new set of variables that are obtained from the initial set of variables. The principal components are computed in such a manner that newly obtained variables are highly significant and independent of each other. The principal components compress and possess most of the useful information that was scattered among the initial variables. If the data set is of 5 dimensions, then 5 principal components are computed, such that, the first principal component stores the maximum possible information and the second one stores the remaining maximum info and so on.

Now, where do Eigenvectors fall into this whole process?

Eigen vectors and Eigen values are the two algebraic formulations are always computed as a pair, i.e, for every eigenvector there is an eigenvalue. The dimensions in the data determine the number of eigenvectors that you need to calculate. Consider a 2-Dimensional data set, for which 2 eigenvectors (and their respective eigenvalues) are computed. The idea behind eigenvectors is to use the Covariance matrix to understand where in the data there is the most amount of variance. Since more variance in the data denotes more information about the data, eigenvectors are used to identify and compute Principal Components. Eigenvalues, on the other hand, simply denote the scalars of the respective eigenvectors. Therefore, eigenvectors and eigenvalues will compute the Principal Components of the data set.

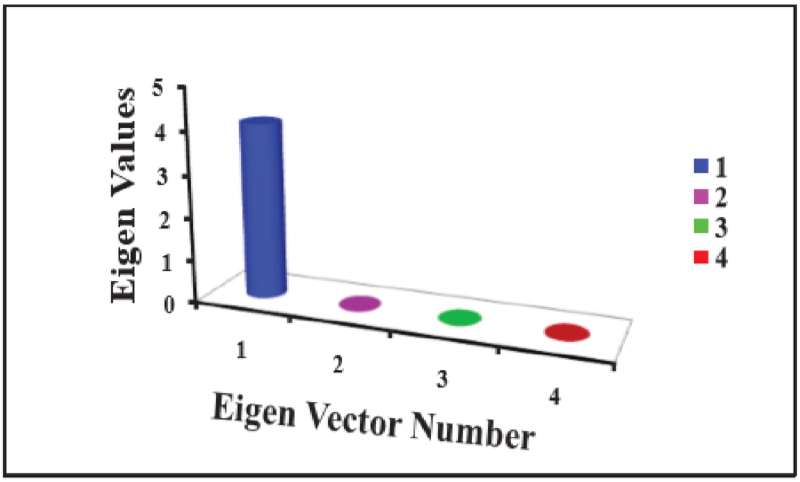


Figure 8. Eigen value spectrum

In fig 7, we have plotted the cylindrical graph for the Eigen values known as the Eigen value spectrum that provides a relationship between the Eigen values and Eigen vector number. Eigen vector number is the total number of Eigen values that are four in number for the given data matrix. All Eigen values are greater than one.

Step 4: Computing the Principal Components

Once we have computed the Eigenvectors and eigenvalues, all we have to do is order them in the descending order, where the eigenvector with the highest eigenvalue is the most significant and thus forms the first principal component. The principal components of lesser significances can thus be removed in order to reduce the dimensions of the data. The final step in computing the Principal Components is to form a matrix known as the feature matrix that contains all the significant data variables that possess maximum information about the data.

Step 5: Reducing the Dimensions of the Dataset

The last step in performing PCA is to re-arrange the original data with the final principal components which represent the maximum and the most significant information of the data set. In order to replace the original data axis with the newly formed Principal Components, you simply multiply the transpose of the original data set by the transpose of the obtained feature vector.

PCA is mathematically defined as an orthogonal linear transformation that transforms the data to a new coordinate system such that the greatest variance by some projection of the data comes to lie on the first coordinate (called the first principal component), the second greatest variance on the second coordinate, and so on.

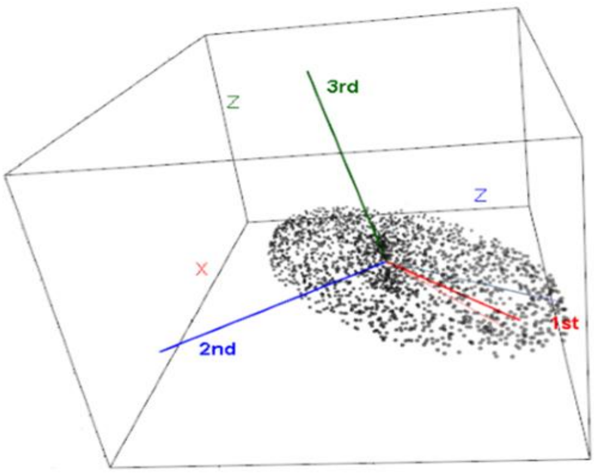


Figure 9. dimensional Geometric rationale of PCA

PCA objective is to rotate rigidly the coordinate axes of the p-dimensional linear space to new 'natural' positions (principal axes) such that: Coordinate axes are ordered such that principal axis 1 corresponds to the highest variance in data, axis 2 has the next highest variance, ..., and axis p has the lowest variance. The covariance among each pair of principal axes is zero, i.e. they are uncorrelated. The origin is the mean of the data points and the axes are provided by the eigenvectors.

Eigen-vectors are solutions of the eigen-equation Av = λv, where a (column) eigen-vector v is one of matrix A eigen-vectors and λ is one of eigen-values (which may be complex). The matrix A has n eigen-values λi and n eigen-vectors vi, i = 1,...,n. Let us derive: Av = λv ⇒ Av−λv = 0 ⇒ (A−λI)v = 0. Matrix I is the identity matrix. The equation (A−λI)v = 0 has the non-zero solution v if and only if det(A−λI) = 0. The polynomial det(A−λI) is called the characteristic polynomial of the matrix A. The fundamental theorem of algebra implies that the characteristic polynomial can be factored, i.e. det(A−λI) = 0 = (λ1−λ)(λ2−λ)...(λn−λ). . Multiple eigen-values arise from multiple roots of the characteristic polynomial.

$$\left.\begin{array}{rcrcrcr} x & + & 3y & - & 2z & = & 5 \\ 3x & + & 5y & + & 6z & = & 7 \\ 2x & + & 4y & + & 3z & = & 8 \end{array}\right\} \Longrightarrow A = \begin{bmatrix} 1 & 3 & -2 \\ 3 & 5 & 6 \\ 2 & 4 & 3 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 5 \\ 7 \\ 8 \end{bmatrix}.$$

$$[A|\mathbf{b}] = \left[\begin{array}{rrr|r} 1 & 3 & -2 & 5 \\ 3 & 5 & 6 & 7 \\ 2 & 4 & 3 & 8 \end{array}\right]$$

This system has a solution if and only if the rank of the matrix A is equal to the rank of the extended matrix [A|b]. The solution is unique if the rank of matrix (A) equals to the number of unknowns.

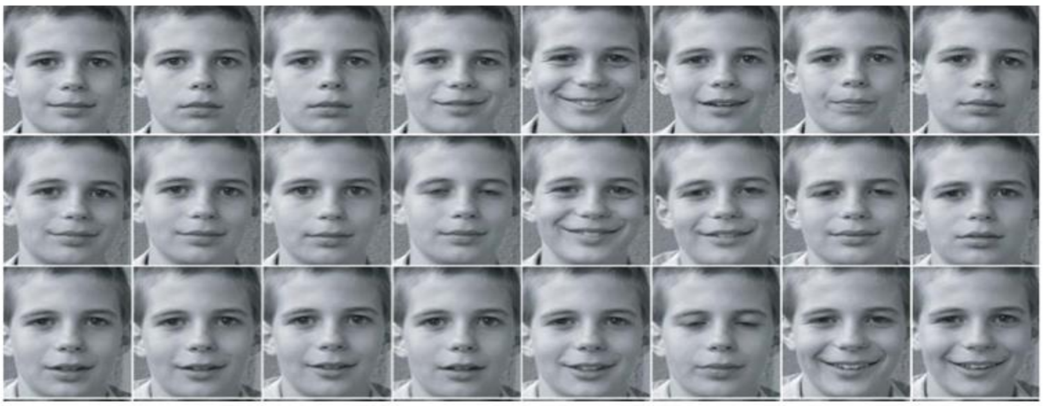


Figure 10. Image dataset of a person

Thus, new basis vectors are calculated for the particular data set. The aim is to reduce the dimensionality of the data so data normalization is needed.

Figure 11. Data gets normalized

4.7 SUPPORT VECTOR MACHINE

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.

