PES UNIVERSITY

(Established under Karnataka Act No. 16 of 2013)

100-ft Ring Road, Bengaluru – 560 085, Karnataka, India

Report on

'MACHINE LEARNING BASED FACIAL DETECTION ALGORITHM FOR VOTER RECOGNITION SYSTEM'

Submitted by

ROUNAK RAJ(PES1201800418)

DEEKSHA H KULAL (PES1201800501)

SANDHYA BALAKRISHNA (PES1201801036)

SUNNY SINGH (PES1201801331)

Jan. - May 2021

Under the guidance of

Prof. Shruthi M.L.J

Assistant Professor

Department of ECE

PES University

Bengaluru -560085

CERTIFICATE

This is to certify that the Report entitled

'MACHINE LEARNING BASED FACIAL DETECTION ALGORITHM FOR VOTER RECOGNITION SYSTEM'

is a bonafide work carried out by

ROUNAK RAJ(PES1201800418)

DEEKSHA H KULAL (PES1201800501)

SANDHYA BALAKRISHNA (PES1201801036)

SUNNY SINGH (PES1201801331)

In partial fulfillment for the completion of 6 semester course work in the Program of Study B.Tech in Electronics and Communication Engineering under rules and regulations of PES University, Bengaluru during the period Jan – May. 2021. It is certified that all corrections/suggestions indicated for internal assessment have been incorporated in the report. The report has been approved as it satisfies the 6th semester academic requirements in respect of CAPSTONE(Phase 1) project work.

Signature with date & Seal
Internal Guide

Signature with date & Seal
Chairperson

Name/s of the student/s **ROUNAK RAJ(PES1201800418)**

DEEKSHA H KULAL (PES1201800501)

SANDHYA BALAKRISHNA (PES1201801036)

SUNNY SINGH (PES1201801331)

Name of the examiners:

DECLARATION

We, ROUNAK RAJ (PES1201800418), DEEKSHA H KULAL (PES1201800501), SANDHYA BALAKRISHNA (PES1201801036) and SUNNY SINGH (PES1201801331) hereby declare that the report entitled, '**MACHINE LEARNING BASED FACIAL DETECTION ALGORITHM FOR VOTER RECOGNITION SYSTEM***',* is an original work done by us under the guidance of Prof.Shruthi.M.L.J, Assistant Professor, Department of ECE, PES University, and is being submitted in partial fulfillment of the requirements for completion of 6th Semester CAPSTONE(Phase 1) project work in the Program of Study B.tech in Electronics and Communication Engineering.

PLACE: Bangalore

DATE: 21.05.2021

NAME AND SIGNATURE OF THE CANDIDATE

ABSTRACT

Elections are fundamental defining characteristics of any democracy that is being governed by the people expressing their choices or articulating opinions in the form of voting.

In this report, a new authentication technique in the voting system using facial recognition of the voter is proposed. In India, currently, there are two voting systems in practice. They are secret Ballot paper and Electronic Voting Machines (EVM), but both of the processes have some limitations or demerits. The voters need to go to distributed places like polling booths and stand in a very long queue to cast their vote, due to these reasons most of the people miss the chance of casting their votes. The people who are not eligible to vote can also cast their votes by unwanted means, which may cause various problems.

The current voting system is not secure and time-consuming as well. That's why in this project we propose a system or way for voting which is very effective or useful in the voting process. The user authentication process of the system is improved by adding face recognition using machine learning techniques in an application that will identify whether the particular user is an authenticated user or not. This project is used to maintain high-level biometric security.

The voter's image is captured and passed to a face detection algorithm which is used to detect his face from the image and save it as the matching point. If he has the right to vote then a voting form is presented to him digitally to cast his vote.

ACKNOWLEDGEMENT

First and foremost, we would like to express our highest appreciation to our supportive academic professor Prof.Shruthi M.L.J. Her supervision and support truly helped during the period of conducting the project. Her never-ending support and guidance deeply encouraged us to perform and work better. We dedicate our complete thankfulness to her support.We are also grateful to our examiners, who have been so kind to provide their sincere assistance and good cooperation during this period in correcting our mistakes. Their cooperation is much indeed appreciated. We would also like to thank our friends who have also guided us equally in all our proceedings. Overall, this project wouldn't have gotten its form without our professors' and peer support.

Thank you!

CONTENTS (page no)

LIST OF FIGURES AND TABLES

CHAPTER 1: INTRODUCTION

1.1 PROBLEM STATEMENT

To design a machine learning based facial detection system for voting application to improve the security and efficiency

1.2 AIM

Voter registration and voting remains one of the most complex and contested parts of the electoral process. In a country like India where there is no trustworthy population census and no reliable identification documents, voter registration is even more complicated. Existing registers are often of poor quality, thus opening up avenues for manipulation and putting pressure on electoral management bodies to establish more reliable registration systems. In such a situation, it is often assumed that biometric technology can provide the required solutions.

1.3 OBJECTIVE

Currently in India, there are two types of voting systems in practice. They are secret ballot paper and Electronic Voting Machines (EVM), but both of the processes have some limitations or demerits. In India, online voting has not yet been implemented. The current voting system is not secure and time consuming as well.

The facial detection algorithm we have implemented in this is very much useful in identifying fraud voters, so we can avoid bogus votes during election commission. As data is stored in a centralized repository, data is accessible at any time as well as backup of the data is possible. Smart voting system provides updated results at each and every minute and it also is cost efficient as it is just one time investment for the government. The security level of our system is also highly improved compared to existing systems as it uses less manpower that cannot be fully trusted with.

1.4 MOTIVATION

With the occurrence of booth capturing, ballot rigging, voter impersonation and bogus voting during elections, this system aims to avoid such malpractices.The process is completely digitized. This increases the efficiency by reducing the manual intervention, thus reducing the possibility of above mentioned occurrences. The person need not carry any identification card to the polling booth. Thus no need to issue voters slip to each and every voter.

1.5 OVERVIEW OF THE REPORT

All the papers read as a part of our literature survey are listed in chapter 2 of this report. The concepts we came across in those papers, their advantages and disadvantages with respect to our project have also been discussed in the same.

Chapter 3 contains the working principle and the procedure we've followed in this project. Feature extraction algorithms, Eigenface approach and the Histogram of Oriented Gradients have been discussed along with the Viola Jones algorithm for face detection. Comparison of different classifiers (Linear discriminant, Support Vector Machine and Ensemble Learning) and their results are also shown in this section.

Chapter 4 discusses the general block diagram of how the entire process is going to be carried out, followed by Chapter 5, where the software required to do so is explained.

Results along with the conclusion of the progress so far along with the method finalized has been discussed in the last chapter, Chapter 6 of this report.

CHAPTER 2:

LITERATURE SURVEY

Face detection is a computer technology that determines the location and size of a human face in an arbitrary (digital) image. The facial features are detected and any other objects like trees, buildings and bodies etc are ignored from the digital image. It can be regarded as a specific case of object-class detection, where the task is finding the location and sizes of all objects in an image that belong to a given class. Face detection can be regarded as a more general case of face localization. In face localization, the task is to find the locations and sizes of a known number of faces (usually one). Basically there are two types of approaches to detect facial parts in the given image i.e. feature base and image based approach. Feature based approach tries to extract features of the image and match it against the knowledge of the face features. The image base approach tries to get the best match between training and testing images.

Detecting a face is a computer technology which lets us know the locations and sizes of human faces. This helps in getting facial features and avoiding other objects and things. In the present situation, human face perception is the biggest research area. It is basically about detecting a human face through some trained features. Here face detection is a preliminary step for many other applications such as face recognition, video surveillance etc.