The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine. Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyperplane:

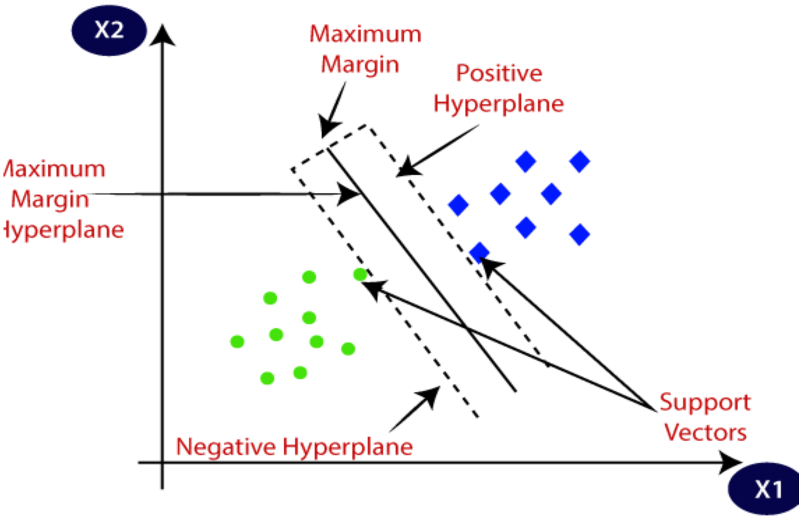


Figure 12. Segregation of two classes

Example: Suppose we see a strange cat that also has some features of dogs, so if we want a model that can accurately identify whether it is a cat or dog, so such a model can be created by using the SVM algorithm. We will first train our model with lots of images of cats and dogs so that it can learn about different features of cats and dogs, and then we test it with this strange creature. So as support vector creates a decision boundary between these two data (cat and dog) and choose extreme cases (support vectors), it will see the extreme case of cat and dog. On the basis of the support vectors, it will classify it as a cat. Consider the below diagram:

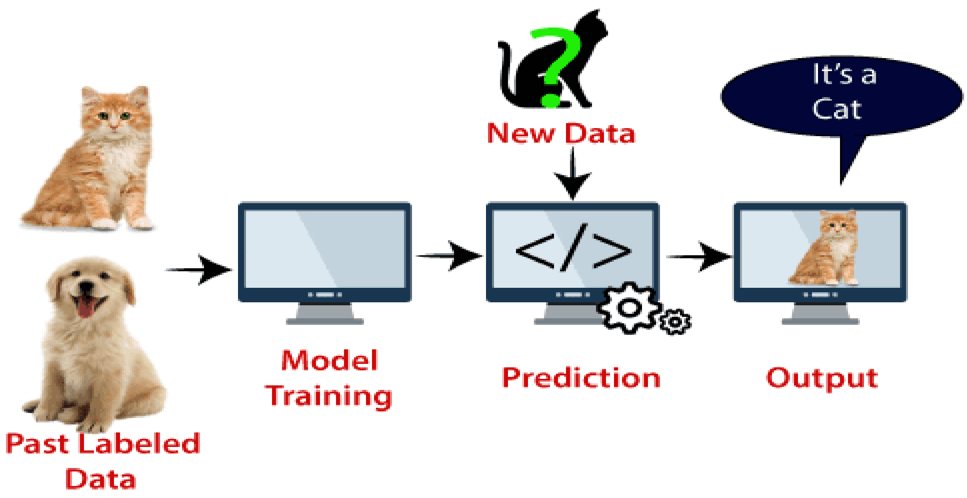


Figure 13. SVM classifier

4.7.1 TYPES OF SVM

SVM can be of two types:

1. Linear SVM: Linear SVM is used for linearly separable data, which means if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier is used called as Linear SVM classifier.

2. Non-linear SVM: Non-Linear SVM is used for non-linearly separated data, which means if a dataset cannot be classified by using a straight line, then such data is termed as non-linear data and classifier used is called as Non-linear SVM classifier.

4.7.2 HYPER PLANE AND THE SUPPORRT VECTOR IN SVM ALGORITHM

Hyperplane:

There can be multiple lines/decision boundaries to segregate the classes in n-dimensional space, but we need to find out the best decision boundary that helps to classify the data points. This best boundary is known as the hyperplane of SVM.

The dimensions of the hyperplane depend on the features present in the dataset, which means if there are 2 features (as shown in image), then hyperplane will be a straight line. And if there are 3 features, then hyperplane will be a 2-dimension plane.

We always create a hyperplane that has a maximum margin, which means the maximum distance between the data points.

Support Vectors:

The data points or vectors that are the closest to the hyperplane and which affect the position of the hyperplane are termed as Support Vector. Since these vectors support the hyperplane, hence called a Support vector.

4.7.3 HOW DOES SVM WORKS?

Linear SVM:

Suppose we have a dataset that has two support vectors (green and blue), and the dataset has two features x1 and x2. We want a classifier that can classify the pair (x1, x2) of coordinates in either green or blue. Consider the below image:

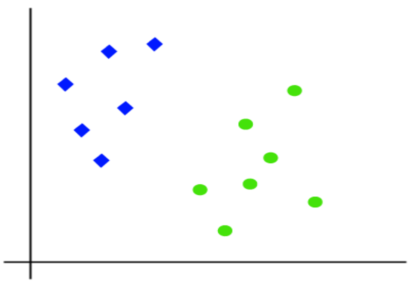


Figure 14. Example of classification

So as it is 2-d space so by just using a straight line, we can easily separate these two classes. But there can be multiple lines that can separate these classes. Consider the below image:

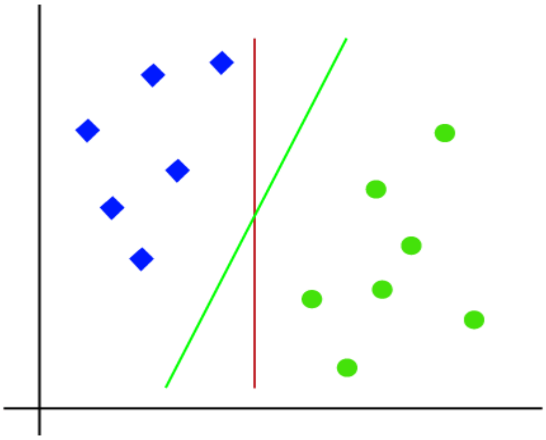


Figure 15. linear SVM classification

Hence, the SVM algorithm helps to find the best line or decision boundary; this best boundary or region is called as a hyperplane. SVM algorithm finds the closest point of the lines from both the classes. These points are called support vectors. The distance between the vectors and the hyperplane is called as margin. And the goal of SVM is to maximize this margin. The hyperplane with maximum margin is called the optimal hyperplane.

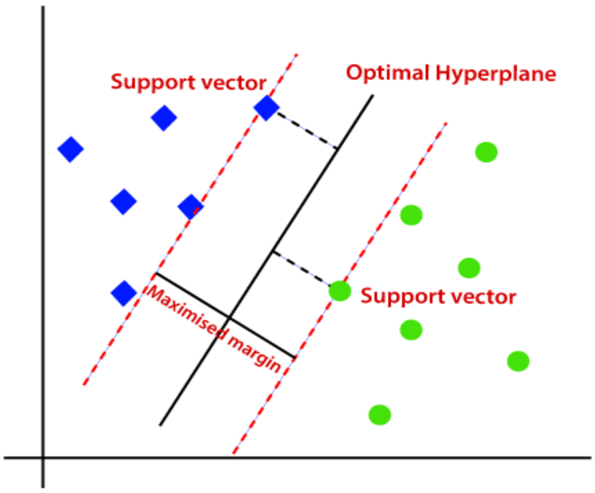


Figure 16. labelled linear classification

Non-Linear SVM:

If data is linearly arranged, then we can separate it by using a straight line, but for non-linear data, we cannot draw a single straight line.

Consider the below image:

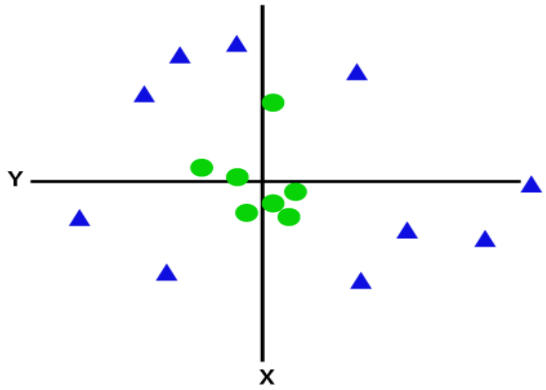


Figure 17(i). Non- Linear SVM classification

So to separate these data points, we need to add one more dimension. For linear data, we have used two dimensions x and y, so for non-linear data, we will add a third dimension z. It can be calculated as:

$$z=x^2 +y^2$$

By adding the third dimension, the sample space will become as below image:

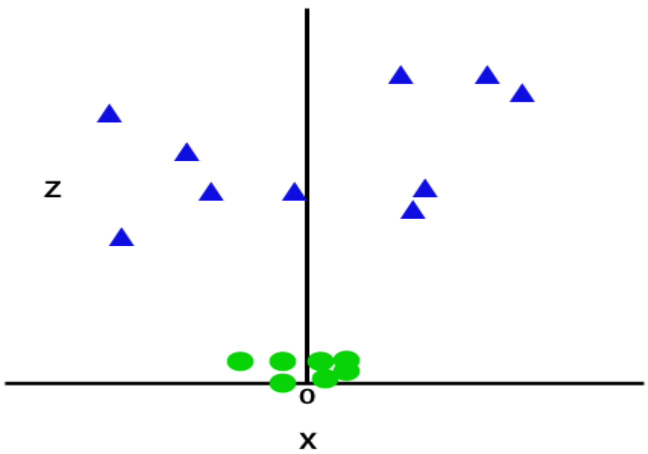


Figure 17(ii). Non- Linear SVM classification

So now, SVM will divide the datasets into classes in the following way. Consider the below image:

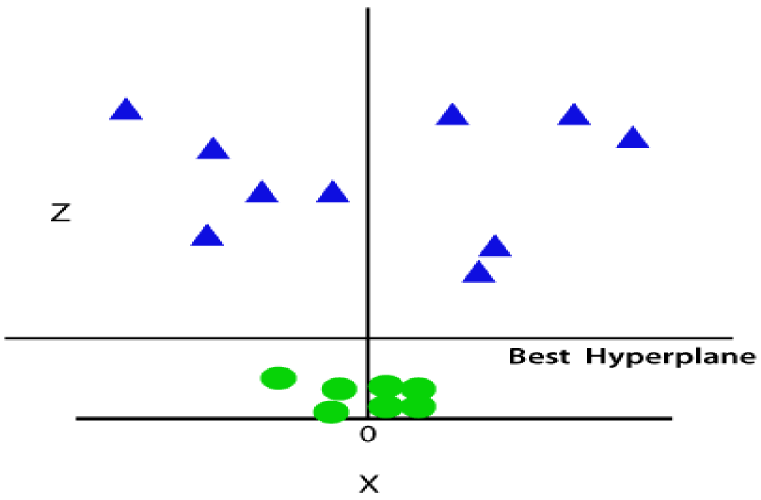


Figure 17(iii). Non- Linear SVM classification

Since we are in 3-d Space, hence it is looking like a plane parallel to the x-axis. If we convert it in 2d space with z=1, then it will become as:

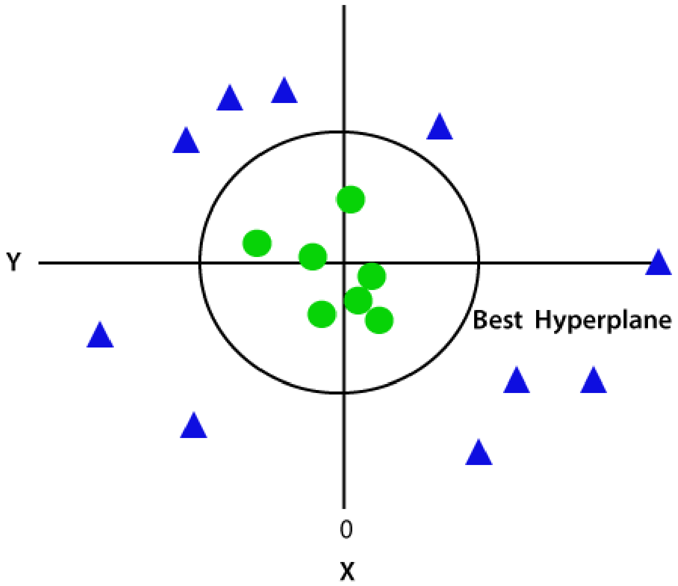


Figure 17(iv). Non- Linear SVM classification

Chapter 5

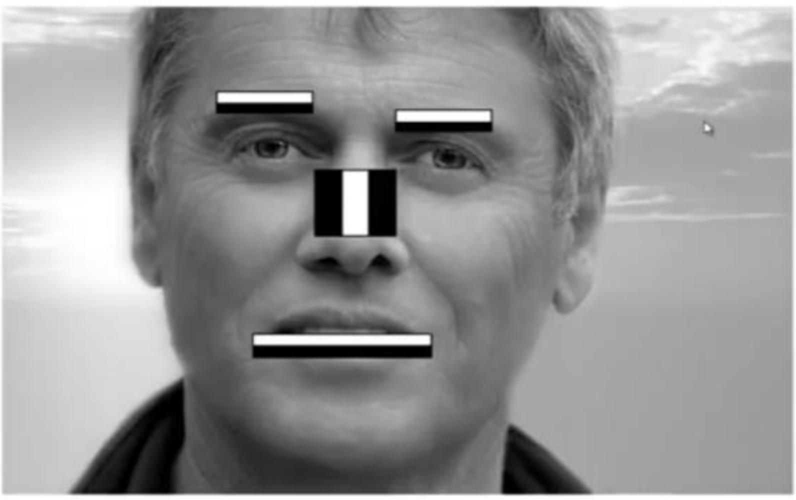
RESULTS AND DISCUSSION

5.1 IMAGE ACQUISITION AND GENERATION OF DATABASE

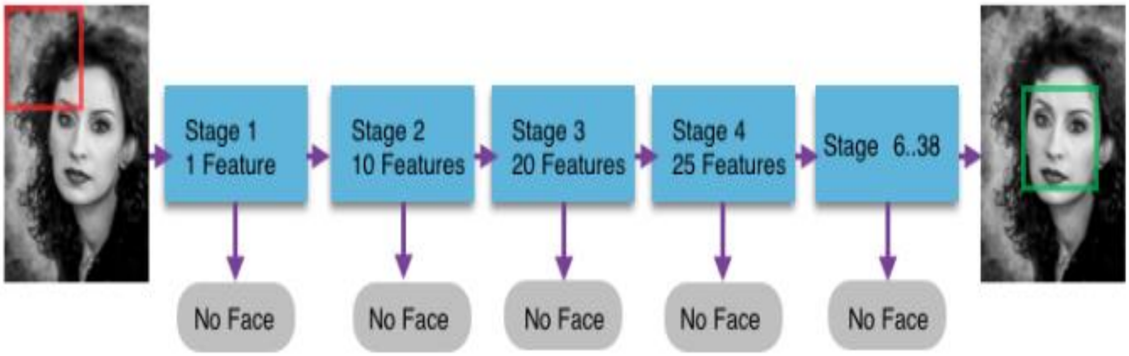
Video is being captured using the web cam and haar cascades works along with the web webcam to detect a region as a face. For this purpose, a window is defined as follows: -

Figure 18. Defining a window for face detection

Assuming the bottom left corner as the origin i.e. (0, 0) a window is defined. Haar filters starts searching for a region which matches the properties of fig.19(i) and continues its search in a linear manner as shown in fig 19.(ii). The region which matches with the properties of fig. 19(i), it detects it as a face.



(i)



(ii)

Figure 19. working of Haar filters.

The video streams are read in the form of images and gets stored in the desired location. There is one location for all the images of different faces. The system captures 101 images of each face and that is converted into gray scale. Different faces will be stored in the database with different faces Ids. For example, first face with face is = 1, second face with faces id = 2. Once all the images are captured, the camera will be released from the system. The next step is to segregate the different faces in respective folders in a different location.

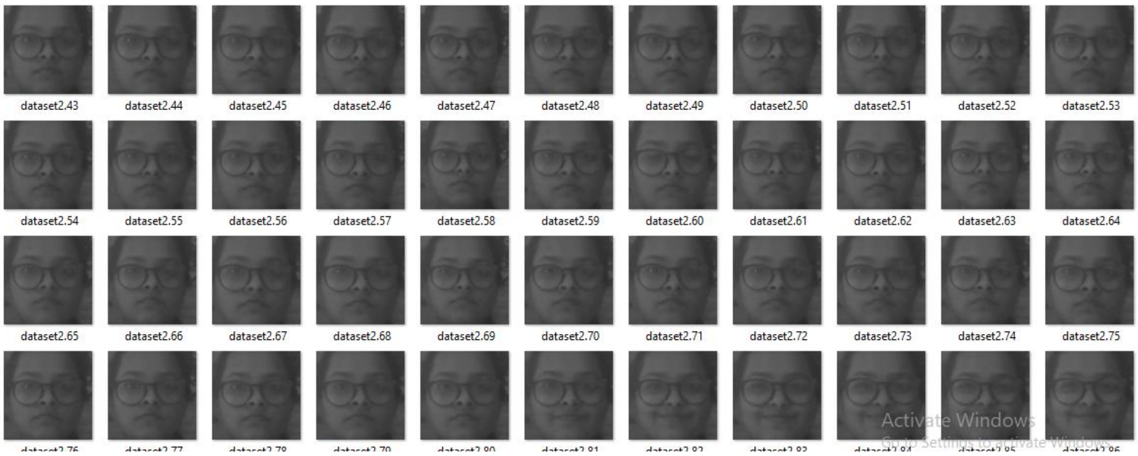


Figure 20. Dataset generation

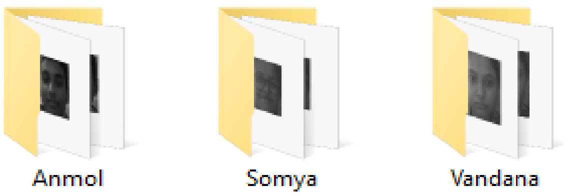


Figure 21. Segregation of faces.

5.2 CONVERTING THE DATASET INTO AN ARRAY.

Since the PCA algorithm works on the arrays, the dataset generated by image acquisition through web cam is used is first converted into an array. To perform array conversion, the data first needs to be converted into a vector by applying necessary methods.

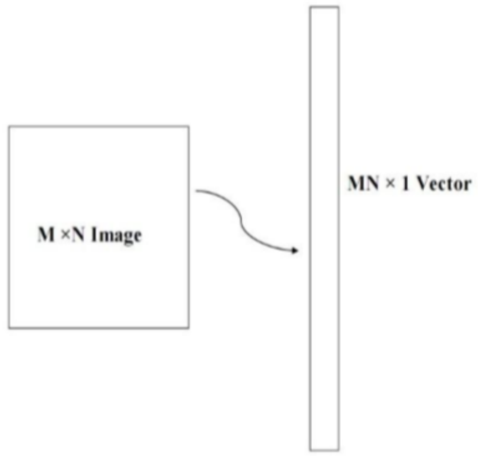


Figure 22. Converting dataset of image into array.

Necessary libraries are imported to perform this operation.

For example: -

LIBRARY	PURPOSE
Numpy	A general purpose array
Matplot	A plotting library for python programming
Logger	To use log files. Log files are the ones that records an event that occur in software. Logging is an act of keeping logs.