Paper 1

Deep Face Recognition for Biometric Authentication

Link - https://ieeexplore.ieee.org/document/8940725

Year of publication: 2019

This paper[1] was focused on Viola Jones method for face detection.The four main components of Viola Jones algorithm include extraction of Haar-like features, computing integral images, speeding up feature extraction using Adaboost and cascading the features selected by Adaboost. The facial areas detected by Viola Jones algorithm are cropped and used for further processing.

The average detection rate is 97.41%.A better recognition rate was obtained by maximizing the number of images used as the face matching reference and governing the conditions at which the facial image retrieval image and testing face image had the same conditions.Compared with other studies where the results obtained tolerance angle for face recognition is 0 - 5 degrees, whereas in this research angle tolerance for face recognition reaches 70 degrees with improvement ratio 1:14.

ADVANTAGES

- Robust than facial variations as compared to holistic methods.
- Easier and cheaper because done in a cascade manner.
- Costly operations are applied only to the interest point of location.

DISADVANTAGES

- Low detection rates. Difficult to detect faces in complex backgrounds or in presence

of multiple faces.
- Sensitive to Illumination,face sizes,poses and expressions.
- Convolution cost is high. It has a high dimensional feature vector.

Paper 2
Appearance based Facial Detection for Recognition

Paper 2 - https://ieeexplore.ieee.org/document/6262545

Year of publication: 2012

This paper[2] focuses on the YCbCr model.The Y component represents the intensity of the light. Cb and Cr components indicate the intensities of the blue and red components relative to the green component.The eye is more sensitive to light intensity changes and less sensitive to hue changes. Detecting and locating the eyes-using eye map,symmetry map and image intensity.

Detecting and locating mouth.

Mapping of the mouth is done in regions that have a strong red component and small amount of blue component compared to other facial regions.

ADVANTAGE

- Helps increase the rate of facial recognition

DISADVANTAGE

- Ellipsoidal vignetting provides good results only to low resolution images.
- Affine transformation had chances of an error in detecting the eyes

Paper 3
LBP-Ferns-Based Feature Extraction for Robust Facial Recognition

Paper 3 - https://ieeexplore.ieee.org/document/7838098
Year of publication: 2016

This paper[3] focuses on DI and DR algorithm.The proposed feature extraction algorithm consists of dimension increasing (DI) and dimension decreasing (DR) parts.LBP is a very popular method and widely used as feature extraction method in face recognition systems.Not all DR techniques can be used.This paper uses OLDA method.The model struggles with dataset containing images with great variation in illumination.facial changes.

Needs to optimise pre-processing techniques so as to deal with such datasets

ADVANTAGES

- The computational complexity and recognition rate of the proposed method is suitable for application in consumer electronics without high performance processors.

- The method is quite robust in terms of expression and facial variation.

DISADVANTAGES

- Highly sensitive to noises in the image.
- Time delay is on a higher side.
- The proposed method struggles with a dataset having images of great facial variation.

Paper 4
Hybrid Approach to Face Recognition System using PCA and LDA

Paper 4 - https://ieeexplore.ieee.org/document/9114426
Year of publication: 2020

The paper[4] proposes a hybrid approach to face recognition with PCA + LDA algorithm in the border control environment.

PCA uses Eigenfaces and Eigenvectors to extract the most relevant features in the face.The covariance matrix of the training data is calculated. Weights are found out after selecting the set of relevant eigenfaces. The input image is projected on the mean image and the classification is done based on the distance measure methods.Illumination has a huge impact in accuracy. We use LDA which overcomes this limitation using linear discriminant criterion.

With further modifications in the techniques such as using MPCA and LPP increases the recognition rate .Viola-Jones method is used to detect the face, and PCA and LDA algorithms were used in the recognition phase in a hybrid way.PCA and LDA used for feature extraction and dimension reduction respectively.LDA is an appearance-based method which can increase the recognition rate.After PCA and LDA algorithm generates the feature space with respect to each other,then the nearest mean is applied.

The proposed system focuses on performing PCA before LDA to avoid the overfitting with the hybrid approach.

ADVANTAGES

- This hybrid approach increases the robustness of the face recognition system in real-time application
- Use of LDA after PCA helps in overcoming PCA limitation
- The method took less time for face recognition
- The proposed method helped in achieving high accuracy

DISADVANTAGES

- Drawback of PCA is high computation and low discriminating power.
- The demerit of LDA is that it is inefficient in extracting features.
- PCA has discriminative information loss.

Paper 5
Fisher Vectors Meet Neural Networks: A Hybrid Classification Architecture

Paper 5 - https://ieeexplore.ieee.org/document/7298998
Year of publication: 2015

This paper [5] proposed a hybrid architecture for image classification that combines the benefits of FV and CNN pipelines. A first set of unsupervised layers involves the extraction and dimensionality reduction of FVs while the subsequent supervised layers are fully connected and trained with back propagation. System does not require training with GPUs or large CPU clusters significantly improved over previous FV systems without incurring the high cost associated with CNNs.Also derive competitive mid-level features from the proposed architecture that are readily applicable to other class sets .FVs contain much fine grained information that is discarded during PCA reduction of high dimensionality FVs.The unsupervised layers are not fine tuned.

ADVANTAGES

- System does not require training with GPUs or large CPU clusters
- significantly improved over previous FV systems without incurring the high cost associated with CNNs
- Also derive competitive mid-level features from the proposed architecture that are readily applicable to other class sets

DISADVANTAGES

- FVs contain much fine grained information that is discarded during PCA reduction of high dimensionality FVs
- The unsupervised layers are not fine tuned

Paper 6
Evaluation of relevance vector machine classifier for a real-time face recognition system

Paper 6 - https://ieeexplore.ieee.org/abstract/document/8307832
Year of publication: 2017

This paper[6] proposed a real-time face recognition system using HOG features and

Relevance Vector Machine classifier is designed in this paper and experiments were carried out using datasets from
standard AT&T database as well as real-time input.Performance evaluation of proposed system is carried out using different architectures for RVM classifier and it is observed that Half-Against-Half gave best recognition with minimum time required for recognition.

ADVANTAGES

- RVM that is employed for the proposed system out performs SVM and other algorithms in terms of sparseness, classification time and accuracy.

DISADVANTAGES

- The classification time will increase with increase in the number of classes

OUTCOME OF THE SURVEY:

For this concept of real time face recognition process,We came across Viola Jones method for face detection which had certain advantages such as its more powerful algorithm though there are better models today. It sets the foundation for it in the field of facial detection.

We propose to use Histogram of Oriented Gradients(HOG) descriptors because we need a robust feature set to discriminate and find faces under difficult illumination backgrounds, wide range of poses, etc, by using feature sets that overcome the existing ones for face detection.

To compare the extent of the feature recognition process, we decided to use a slightly different approach by the EigenFaces algorithm due to its advantages.

The use of supervised machine learning techniques increases the performance rate of facial recognition where the Support Vector Machine(SVM) classifier is more efficient both in the learning and classification task.

CHAPTER 3:

METHODOLOGY

The working principle and the procedure followed is mentioned briefly below.

3.1 WORKING PRINCIPLE

- The database consisting of the eligible voters are pre-processed.
- The neural net performs feature extraction and feature classification.
- The training process is carried out.
- On the other hand, the camera captures the picture of the voter and then it is pre-processed and provided for testing.
- Then it is checked if the recognized voter is eligible to vote.

Eligible voters will be allowed to vote.

3.2 PROCEDURE FOLLOWED

Face Recognition consists of five main phases:

- Database Collection - The individual recognition for the face appears in the input database collected.
- Face detection - Detection of the region consisting of only the face.
- Feature extraction - The general efficiency of the facial recognition system is essentially reliant on the utilized methods of facial features extraction.
- Classification - Utilizes tools such as the learning techniques and automatically recognizes faces of learning techniques.