The method:-

logging.basicConfig(level=logging.INFO, format='%(asctime)s %(message)s')

is used to keep log files. The "level" parameter is used to set to set the root logger which is used to set it to the specified severity level. The format parameter sets the format of the log file. Empty lists are generated to first convert the data into a list and then the list is converted into an array using general purpose array, Numpy. For array conversion, data is first kept into a list and this list is converted into an array. Two lists are created, one for storing images and another for the names.

5.3 SPLIT ARRAY INTO TRAINING AND TEST SET.

The array is split into subarray to create test and train sets. The train set will consist of 75% of the total database and test set will consist of 25% of the total database from the array. This is done by the method "Train_Test_Split()". These subarrays are trained and tested with PCA components. It first extracts the Eigen faces out of total images present in the dataset. Then, pca=PCA(n_components=n_components, svd_solver='randomized',whiten=True).fit(X_train) is used to make instance of the model. It learns some quantities from the data, most importantly the components and variance. After that, transform() method is applied for dimensionality reduction, which keeps the dominant features from the obtained Eigen faces. This will be the principle component.

5.4 TRAINING SVM CLASSIFICATION MODEL.

Training of SVM model is done by two parameters:- kernel coefficient and penalty parameter.

Kernel coefficient is represented by gamma.

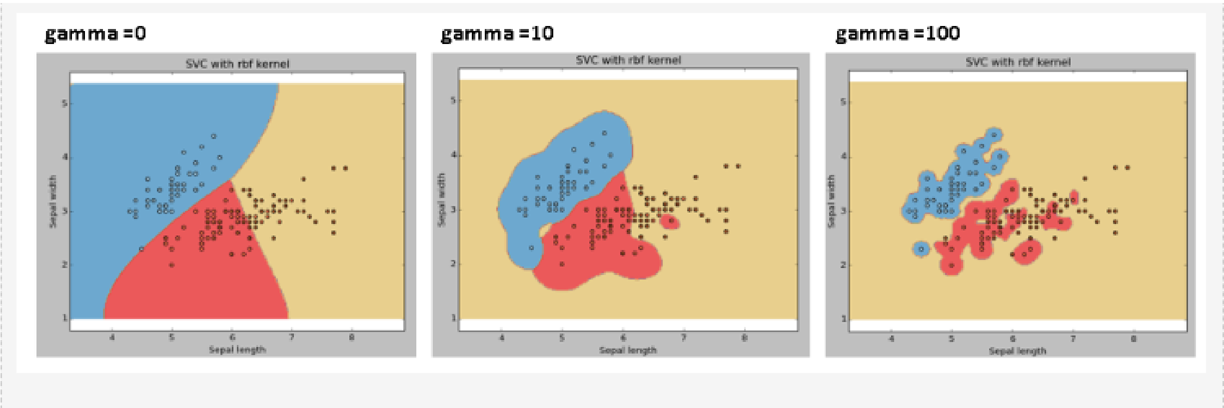


Figure 23. Significance of gamma

It plays a major role proper classification. The higher the value of gamma, it will try for exact fit as per the training dataset.

Penalty parameter is represented by C: -

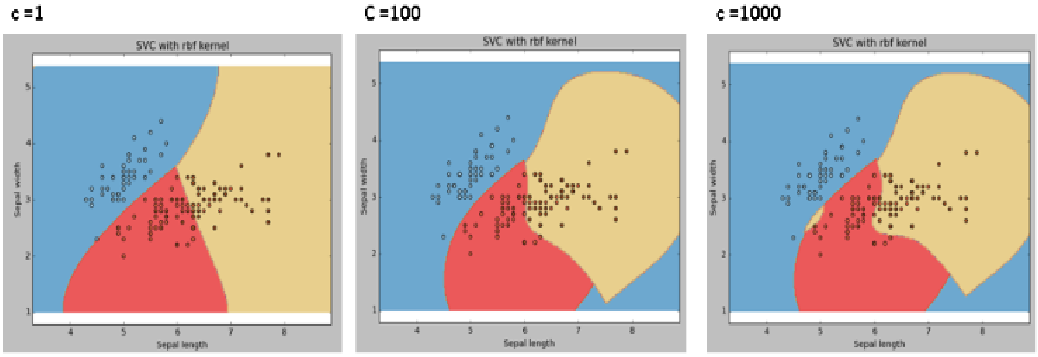


Figure 24. Significance of penalty parameter

It controls tradeoffs between smooth decision boundaries and classifying the training points correctly. A list for both these parameters is created (adjusted) and applied to SVM classifier with “rbf” kernel. After this, classifier’s performance is predicted with best_estimator. If it is very less than one means it shows bad performance. If it is near to 1, means it show good performance. And then, model accuracy is measured with “score()” method. The performance of the classifier is checked with classification report and confusion matrix.

5.5 CLASSIFICATION REPORT.

A classification report is a report representing the performance and quality of the classifier. The below is the figure of a classification report.

	precision	recall	f1-score	support
Anmol	0.93	1.00	0.96	27
Somya	1.00	0.96	0.98	24
Vandana	1.00	0.96	0.98	25
accuracy			0.97	76
macro avg	0.98	0.97	0.97	76
weighted avg	0.98	0.97	0.97	76

Figure 25. Classification Report

This classification report shows the number of the support vectors present for each faces. Here there are 27, 24, and 25 support vectors for Anmol, Somya and Vandana respectively. And the accuracy in this case is showing 0.97 which is 97%.

5.6 CONFUSION MATRIX

The confusion matrix displays the performance of the classifier. Below is the example of a confusion matrix of identifying whether the person has heart disease or not.

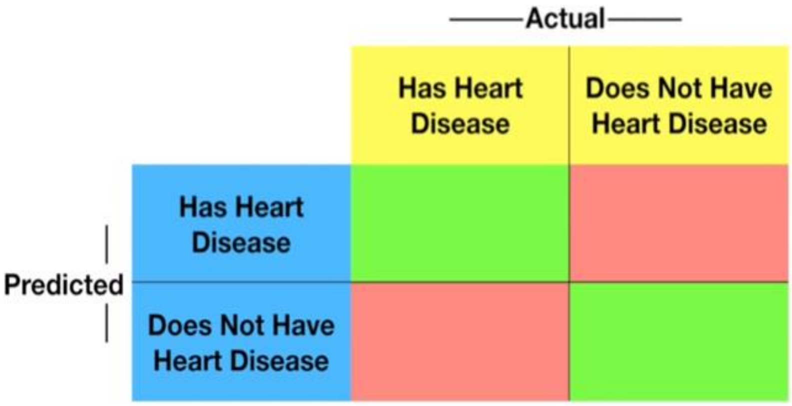


Figure 26. Confusion matrix

The green cells in the above matrix will display the number of true predictions where the predicted results matches with the actual result, and the red cells shows that the predicted results are actual results are contradicting each other. Means the person is not having a heart disease but the algorithm predicted that it has and vice versa.

5.7 EVALUATION OF MODEL QUALITY ON TEST SET.

The Quality of model in evaluated on the test set. Two variables are used i.e. predicted names and true names. The true names are directly taken from the test set and the predicted names are the ones which are obtained through PCA and SVM classification. The plot of all the sets of images is displayed which are present in dataset with true name and predicted names.

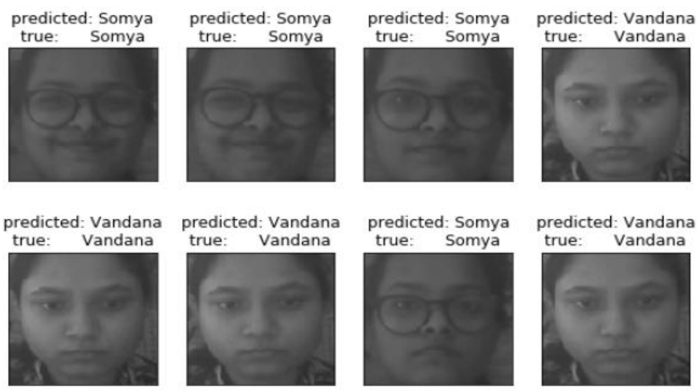


Figure 27. Outcome plot

5.8 COMPARISON WITH DIFFERENT DATASETS.

In this section the results are compared by taking different number of eigen faces and different number of faces. The classification report and confusion matrix is displayed for each case. The rows and columns are in the series of Anmol, Somya and Vandana.

5.8.1 DIFFERENT NUMBER OF EIGEN FACES.

For this comparison different number of eigen faces are taken when number of faces are kept three. Three cases are taken for three different eigen values.

Case 1: For n_components = 75

The classification report and confusion matrix for this case is shown below.

	precision	recall	f1-score	support
Anmol	0.96	1.00	0.98	27
Somya	1.00	0.96	0.98	24
Vandana	1.00	1.00	1.00	25
accuracy			0.99	76
macro avg	0.99	0.99	0.99	76
weighted avg	0.99	0.99	0.99	76

```
[[27  0  0]
 [ 1 23  0]
 [ 0  0 25]]
```

Figure 28. case for 75 eigen faces

The accuracy in this case is 99% and according to confusion matrix there is only one mismatch between Anmol and Somya.

Case 2: For n_components = 150

The classification report and confusion matrix for this case is show below.

	precision	recall	f1-score	support
Anmol	0.93	1.00	0.96	27
Somya	1.00	0.96	0.98	24
Vandana	1.00	0.96	0.98	25
accuracy			0.97	76
macro avg	0.98	0.97	0.97	76
weighted avg	0.98	0.97	0.97	76

```
[[27  0  0]
 [ 1 23  0]
 [ 1  0 24]]
```

Figure 29. (i)

In this case, the accuracy is 97% and the confusion matrix has two mismatches between Anmol - Somya and Anmol – Vandana.

Figure 29. (ii) case for 150 eigen faces

Case 3: For n_components = 200

The classification report and confusion matrix for this case is shown below.

	precision	recall	f1-score	support
Anmol	0.92	0.89	0.91	27
Somya	0.50	0.96	0.66	24
Vandana	1.00	0.16	0.28	25
accuracy			0.67	76
macro avg	0.81	0.67	0.61	76
weighted avg	0.81	0.67	0.62	76

```
[[24  3  0]
 [ 1 23  0]
 [ 1 20  4]]
```

(i)

Here the accuracy is 67%. And according to the confusion matrix there are six mismatches.

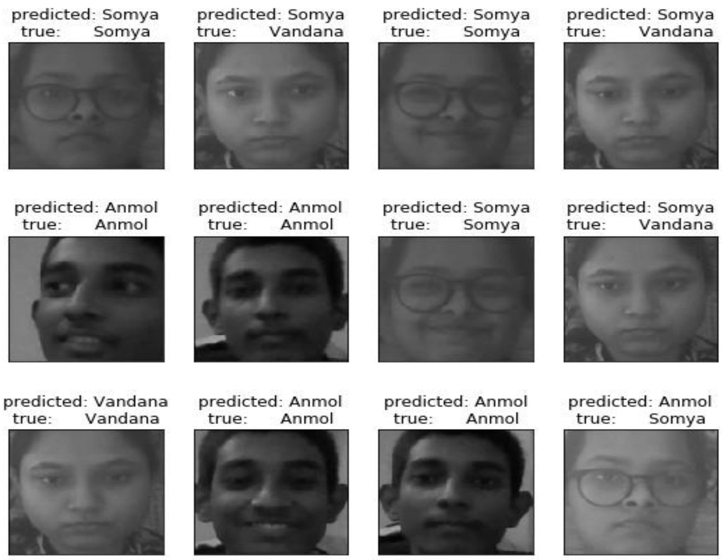


Figure 30 (ii). case for 200 eigen faces

In the above plot, we can see that there are multiple images where predicted name does not match with the true name.

From the above three cases we can conclude that as the number of Eigen faces increases, the accuracy of the SVM classifier decreases. This is because when more Eigen faces are present, there will be more number of features for classification and the system may get confused between the faces. We can maintain the desired accuracy by keeping the number of Eigen faces low.

5.8.2 DIFFERENT NUMBER OF FACE IDs.

In this section, the accuracy is compared where different number of faces which are present.

Case 1: Three faces with 75 eigen faces.

The classification report and confusion matrix is shown in the figure below.

	precision	recall	f1-score	support
Anmol	0.96	1.00	0.98	27
Somya	1.00	0.96	0.98	24
Vandana	1.00	1.00	1.00	25
accuracy			0.99	76
macro avg	0.99	0.99	0.99	76
weighted avg	0.99	0.99	0.99	76

(i)

In this case, that is when three faces are present the accuracy obtained is 99%.

Case 2: Four faces with 75 Eigen faces.

In this case, four faces are present with 75 Eigen faces on total. The corresponding classification report and confusion matrix is shown.

Anmol	1.00	1.00	1.00	30
Happy	1.00	1.00	1.00	24
Somya	1.00	1.00	1.00	17
Vandana	1.00	1.00	1.00	30
accuracy			1.00	101
macro avg	1.00	1.00	1.00	101
weighted avg	1.00	1.00	1.00	101

(ii).

Figure 31. Comparison with different number of faces

The accuracy obtained in this case is 100% that all the matches are successful.

From the above method, we can conclude we can maintain the desired accuracy by keeping the count of Eigen faces low. we have seen that the accuracy was decreasing when the count of Eigen faces were increasing. Also, cases are shown when the number of faces increases keeping the Eigen value constant. With three faces, Accuracy was less compared to when there were four faces. With these results, we can conclude that the count of Eigen faces plays a vital in deciding the accuracy of the classification. It has to be adjusted such that it is not too low when the classifier does not get enough faces for classification and too high when it may get confused.

Chapter 6

CONCLUSION

This system uses Haar filters for face detection which can properly detect a face from an image. PCA algorithm gives Eigenfaces with unique feature for each face and works with SVM algorithm for classification between them. Eigenfaces, being the dominant feature of a person gives better results while comparing with the faces present in the dataset. SVM performs the classification with the Eigenfaces in a nonlinear environment and predict the name of a person based on classification results. Eigenfaces play a major role in deciding the classifier accuracy. It has to be adjusted such that it is not too low when the classifier does not get enough faces for classification and too high when it may get confused. The suggested system gives better accuracy for unique identification due to Eigenfaces and their classification with SVM classifier.

Chapter 7

SCOPE FOR FUTURE WORK

The project can be improvised by making the identification process happening in real time. For example, a person stands in front of the camera, and the system should identify the person and display is name after required processing. It can also be used in a public place, for example, a mall where if a person comes for billing, it should display the due amount to be paid by him. It can also be used as a system which identifies a person and displays details about him/her (details such as address, profession, salary… etc). Or it can be also used as system locker, for example, only the identified person will be able to open a lock.

REFERENCES

[1]. Rapid Object Detection using a Boosted Cascade of Simple Features

[2]. Classification Mechanism of Support Vector Machines Chen Junli Jiao Licheng Key Lab. for Radar Signal Processing, Xidian Univ., Xi’an 710071, China.

[3]. Article on Applying the Haar-cascade Algorithm for Detecting Safety Equipment in Safety Management Systems for Multiple Working Environments.

[4]. Overfitting and Underfitting Analysis for Deep Learning Based End-toend Communication Systems. School of Electronics and Information Technology, Sun Yat-sen University, Guangzhou, China.

[5]. Evaluation of face recognition technique using PCA wavelets and SVM. Istanbul university, engineering faculty, computer engineering department, 34320, Avcilar, Istanbul, Turkey.

[6]. Data Analysis Using Principal Component Analysis, 2014 International Conference on Medical Imaging, m-Health and Emerging Communication Systems (MedCom).

[7]. Face Detection and Recognition using Support Vector Machine, International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 – 8958, Volume-8 Issue-4, April, 2019.

[8]. Data classification using support vector machine, Journal of Theoretical and Applied Information Technology

[9]. Real Time Face Recognition System (RTFRS), 4th International symposium on digital forensics and security, Little Rock AR.

[10]. Face Recognition Using Principal Component Analysis Method, International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 1, Issue 9, November 2012.