The below given figure 3.1 depicts the flow of methodology we intend to follow.

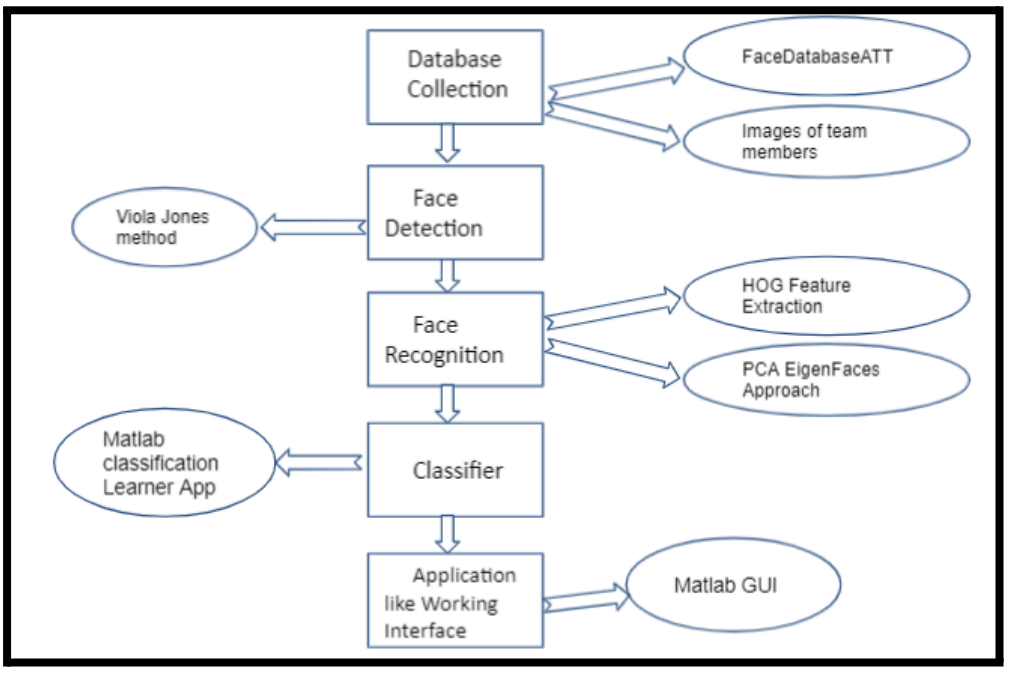


Figure 3.1 Flowchart of Methodology

3.3 DATABASE COLLECTION

Images of 40 people with 10 images for each person are taken from FacedatabaseATT.

Images of the members of the team are also collected and placed in the same folder,Web camera with the snapshots taken with a rate of 1 snapshot per second. 80% of data will be used for training and the rest 20% for testing.

3.4. FACE DETECTION BY VIOLA JONES METHOD

It was developed in 2001 by Paul Viola and Michael Jones, the Viola-Jones algorithm is an object-recognition framework that allows the detection of image features in real-time. Despite being an outdated framework, Viola-Jones is quite powerful and its application has proven to be exceptionally notable in real-time face detection.

It has two phases of training and detection. We discuss detection first.

Viola-Jones was designed for frontal faces, so it is able to detect the frontal the best rather than faces looking sideways, upwards or downwards.

1) Conversion to Grayscale

Before detecting a face, the image is converted into grayscale, since it is easier to work with and there's less data to process. The Viola-Jones algorithm first detects the face on the grayscale image and then finds the location on the colored image.

2) Outline A Bounding Box

Viola-Jones outlines a box on the image and searches for a face within the box. It is essentially searching for these haar-like features, which will be explained later. The box moves a step to the right after going through every tile in the picture. We can change the box size and step size according to our needs. With smaller steps, a number of boxes detect face-like features (Haar-like features) and the data of all of those boxes put together, helps the algorithm determine where the face is.

3) Haar-like Features

Haar-like features are named after Alfred Haar, a Hungarian mathematician in the 19th century who developed the concept of Haar wavelets. The features below show a box with a light side and a dark side, which is how the machine determines what the feature is.There are 3 types of Haar-like features that Viola and Jones identified are edge features, line features and four-sided features.

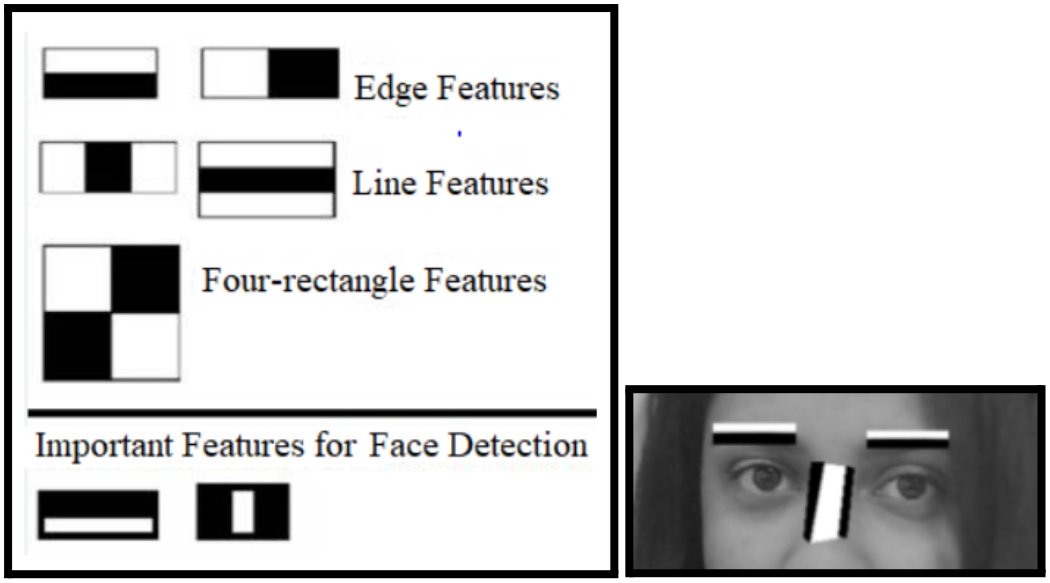


Figure 3.2 Haar Features in Viola Jones

Sometimes one side will be lighter than the other, as in the edge of an eyebrow. Sometimes the middle portion may be shinier than the surrounding boxes, which can be interpreted as a nose as shown in the figure 3.2.

These features help the machine understand what the image is. If we imagine what the edge of a table would look like on a black & white image. One side will be lighter than the other, creating that edge-like black & white feature. In the two important features for Face Detection, the horizontal and the vertical features describe what eyebrows and the nose, respectively, look like to the machine.

Additionally, when the images are inspected, each feature has a value of its own. It's quite easy to calculate by subtracting white area from the black area.

If we imagine our haar-like feature was converted into a grid. Then each square represents a pixel. Suppose we choose a 8 x 4 grid as in figure 3.3, but in reality, there would be many more pixels and thus a much larger grid for a certain feature. The numbers in the boxes represent the darkness of the features. The higher it is, the darker the pixel. Thus we can see the numbers are higher on the right side than on the left side. Now if we add up the numbers on the two left-sided (white) columns, and subtract it from the sum of the right-sided columns, we will get the value of the particular feature.

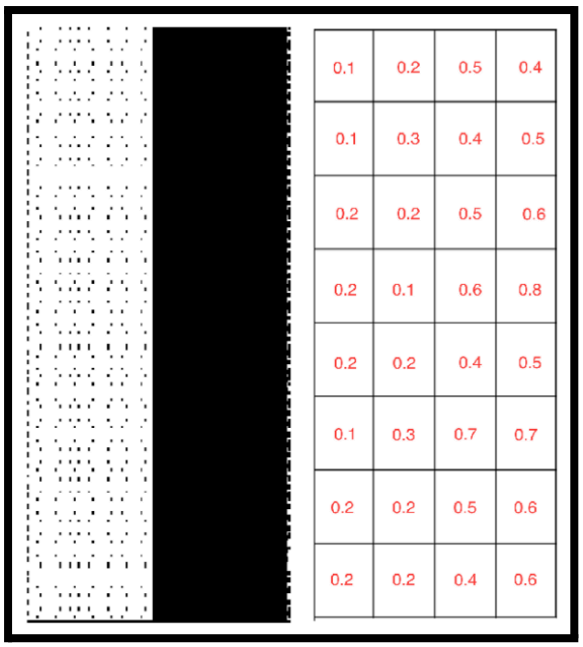


Figure 3.3 Haar Feature with its values

So in this case, the value of our feature is → (0.5 + 0.4 + 0.5 +0.6 + 0.4 + 0.7 + 0.5 + 0.4 +0.4 + 0.5 + 0.6 + 0.8 + 0.5 + 0.7 + 0.6 + 0.6) -(0.1 + 0.1 + 0.2 + 0. 2+ 0.2 + 0.1 + 0.2 + 0.2 +0.2 + 0.3 + 0.2 + 0.1 + 0.2 + 0.3 + 0.2 + 0.2) = 8.7 - 3 = 5.7

4. Integral Image

So we calculated the value of a feature. In reality, these calculations can be very intensive since the number of pixels would be much greater within a large feature.The integral image plays its part in allowing us to perform these intensive calculations quickly so we can understand whether a feature of a number of features fit the criteria. To calculate the value of a single box in the integral image, we take the sum of all the boxes to its left.

So why do we use the integral image?

Because Haar-like features are actually rectangular, and the integral image process allows us to find a feature within an image very easily as we already know the sum value of a particular square and to find the difference between two rectangles in the regular image, we just need to subtract two squares in the integral image. All we have to do is look at the 4 corners of our feature in figure 3.4, and add the purples, subtract the greens (168–114 + 79–110 = 23). So even if we had a 1000 x 1000 pixels in our grid, the integral image method makes the calculations much less intensive and can save a lot of time for any facial detection model.

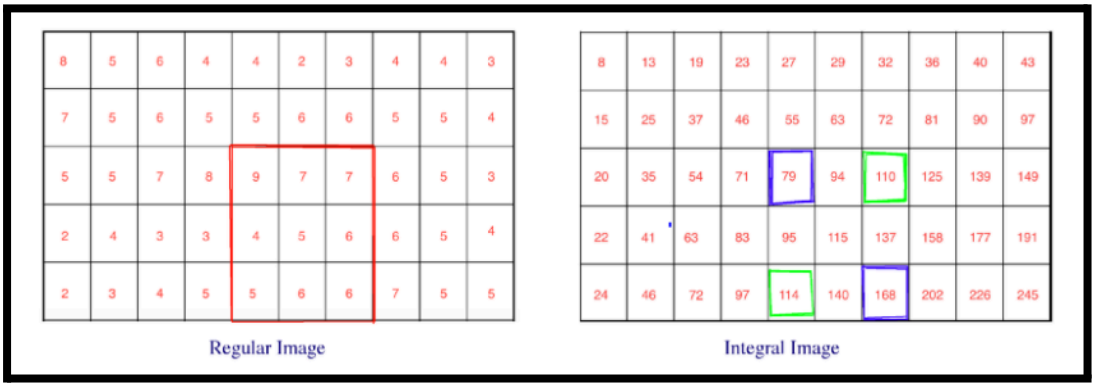


Figure 3.4 Integral Image Calculation

5. Training Classifiers

We're training the machine to identify these features. We're feeding it information, and subsequently training it to learn from the information to predict. So ultimately, the algorithm is setting a minimum threshold to determine whether something can be classified as a feature or not.

The algorithm shrinks the image to 24 x 24 and looks for the trained features within the image. It needs a lot of facial image data to be able to see features in the different and varying forms. That's why we need to supply lots of facial image data to the algorithm so it can be trained. We would also need to supply the algorithm with non-facial images so it can differentiate between the two classes.

6. Adaptive Boosting (AdaBoost)

The algorithm learns from the images we supply it and is able to determine the false positives and true negatives in the data, allowing it to be more accurate. We would get a highly accurate model once we have looked at all possible positions and combinations of those

features. Training can be super extensive because of all the different possibilities and combinations we would have to check for every single frame or image.

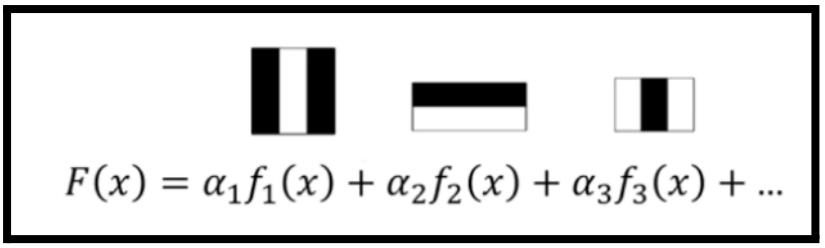


Figure 3.5 Adaboost Classifier

Let's say we have an equation for our features that determines the success rate with f1, f2 and f3 as the features and a1, a2, a3 as the respective weights of the features as in figure 3.5. Each of the features is known as a weak classifier. The left side of the equation F(x) is called a strong classifier. Since one weak classifier may not be as good, we get a strong classifier when we have a combination of two or three weak classifiers. As we keep adding, it gets stronger and stronger. This is called an ensemble. We want to make sure that we have the most important features in front, but the question is how do we find the most important or the 'best' features? That's where Adaptive Boosting comes into play.

In the next step, adaptive boostiNg uses another feature, the one to best complement our current strongest feature. So it doesn't look for the second-best feature, but one that complements the current best feature. So it increases the importance of the images that it got wrong as false negatives, and finds the next best feature that would fit these images, in a way, increasing the weight of these images on the overall algorithm. So, as new features are added, we would come down to one image at the end that would be given a higher weight. Once the algorithm is optimized and is able to calculate all positives and negatives correctly, we move on to the next step: cascading.

7. Cascading

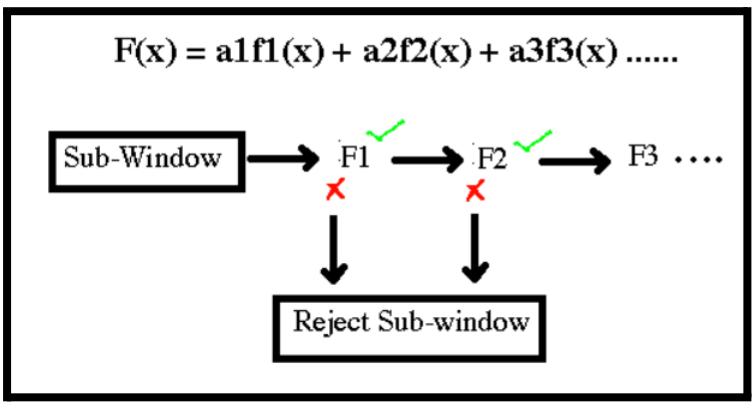


Figure 3.6 Cascading

Cascading is another sort of hack to boost the speed and accuracy of our model. So we start by taking a subwindow and within this subwindow, we take our most important or best feature and see if it is present in the image within the subwindow. If it is not in the subwindow, then we don't even look at the subwindow, we just discard it. Then if it is present, we look at the second feature in the subwindow. If it isn't present, then we reject the subwindow. We go on for the number of features, and reject the subwindows without the feature. This is shown in figure 3.6.Evaluations may take split seconds but since we have to do it for each feature, it could take a lot of time. Cascading speeds up this process a lot, and the machine is able to deliver results much faster.