APPENDIX

```
%matplotlib inline
import pickle
import numpy as np

# Import OpenCV2 for image processing
#!pip install opencv-python
import cv2
# Below code will trigger the camera and create database

# In[1]:
# Start capturing video
vid_cam = cv2.VideoCapture(0)

# Detect object in video stream using Haarcascade Frontal Face
face_detector=cv2.CascadeClassifier('C:\\Users\\hppc\\Downloads\\haarcascade_frontalface_
default.xml')

# For each person, one face id
face_id = 1                    #Changed for each face ID

# Initialize sample face image
count = 0

# Start looping
while(count<200):

    # Capture video frame
    _, image_frame = vid_cam.read()

    # Convert frame to grayscale
    gray = cv2.cvtColor(image_frame, cv2.COLOR_BGR2GRAY)

    # Detect frames of different sizes, list of faces rectangles
    faces = face_detector.detectMultiScale(gray, 1.3, 5)

    # Loops for each faces
    for (x,y,w,h) in faces:

        # Crop the image frame into rectangle
        cv2.rectangle(image_frame, (x,y), (x+w,y+h), (255,0,0), 2)
```

```
        # Increment sample face image
        count += 1

        # Save the captured image into the datasets folder
        cv2.imwrite("D:\\project\\dataset"+str(face_id)+'.'+str(count)+".jpg",gray[y:y+h,x:x+w])

        # Display the video frame, with bounded rectangle on the person's face
        cv2.imshow('frame', image_frame)

    # To stop taking video, press 'q' for at least 100ms
    if cv2.waitKey(100) & 0xFF == ord('q'):
        break

    # If image taken reach 100, stop taking video
    elif count>100:
        break

# Stop video
vid_cam.release()

# Close all started windows
cv2.destroyAllWindows()

from time import time
import logging
import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split
from sklearn.model_selection import GridSearchCV

from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.decomposition import PCA
from sklearn.svm import SVC


print(__doc__)

# Display progress logs on stdout
logging.basicConfig(level=logging.INFO, format='%(asctime)s %(message)s')


###############################################################################
```

```
import glob
import re

data = []
labs = []

for name in glob.glob('C:\\Users\\Lenovo\\Desktop\\project\\dataset\\*\\*.jpg'):
    img = cv2.imread(name)
    img = cv2.resize(img, (128,128))
    data.append(img.flatten())
    lab = re.search(r'\\([a-zA-Z]+)\\([a-zA-Z0-9.]+).jpg', name).group(1)
   # lab = name.split(".")[0].split('\\')[-1]
    labs.append(lab)
    print(name)
    print(lab)
    print(img.shape)

print('Number of files read:', len(data))

import numpy as np
X = np.array(data)
y = np.array(labs).reshape(-1,1)

X.shape
###############################################################################
# Split into a training set and a test set using a stratified k fold

# split into a training and testing set
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.25, random_state=42)


###############################################################################
# Compute a PCA (eigenfaces) on the face dataset (treated as unlabeled
# dataset): unsupervised feature extraction / dimensionality reduction
n_components = 25

print("Extracting the top %d eigenfaces from %d faces"
      % (n_components, X_train.shape[0]))
t0 = time()
pca = PCA(n_components=n_components, svd_solver='randomized',
          whiten=True).fit(X_train)
print("done in %0.3fs" % (time() - t0))
```

```
print("Projecting the input data on the eigenfaces orthonormal basis")
t0 = time()
X_train_pca = pca.transform(X_train)
X_test_pca = pca.transform(X_test)
print("done in %0.3fs" % (time() - t0))

###############################################################################
# Train a SVM classification model

print("Fitting the classifier to the training set")
t0 = time()
param_grid = {'C': [1e3, 5e3, 1e4, 5e4, 1e5],
            'gamma': [0.0001, 0.0005, 0.001, 0.005, 0.01, 0.1], }
clf = GridSearchCV(
    SVC(kernel='rbf', class_weight='balanced'), param_grid)

clf = clf.fit(X_train_pca, y_train) # Train the Machine learning model
print(y_train)
print("done in %0.3fs" % (time() - t0))
print("Best estimator found by grid search:")
print(clf.best_estimator_)
print(clf.score(X_test_pca, y_test))

y_pred = clf.predict(X_test_pca)

###############################################################################
# Quantitative evaluation of the model quality on the test set

print("Predicting people's names on the test set")
t0 = time()
# y_pred = clf.predict(X_test_pca)
y_pred = clf.predict(X_test_pca)

print("done in %0.3fs" % (time() - t0))

target_names=['Anmol','Somya','Vandana']
n_classes = 3
y_test1=[name[0] for name in y_test]
print(classification_report(y_test, y_pred, target_names=target_names))
print(confusion_matrix(y_test1, list(y_pred), labels=target_names))


###############################################################################
# Qualitative evaluation of the predictions using matplotlib
```

```
def plot_gallery(images, titles, h, w, n_row=3, n_col=4):
    """Helper function to plot a gallery of portraits"""
    plt.figure(figsize=(1.8 * n_col, 2.4 * n_row))
    plt.subplots_adjust(bottom=0, left=.01, right=.99, top=.90, hspace=.35)
    for i in range(n_row * n_col):
        plt.subplot(n_row, n_col, i + 1)
        plt.imshow(images[i].reshape((h, w, 3)), cmap=plt.cm.gray)
        plt.title(titles[i], size=12)
        plt.xticks(())
        plt.yticks(())

# plot the result of the prediction on a portion of the test set

def title(y_pred, y_test, target_names, i):
    pred_name = y_pred[i]
    true_name = y_test[i]
    return 'predicted: %s\ntrue:      %s' % (pred_name, true_name)



prediction_titles = [title(y_pred, y_test1, target_names, i) for i in range(len(y_pred))]

print(prediction_titles)
plot_gallery(X_test, prediction_titles, 128, 128)
```

2nd International Conference on
Inventive Research in Computing Applications
(ICIRCA 2020)

ICIRCA 2020

15-17, July 2020 | http://www.icirca18.com/2020 | icirca2018@gmail.com

ACCEPTANCE LETTER

Paper ID: **ICIRCA0273**

To:
Vaishnavi Chawda , Vandana Arya, Shuchi Pandey, Shristi,
Manohar Valleti .Assistant Professor,
Electronics and Communication Eng.
CMR Institute of Technology, Bengaluru, India.

Title: **Unique Face Identification System using Machine Learning**

Acceptance Letter –2nd INTERNATIONAL CONFERENCE ON INVENTIVE RESEARCH IN COMPUTING APPLICATIONS- Reg

Dear Author,

International Conference on Inventive Research in Computing Applications (ICIRCA 2020) is being organized on 15-17, July 2020. ICIRCA 2020 will provide an outstanding international forum for sharing knowledge and results in all fields of engineering and technology. It provides an opportunity for the quality key experts to bring up innovative ideas. Recent updates in technology will be a platform for the upcoming researchers. The conference will be Complete, Concise, Clear and Cohesive in terms of research related to Innovative Mechanisms for Industrial Needs. The Organizing Committee is pleased to inform you that the following peer-reviewed & refereed conference paper has been accepted for Oral presentation at the International Conference on Inventive Research in Computing Applications (ICIRCA 2020 on15-17, July 2020.

All Presented and Registered papers will be submitted for inclusion into IEEE Xplore

Yours sincerely,

Dr. S. Smys,
Organizing secretary
ICIRCA – 2020

Proceedings by IEEE

Previous Publication Links
2018 publication link

Unique Face Identification System using Machine Learning

Vaishnavi Chawda[1], Vandana Arya[2], Shuchi Pandey[3], Shristi[4], Manohar Valleti[5]

[1,2,3,4,5]Department of Electronics and Communication Engineering
CMR Institute of Technology, Bengaluru, 560037, India.
[1]vach16ec@cmrit.ac.in, [2]vaar16ec@cmrit.ac.in, [3]shpa16te@cmrit.ac.in, [4]shti16ec@cmrit.ac.in
[5]*Assistant Professor*: manohar.v@cmrit.ac.in

Abstract— **The recognition of human faces plays an important role in many applications, for example in video surveillance and the management of facial image databases. This paper will design and implement a security system based on a machine learning algorithms. Principal Component Analysis (PCA) is the algorithm that represents the faces economically. It extracts the most dominant Eigenfaces from the present set of the faces. Comparison of video frames can be done by using this technique. Faces can be recognized in frames using the haar cascade to extract the characteristics of a human face. The SVM algorithm is used to classify between data sets using the kernel. The performance of the identification system also depends on the extraction of the attributes and their classification in order to obtain accurate results. These algorithms give different accuracy rates under different conditions, as observed experimentally. The precision and efficiency with which the model identifies people is the real added value of this paper.**

Keywords—ML, Facial recognition, Haar cascade, PCA, SVM, Eigen faces.

I. Introduction

Machine learning is an area derived from artificial intelligence (AI). AI created modern and intelligent machines. With the quick development of artificial intelligence in past few years, face recognition (FR) is gaining more attention. Compared to traditional fingerprint recognition, iris recognition and card recognition, FR offers numerous advantages, those are high parallelism, non-contact and ease of use. Facial knowledge in biometric software can accurately identify people by comparing and analyzing face based models. It is used mainly for safety and security. In fact, face-to-face technology has obtained considerable recognition as it has a broad range of applications [1].

With the help of ML, our own neural network did not have to be set up and have access to a trained model that can be used [4]. It does exactly what a series of numbers (face encodings) need to be printed when the image of someone's faces is passed; If you compare the face encodings of faces from different images, you can determine whether a person's face matches a person pictures of [1].

Various face recognition (FR) techniques are used, such as common face detection methods and adaptive regional mixed matching methods. Most of the features of the FR system are based on various nodes in the human face. Measured values for face-related variables help to uniquely identify a person. This technology allows the application to accurately and quickly identify the target individual using face-to-face data.

The recognition of the face has many advantages. It is contactless compared to other biometric techniques. Face images can be taken and analyzed remotely without ever requiring user / person interaction. No user can successfully imitate another person as a unique machine learning face recognition system. Face recognition can serve as an excellent safety measure for recording time and attendance. Face recognition is also a low-cost technology, as it requires less processing than other biometric techniques. Face recognition technology is rapidly evolving with new approaches to 3D modeling to overcome the problems of existing technology [12].

Face recognition has great potential for use in government agencies, public institutions, security, e-commerce, retail, education and many other areas. More and more security incidents have triggered a feeling of insecurity among individuals and companies. They do what is feasible and affordable for them to feel more secure. The installation of security systems at home and in the office is a first protective layer against security threats. The present systems, which are used to recognize faces from videos, are more expensive. The traditional method of recognizing and identifying people is a manual method that is very time consuming and requires more effort for people. In this project, a security system is designed and implemented based on machine-learning algorithms. Therefore, designing and implementing a system is suggested that identifies and recognizes faces using effective techniques that can be most effective for humanity.

The remaining paper is planned as follows. In the next section, a Literature review has been discussed. Section 3 talks about various stages, starting with Haar Cascade, Dataset generation followed by Principle Component Analysis and Support Vector Machine. In Section 4, the experimental results have been detailed out. And the last section is the conclusion.

II. Literature review

The unique technology of the identification system was originally developed for the purpose of security surveillance in order to anticipate various functions such as detection of crime, theft, robbery, etc. The present systems, which are used to recognize faces from videos, are more expensive. The traditional method of recognizing and identifying people is a manual method that is very time consuming and requires more effort for people.

In the domain of FR, different tools have been described and implemented. Principle component analysis (PCA) is one of the first effective and best analyzes approach to the FR domain [13]. This approach takes the entire picture as a vector and uses it to produce statistical information. This brings all of the image vectors together and creates an image matrix and then the Eigenvectors of this equation will be calculated. The images on the face can then be represented as a linear solution.

Teddy M et al. says the process of FR with the Haar Cascade and Eigenface method is capable of optimizing facial recognition with more than one face with good accuracy [14]. Support vector machines (SVMs) based on statistical learning theory have been proposed as a new intelligence technique for both regression and classification tasks. The design of SVM follows the concept of structural risk minimization (SRM), which has been shown to be superior to the conventional theory of empirical risk minimization (ERM) used by neural networks [16].