3.5 FEATURE EXTRACTION

Feature Extraction or a Feature descriptor is a simplified representation of the image that contains only the most important information about the image.

Here we are introduced to a popular feature extraction technique for images known as the Histogram of Oriented Gradients or HOG as it is commonly known. We will understand what is the HOG feature descriptor, how it works (the complete math behind the algorithm), and finally, implement it.

3.5.1 HISTOGRAM OF GRADIENTS

HOG, or Histogram of Oriented Gradients, is a feature descriptor that is often used to extract features from image data. It is widely used in computer vision tasks for object detection.

Some important aspects of HOG that makes it different from other feature descriptors are:

- The HOG descriptor focuses on the structure or the shape of an object. HOG not only identifies edge features but is able to provide the edge direction as well. This is done by extracting the gradient and orientation of the edges.
- Additionally, these orientations are calculated in 'localized' portions. This means that the complete image is broken down into smaller regions and for each region, the gradients and orientation are calculated.
- Finally the HOG would generate a Histogram for each of these regions separately. The histograms are created using the gradients and orientations of the pixel values, hence the name 'Histogram of Oriented Gradients'

3.5.2 STEPS TO CALCULATE HISTOGRAM OF GRADIENTS

1. Preprocess the Data

Preprocessing data is a crucial step in any machine learning project and that's no different when working with images.

We need to preprocess the image and bring down the width to height ratio to 1:2. The image size should preferably be 64 x 128. This is because we will be dividing the image into 8*8 and 16*16 patches to extract the features.

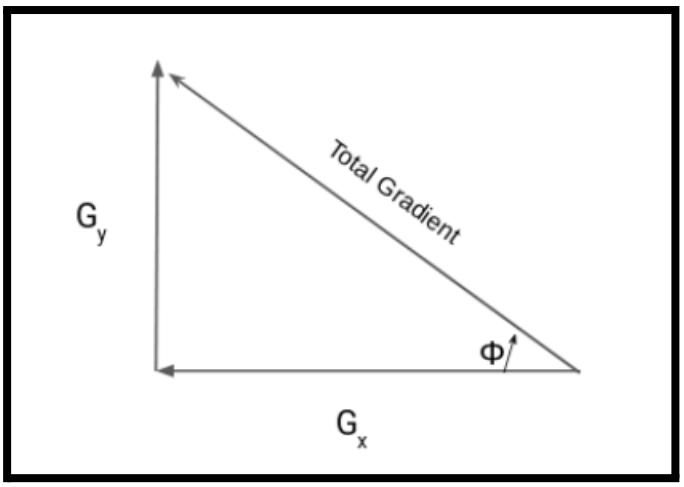
2. Calculating Gradients

Gradients are the small change in the x and y directions.To determine the gradient in the x-direction, we need to subtract the value on the left from the pixel value on the right. Similarly, to calculate the gradient in the y-direction, we will subtract the pixel value below from the pixel value above the selected pixel.

3. Calculating Magnitude and Orientation

We will now determine the magnitude and direction for each pixel value. For this step, we will be using the Pythagoras theorem as shown in figure 3.7.

Total Gradient Magnitude = $\sqrt{G_x^2 + G_y^2}$ Orientation = $tan(\phi) = \frac{G_y}{G_x}$



3.7 Gradient Calculation

We need to generate the histogram using these gradients and orientations.

4. Different Methods to Create Histograms using Gradients and Orientation

A histogram is a plot that shows the frequency distribution of a set of continuous data. We have the variable (in the form of bins) on the x-axis and the frequency on the y-axis. Here, we are going to take the angle or orientation on the x-axis and the frequency on the y-axis.

Method 1:

We will take each pixel value, find the orientation of the pixel and update the frequency table.The same process is repeated for all the pixel values, and we end up with a frequency table that denotes angles and the occurrence of these angles in the image. This frequency table can be used to generate a histogram with angle values on the x-axis and the frequency on the y-axis.This method is shown in figure 3.8.

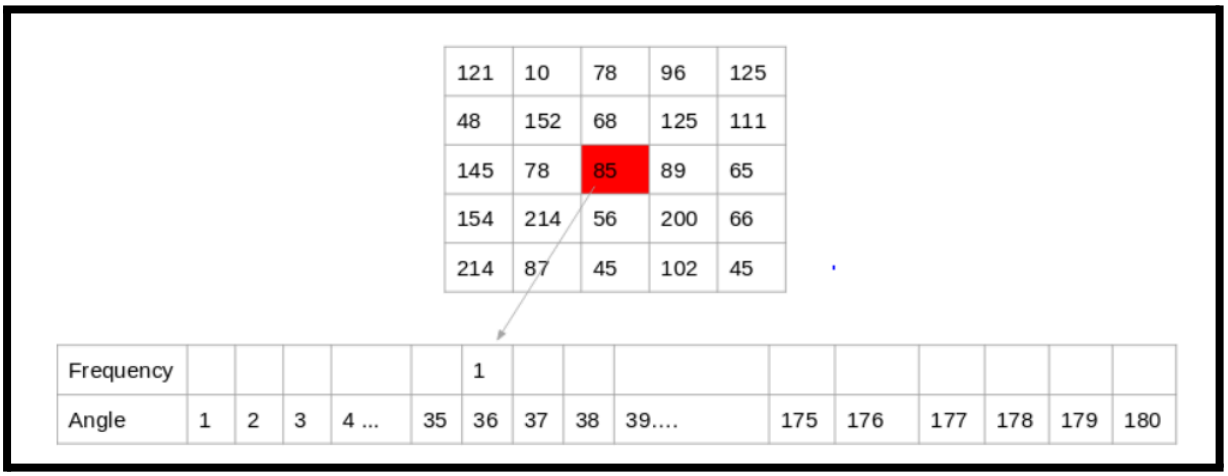


Figure 3.8 Histogram Calculation in Method 1

Method 2:

This method is similar to the previous method, except that here we have a bin size of 20. So, the number of buckets we would get here is 9. Again, for each pixel, we will check the orientation, and store the frequency of the orientation values in the form of a 9 x 1 matrix. This method is shown in figure 3.8.

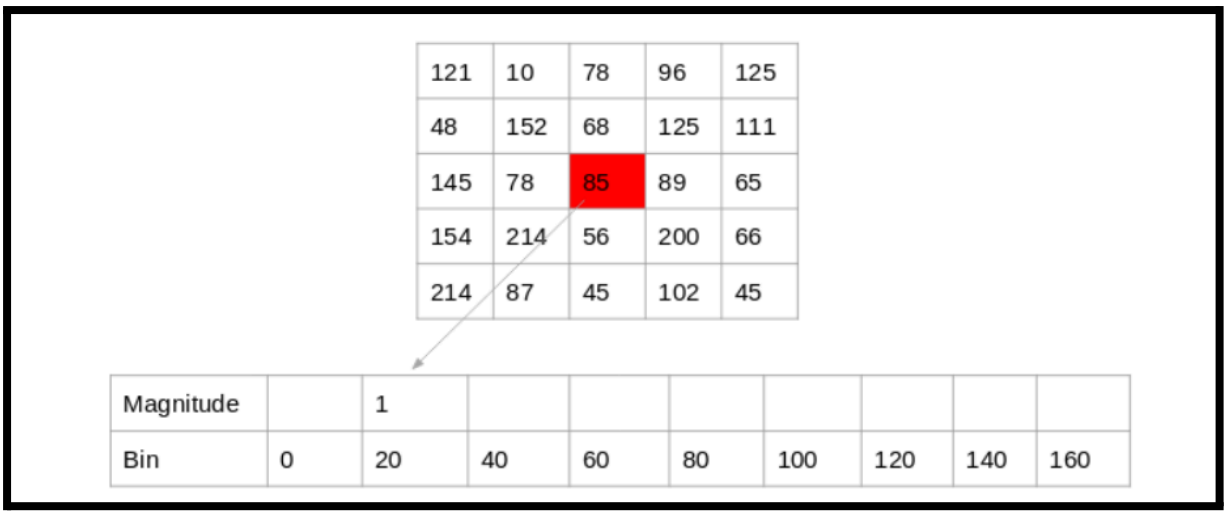


Figure 3.9 Histogram Calculation in Method 2

Method 3:

Instead of using the frequency, we can use the gradient magnitude to fill the values in the matrix. This method is shown in figure 3.9.

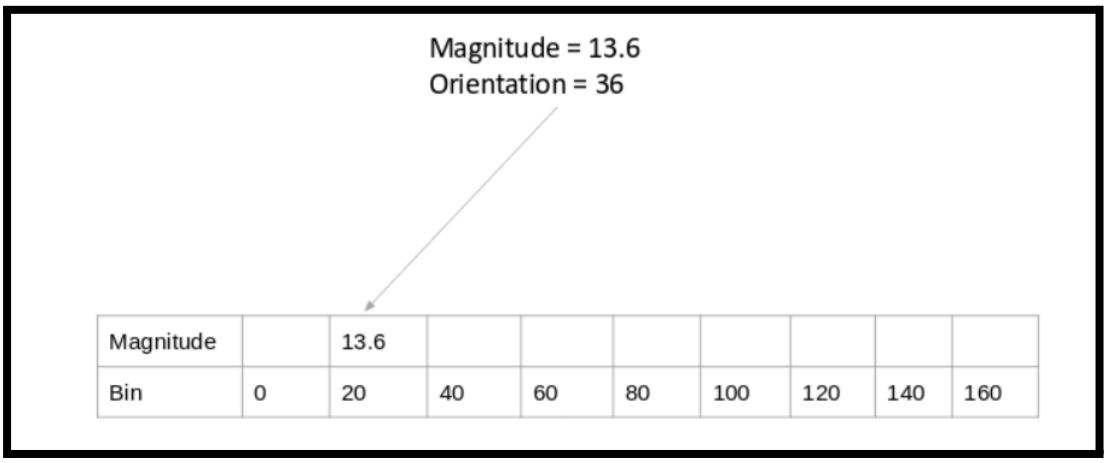


Figure 3.10 Histogram Calculation in Method 3

Method 4:

Small modification to the above method. Here, we will add the contribution of a pixel's gradient to the bins on either side of the pixel gradient. We must remember that the higher contribution should be to the bin value which is closer to the orientation.This method is shown in figure 3.10.

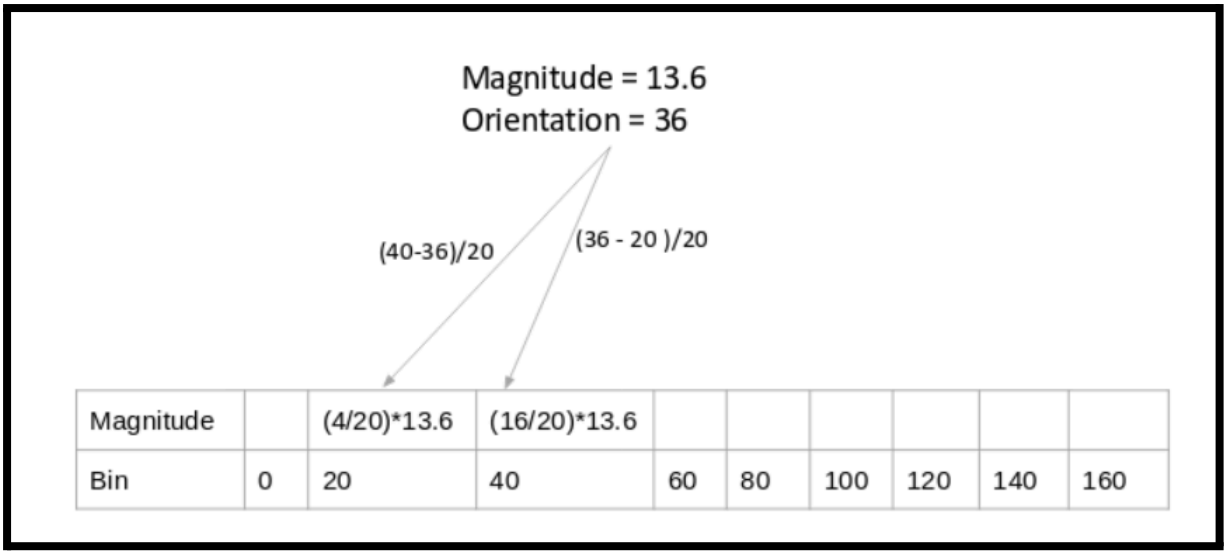


Figure 3.11 Histogram Calculation in Method 4

5. Calculate Histogram of Gradients in 8×8 cells (9×1)

The image is divided into 8×8 cells, and the histogram of oriented gradients is computed for each cell. By doing so, we get the features (or histogram) for the smaller patches which in turn represent the whole image. We can certainly change this value here from 8 x 8 to 16 x 16 or 32 x 32.

If we divide the image into 8×8 cells and generate the histograms, we will get a 9 x 1 matrix for each cell.

6. Normalize gradients in 16×16 cell (36×1)

Although we already have the HOG features created for the 8×8 cells of the image, the gradients of the image are sensitive to the overall lighting. This means that for a particular picture, some portion of the image would be very bright as compared to the other portions.

We cannot completely eliminate this from the image. But we can reduce this lighting variation by normalizing the gradients by taking 16×16 blocks.

Here is an example that can explain how 16×16 blocks are created:

Here, we will be combining four 8×8 cells to create a 16×16 block. And we already know that each 8×8 cell has a 9×1 matrix for a histogram. So, we would have four 9×1 matrices or a single 36×1 matrix. To normalize this matrix, we will divide each of these values by the square root of the sum of squares of the values. Mathematically, for a given vector V:

V = $[a_1, a_2, a_3 \ldots a_{36}]$

We calculate the root of the sum of squares:

$$k = \sqrt{a_1^2 + a_2^2 + \ldots + a_{36}^2}$$

And divide all the values in the vector V with this value k:

Normalised Vector = $[\frac{a_1}{k}, \frac{a_2}{k}, \frac{a_3}{k}, \ldots \frac{a_{36}}{k}]$

The resultant would be a normalized vector of size 36×1.

3.5.3 EIGENFACES APPROACH

PCA (Principal Component Analysis) is a dimensionality reduction technique that was proposed by Pearson in 1901. It uses Eigenvalues and EigenVectors to reduce dimensionality and project a training sample/data on small feature space.