III. Methodology

When a camera is operating it captures images of the person to create a database. The further process can be carried out by preparing the dataset, recognizing faces and pre-processing the taken picture [11]. The technique starts when the basic image is taken. The next process followed is database development, face recognition, preprocessing, feature extraction, and classification phases. Faces can be recognized from frames by using Haar cascade to extract specific facial features. Haar Cascade is basically a classifier used to determine objects that are trained from the source. Haar sequences are trained by superimposing positive images on a series of negative images [2]. The features are extracted using PCA, which are known as Eigenfaces. These Eigenfaces are compared with images present in the database [7]. Support Vector machine (SVM) performs the classification with the Eigenfaces. The performance of an identification system also depends on the extraction of the characteristics and their classification in order to obtain exact results [10]. By extracting and classifying facial features, a person can be identified based on the train and test sets prepared by the algorithm [7].

A. *Haar cascades*

Real-time video identification system identifies and detects faces from a stream of images using Haar cascade filters. When the camera starts to work, Haar filters work with the camera to detect a face from the entire region of the image. Fig. 1 shows some of the Haar filters which are used for detecting faces [8].

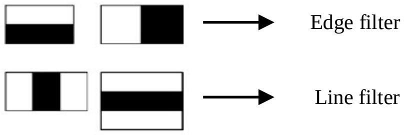


Fig. 1. Haar filters

.

The Haar filters works on black and white pixels, so the image must be changed to a black and white image. Fig. 2 shows how these filters work on the image to detect a face. A rectangular region is defined as a search operation. This box starts to search from the top left corner of the image and proceeds in a linear manner [2].

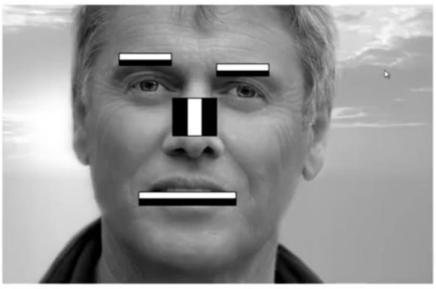


Fig. 2. Haar filters for face detection

As shown in fig. 2 human eyebrows were detected using horizontal line filters. The eyebrows were found because the hair color has darker pixels and the surrounding area has lighter pixels. As well as the nose is composed of brighter pixels on the length of the nose and dark pixels on the surrounding area, hence, it is identified by vertical line filters. The system successfully detects a face when all such features like eyebrows, nose and lips are detected inside the defined rectangular box [2].

B. *Dataset generation*

In this experiment, each face with a unique id is taken with multiple images and are stored in the desired location. To generate a dataset for each id, a total number of different images taken is 101. For example, a person has a face Id as 1, will have 101 images stored in the database of the system. Below figure shows the dataset that is generated by taking multiple images.

Fig. 3. Dataset generation

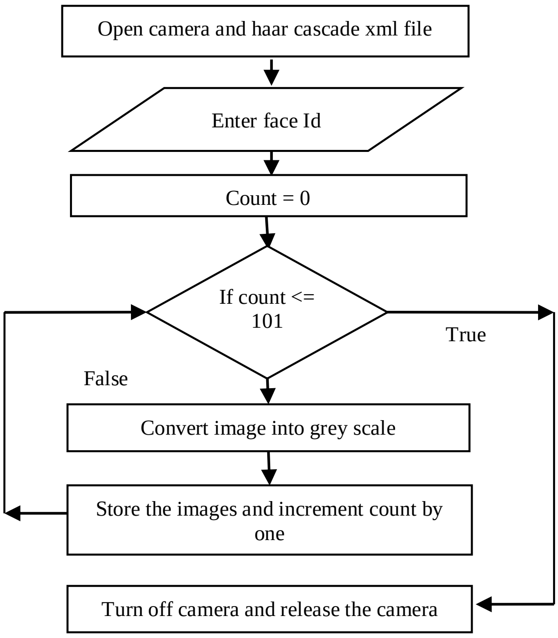


Fig. 4. Flow chart for dataset generation

Each image will have different facial expression of the person so that the person will be identified even when he is changing his facial expressions.

C. Applying Principle Component Analysis algorithm

This algorithm was introduced to overcome the overfitting problem that occurs when a model tries to predict a trend in very crowded data. This is the result of a very complex model with many parameters [6]. A model that studies noise rather than signal is considered "overfit" because it matches the training dataset but is not suitable for new datasets [3]. Here PCA is used in the form of dimensionality reduction for extracting Eigenface. Eigenfaces are the facial images which contain dominant features of a face as shown in fig. 5. Steps for computing PCA algorithm are described below [7].

1) Calculating mean of faces

The first step is to calculate the mean of images which takes the features shared by all the images. Consider a set of faces:

$$X = \{x_1, x_2, x_3, ..\} \tag{1}$$

Where, x = faces captured
X = set of faces captured for a single person

2) Normalization

Faces are normalized by subtracting the mean face from each of the faces. These normalized faces give unique features which are nothing but the Eigenfaces. Normalized value is calculated as follows.

$$q = x - m \tag{2}$$

Where, q = unique feature of a face / Eigenface
x = face capture
m = mean face

$$A = \{q_1, q_2, q_3 \ ...\} \tag{3}$$

Where, q = normalized face with unique feature / Eigenfaces
A = set of Eigenfaces

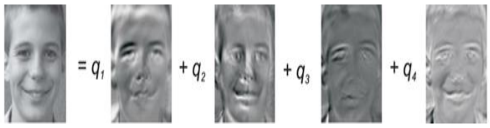


Fig. 5. Eigen faces

3) Creation of Eigenvector

Eigenvector is computed from the covariance matrix which is computed as:

$$C = AA^T \quad \text{where } A = \{q_1, q_2, q_3 \ ...\} \tag{4}$$

Where, A = Eigen faces.
A^T = transform of A.
C = Covariance matrix

Arranging the matrix in decreasing order, topmost vector will be having, the Eigenvector with highest Eigen value. This will be final principle component. Once the Eigenfaces are obtained it will be split into train and test datasets. The training and testing datasets contains 75% and 25% of the total available Eigenfaces respectively.

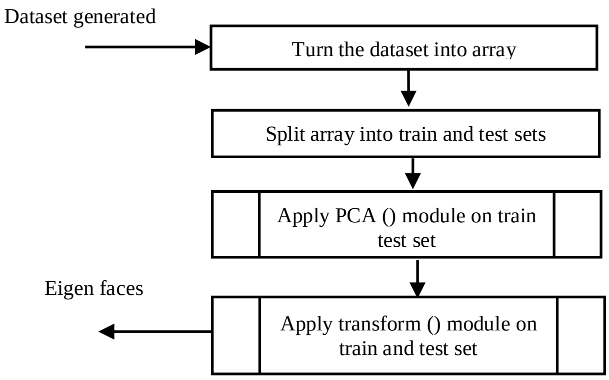


Fig .6. Flow chart for PCA

D. Support Vector Machine

SVM is used for classification between the Eigenfaces. The classifier searches for support vector which are Eigen faces here, just near to the support-vector of the other class. This process is done for all the classes and a boundary is drawn in between which is known as a hyperplane. As shown in the fig. 7 and fig. 9 there is a positive and negative hyperplane. The distance between these two should be such that it should have maximum margin and the corresponding hyperplane is called a maximum hyperplane. This algorithm is specifically called linear support vector machine (LSVM) because the classes can be easily classified by drawing a straight line as shown in fig.7. For some data sets, it's almost impossible to draw a single line to classify the classes, for this a transformation is applied on the dataset. A function can be used to transform our data into a higher dimensional feature space [9].

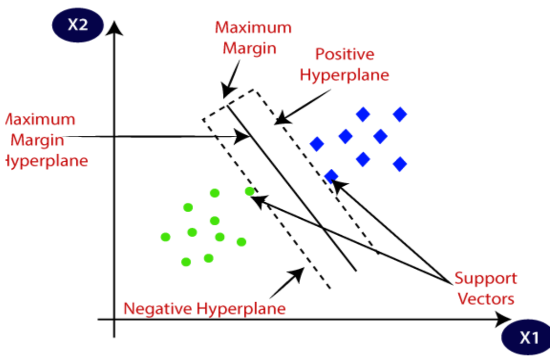


Fig. 7. Segregation of two classes using SVM

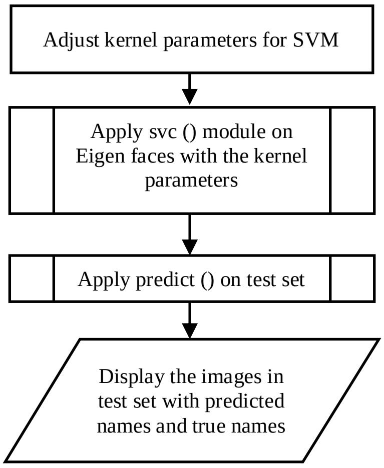


Fig .8. Flow chart for SVM

A kernel function also called a kernel trick, is defined as a function that uses inputs as a vector in the original space and returns the product point vector in the feature space. This core function is used to apply the point product between two vectors so that each point is assigned to a larger dimension space through multiple transformations [1], as shown in the fig. 10.