ALGORITHM

Training Algorithm

1) Consider a set of M images of dimension $N \times N$.
2) Convert images of dimension $N \times N$into size $N^2 \times 1$.
3) Now we calculate the average of all these face vectors and subtract it from each vector

$$\psi = \frac{1}{m} \sum_{i=1}^{m} x_i$$

$$a_i = x_i - \psi$$

4) Now we take all face vectors so that we get a matrix of size of $N^2 \times M$.
5) Now, we find Covariance matrix by multiplying A with A^T. A has dimensions $N^2 \times M$, thusA^T has dimensions $M \times N^2$. When we multiplied this gives us matrix of $N^2 \times N^2$, which gives us N^2 eigenvectors of N^2 size which is not computationally

efficient to calculate. So we calculate our covariance matrix by multiplying A^T and A. This gives us $M \times M$ matrix which has M (assuming M << N2) eigenvectors of size M.

$$C = A^T A$$

6) We calculate eigenvalues and eigenvectors of the above covariance matrix using the formula below.

$$A^T A v_i = \lambda_i v_i$$

$$AA^T A v_i = \lambda_i A v_i$$

$$C' u_i = \lambda u_i \text{ where } u_i = A v_i \text{ and } C' = AA^T$$

From the above statement It can be concluded that C' and C have same eigenvalues and their eigenvectors are related by the equation $u_i = Av_i$. Thus, the M eigenvalues (and eigenvectors) of covariance matrix gives the M largest eigenvalues(and eigenvectors) of C'.

7) We select the K eigenvectors of C' corresponding to the K largest eigenvalues (where K < M). These eigenvectors have size N^2.
8) In this step we used the eigenvectors that we got in the previous step. We take the normalized training faces (face – average face) x_i and represent each face vector in the linear combination of the best **K** eigenvectors.

$$x_i - \psi = \sum_{j=1}^{K} w_j u_j$$

These u_j are called **EigenFaces**.

9) We take the coefficient of eigenfaces and represent the training faces in the form of a vector of those coefficients.

$$x_i = \left[w_1^i \; w_2^i \; w_3^i \ldots\ldots w_k^i \right]$$

Testing Algorithm

- Given an unknown face *y*, we need to first preprocess the face to make it centered in the image and have the same dimensions as the training face

- 2Now, we subtract the face from the average face Ψ.
  $\phi = y - \Psi$
- We project the normalized vector into eigenspace to obtain the linear combination of eigenfaces.

$$\phi = \sum_{I=1}^{k} w_i u_i$$

we generate the vector of the coefficient such that

$$\Omega^T = \left[w_1 w_2 w_3 .... w_k\right]^T$$

- We take the vector generated in the above step and subtract it from the training image to get the minimum distance between the training vectors and testing vectors
  $e_r = min\left|\left|\Omega - \Omega_l\right|\right|$
- If this e_r is below tolerance level *Tr*, then it is recognised with *l* face from training image else the face is not matched from any faces in the training set.

3.6 CLASSIFIER

3.6.1 MATLAB CLASSIFICATION LEARNER APPLICATION

The Classification Learner app trains models to classify data. Using this app, we can explore supervised machine learning using various classifiers. We can explore our data, select features, specify validation schemes, train models, and assess results. We can perform automated training to search for the best classification model type, including decision trees, discriminant analysis, support vector machines, logistic regression, nearest neighbors, naive Bayes, ensemble, and neural network classification.

We can perform supervised machine learning by supplying a known set of input data (observations or examples) and known responses to the data (e.g., labels or classes). We use the data to train a model that generates predictions for the response to new data. To use the model with new data, or to learn about programmatic classification, we can export the model to the workspace or generate MATLAB code to recreate the trained model.

We can use Classification Learner to train models of these classifiers: decision trees, discriminant analysis, support vector machines, logistic regression, nearest neighbors, naive Bayes, ensembles, and neural networks. In addition to training models, we can explore our data, select features, specify validation schemes, and evaluate results. We can export a model to the workspace to use the model with new data or generate MATLAB code to learn about programmatic classification.

Training a model in Classification Learner consists of two parts:

Validated Model: Train a model with a validation scheme. By default, the app protects against overfitting by applying cross-validation. Alternatively, we can choose holdout validation. The validated model is visible in the app.

Full Model: We train a model on full data without validation. The app trains this model simultaneously with the validated model. However, the model trained on full data is not visible in the app. When we choose a classifier to export to the workspace, Classification Learner exports the full model.

The app displays the results of the validated model. Diagnostic measures, such as model accuracy, and plots, such as a scatter plot or the confusion matrix chart, reflect the validated model results. We can automatically train a selection of all classifiers, compare validation results, and choose the best model that works for our classification problem. When we choose a model to export to the workspace, Classification Learner exports the full model. Because Classification Learner creates a model object of the full model during training, we experience no lag time when we export the model. We can use the exported model to make predictions on new data.

3.6.2 COMPARISON OF THE RESULTS OBTAINED FROM THE MATLAB CLASSIFIER APPLICATION FOR DIFFERENT MODELS

The MATLAB classification learner app trains the input features with many models such as SVM,trees,ensemble,neural networks,etc. It gives the accuracy for each of the models so that we could choose the best one. Here is the accuracy of the top three models with highest accuracy in table 3.11 and classification error rate in table 3.12.

APPROACH/MODEL	SVM	LINEAR DISCRIMINANT	ENSEMBLE LEARNING
HOG	95.3% (Quadratic and Cubic)	98.1%	93.4%
EIGENFACE	94.7%	97.8%	97.2%

Table 3.12 - Table showing accuracy of different models

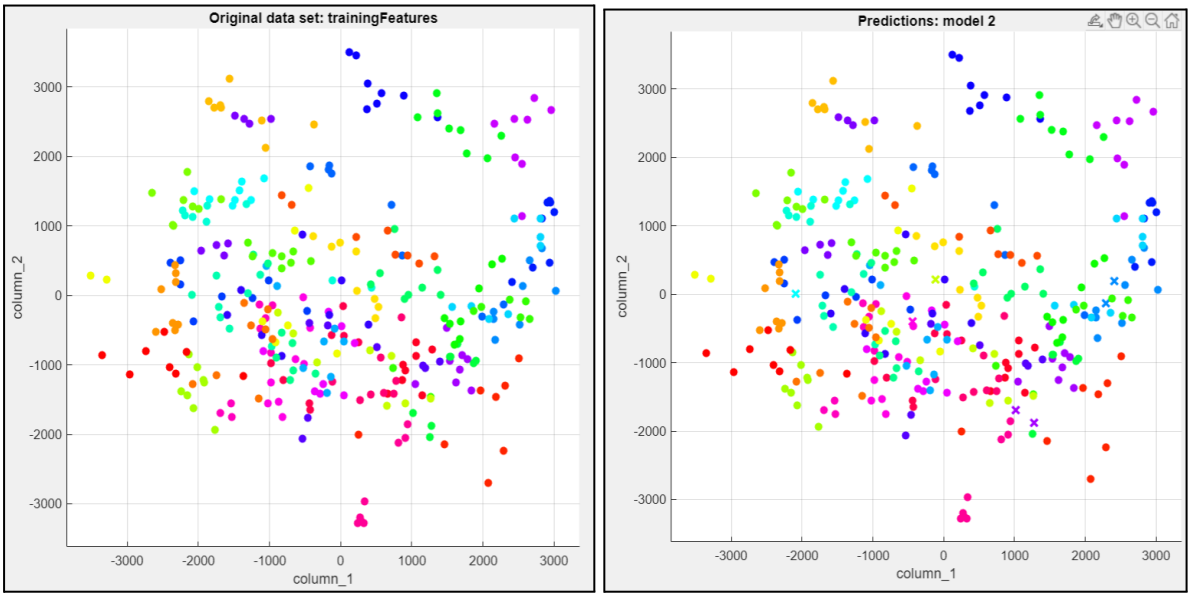
APPROACH/MODEL	SVM	LINEAR DISCRIMINANT	ENSEMBLE LEARNING
HOG	0.033%	0.014%	0.05%
EIGENFACE	0.035%	0.014%	0.019%