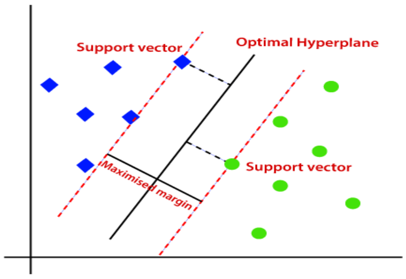


Fig. 9. Linear classification

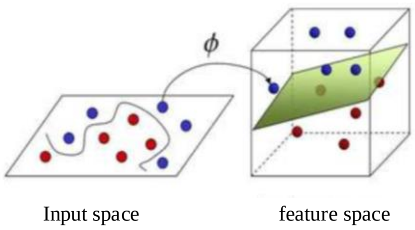


Fig. 10. Transformation for non-linear classification

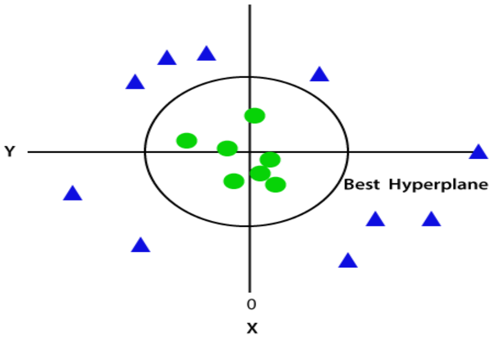


Fig. 11. Non-linear classification

Figure 11. shows classification of classes in a nonlinear environment.

IV. EXPERIMENTAL RESULTS

The system displays the image of a person with its true and the predicted name. When a person comes in front of the camera, images are captured and stored in the database. PCA and SVM algorithms are applied on this database for extracting Eigenfaces and classification respectively. The output of the system is shown below with a true name taken directly from the database and predicted name which is obtained by the algorithms.

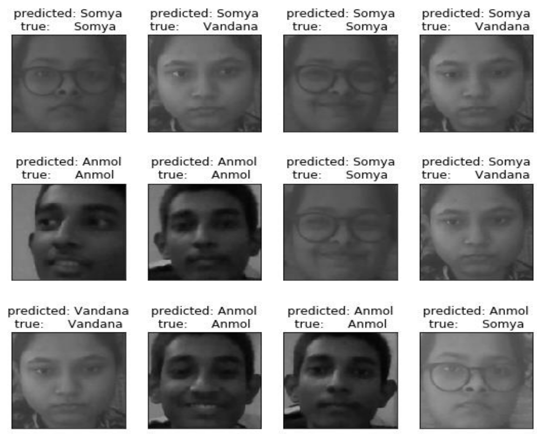


Fig. 12. Output of the system

The model accuracy is tested based on a different number of faces and different Eigenfaces.

A. *Different number of eigenfaces*

For this comparison different number of Eigenfaces are taken when number of faces is kept three.

TABLE 1

Number of Eigen faces	Accuracy obtained
75	99
150	97
200	67

Table 1 shows system accuracy with different number of Eigenfaces.

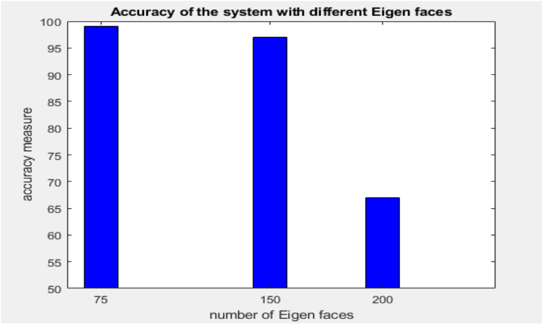


Fig. 13. Graph representing the accuracy with different Eigenfaces

Fig. 13 shows the system accuracy when Eigenfaces present are 75, 150 and 200 and the obtained accuracy are 99, 97 and 67 respectively. From the above Table, it is clear that the accuracy of the system decreases as the number of Eigenfaces is increased because the system may get confused between faces when multiple Eigenfaces are present.

B. *Different number of faces*

In this section, the accuracy is compared where a different number of faces are present. The Eigenfaces in this case are kept 75.

TABLE 2

Number of faces	Accuracy
2	100
3	99
4	100

The above table represents the accuracy of the system when a different number of faces are present i.e. when the system has a database of three persons, accuracy is 99% and with database of four persons, it is 100%.

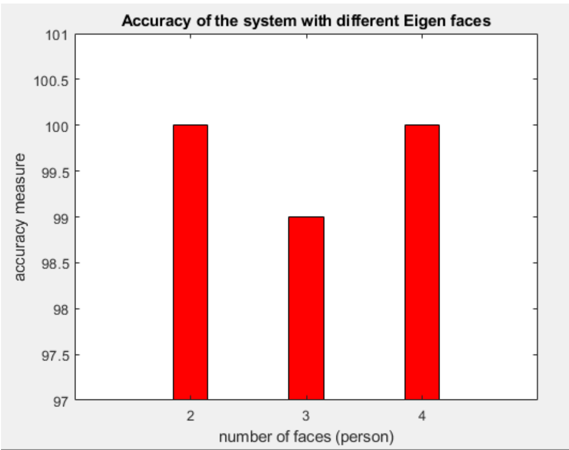


Fig. 14. Graph representing accuracy when a different number of person ids are present inside the system.

According to the table, accuracy with two faces is 100% as less Eigenfaces will be present for classification, hence better will be the performance of classifier. When three faces are present for classification, accuracy obtained is 99% and with four faces present inside the system, accuracy is 100%.

From the above two cases, it can be concluded that desired accuracy can be maintained by keeping the count of Eigenfaces low and have seen that the accuracy was decreasing when the count of Eigenfaces was increasing. Also, cases are shown when the number of faces increases keeping the number of Eigenfaces as constant. With three faces, Accuracy was less compared to when there were four faces. With these results, the count of Eigenfaces plays a vital in deciding the accuracy of the classification can be concluded. It has to be adjusted such that it is not too low when the classifier does not get enough faces for classification and too high when it may get confused.

V. CONCLUSION

This system uses Haar filters for face detection which can properly detect a face from an image. PCA algorithm gives Eigenfaces with unique feature for each face and works with SVM algorithm for classification between them. Eigenfaces, being the dominant feature of a person gives better results while comparing with the faces present in the dataset. SVM performs the classification with the Eigenfaces in a nonlinear environment and predict the name of a person based on classification results. Eigenfaces play a major role in deciding the classifier accuracy. It has to be adjusted such that it is not too low when the classifier does not get enough faces for classification and too high when it may get confused. The suggested system gives better accuracy for unique identification due to Eigenfaces and their classification with SVM classifier.

REFERENCES

[1] Classification Mechanism of Support Vector Machines Chen Junli Jiao Licheng Key Lab. for Radar Signal Processing, Xidian Univ., Xi'an 710071, China

[2] Article on Applying the Haar-cascade Algorithm for Detecting Safety Equipment in Safety Management Systems for Multiple Working Environments.

[3] Overfitting and Underfitting Analysis for Deep Learning Based End-to-end Communication Systems. School of Electronics and Information Technology, Sun Yat-sen University, Guangzhou, China.

[4] Bashar, A. (2019),"Survey On Evolving Deep Learning Neural Network Architectures", Journal of Artificial Intelligence, 1(02), 73-82.

[5] Jacob, I. J. (2019),"Capsule Network Based Biometric Recognition System". Journal of Artificial Intelligence, 1(02), 83-94.

[6] Evaluation of face recognition technique using PCA wavelets and SVM. Istanbul university, engineering faculty, computer engineering department, 34320, Avcilar, Istanbul, Turkey.

[7] Data Analysis Using Principal Component Analysis, 2014 International Conference on Medical Imaging, m-Health and Emerging Communication Systems (MedCom).

[8] Evaluation of Haar Cascade Classifiers Designed for Face Detection, World Academy of Science, Engineering and Technology International Journal of Computer, Electrical, Automation, Control and Information Engineering Vol:6, No:4, 2012

[9] Face Detection and Recognition using Support Vector Machine, International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 – 8958, Volume-8 Issue-4, April, 2019.

[10] Data classification using support vector machine, Journal of Theoretical and Applied Information Technology

[11] Face Recognition Using Principal Component Analysis Method, International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 1, Issue 9, November 2012

[12] Real Time Face Recognition System (RTFRS), 4th International symposium on digital forensics and security, Little Rock AR.

[13] W. Zhao, R. Chellappa, A. Krishnaswamy, Discriminant analysis of principalcomponents for face recognition, in: Third IEEE International Conference onAutomatic Face and Gesture Recognition, 1998, pp. 336–341.

[14] T. Mantoro, M. A. Ayu and Suhendi, "Multi-Faces Recognition Process Using Haar Cascades and Eigenface Methods," 2018 6th International Conference on Multimedia Computing and Systems (ICMCS), Rabat, 2018, pp. 1-5, doi: 10.1109/ICMCS.2018.8525935.

[15] A. Priadana and M. Habibi, "Face Detection using Haar Cascades to Filter Selfie Face Image on Instagram," 2019 International Conference of Artificial Intelligence and Information Technology (ICAIIT), Yogyakarta, Indonesia, 2019, pp. 6-9.

[16] A. Al-Anazi, I.D. Gates, A support vector machine algorithm to classify lithofacies and model permeability in heterogeneous reservoirs, Engineering Geology, Volume 114, Issues 3–4, 2010, Pages 267-277, ISSN 0013-7952.