Table 3.13 - Table showing classification error rate of different models

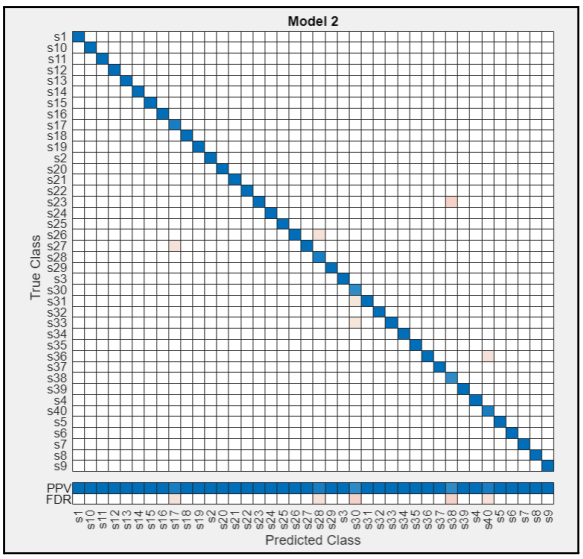
3.6.3 EIGENFACE APPROACH

3.6.3.1 Linear Discriminant Analysis

It is a generalization of Fisher's linear discriminant, a method used in Statistics, pattern recognition and machine learning to find a linear combination of features that characterizes or separates two or more classes of objects or events. Figure 3.13 shows scatter plots and confusion matrix for Linear Discriminant model in EigenFace Approach



Before Training **After Training**

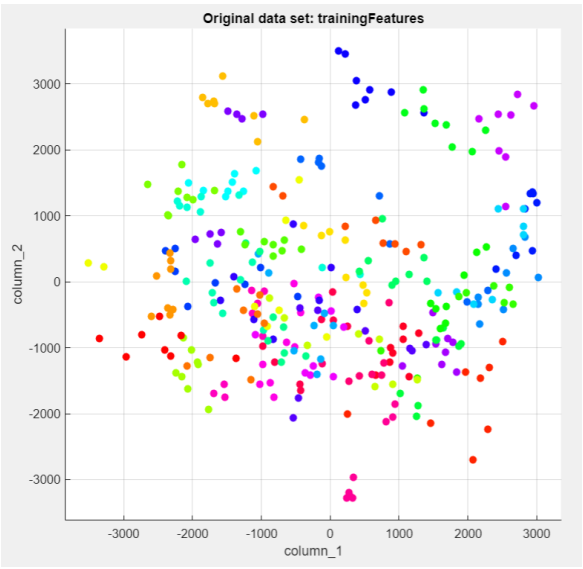


Confusion Matrix

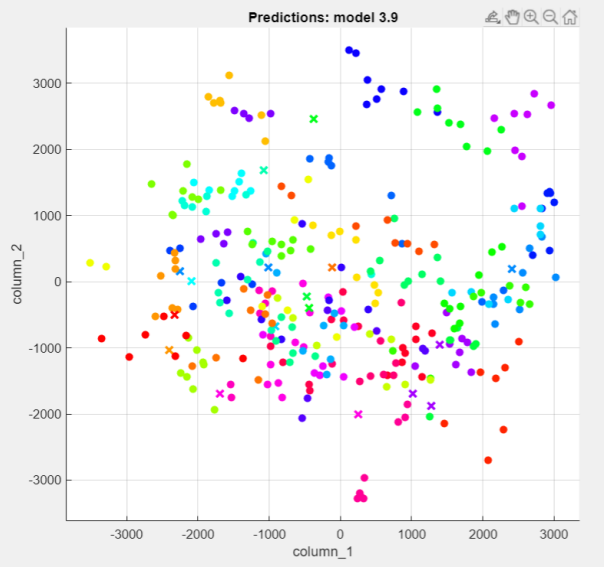
Figure 3.14 Scatter Plots and Confusion Matrix for Linear Discriminant model in EigenFace Approach

3.6.3.2 SVM

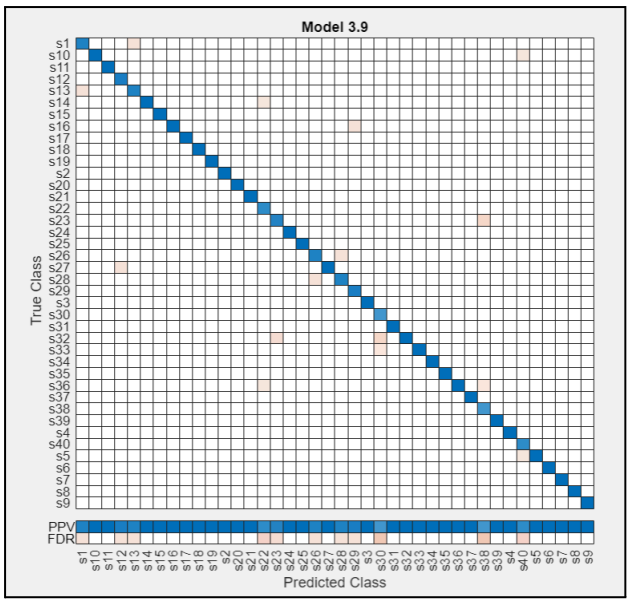
The support vector machine (SVM) is a predictive analysis data-classification algorithm that assigns new data elements to one of labeled categories.Figure 3.14 shows scatter plots and confusion matrix for SVM model in EigenFace Approach



Before Training



After training

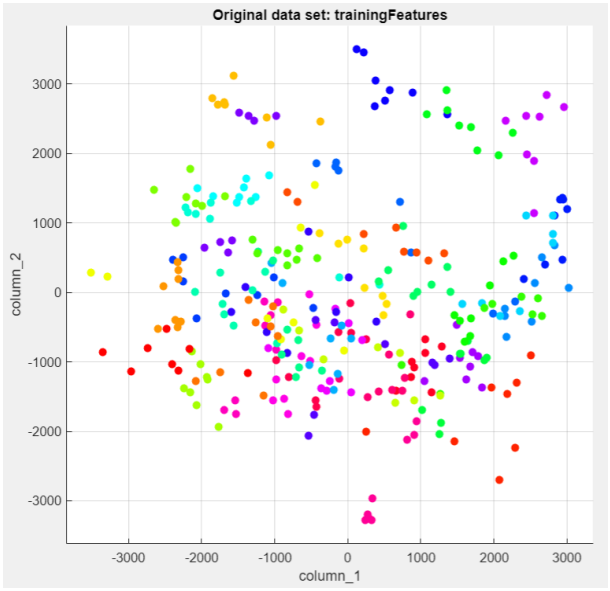


Confusion Matrix

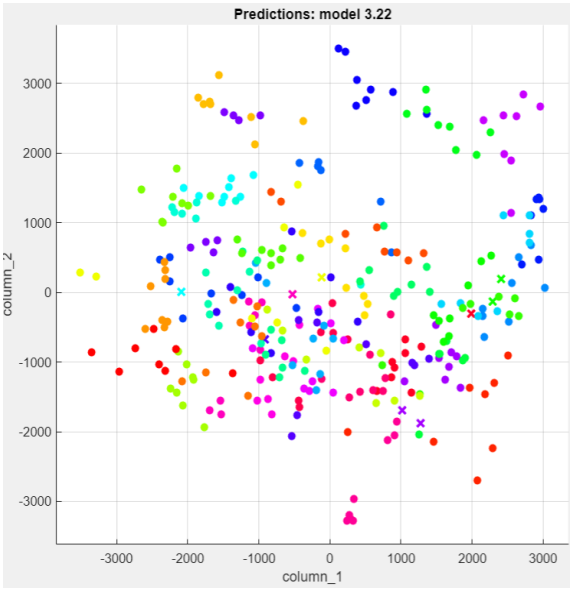
Figure 3.15 Scatter Plots and Confusion Matrix for SVM model in EigenFace Approach

3.6.3.3 Ensemble Learning

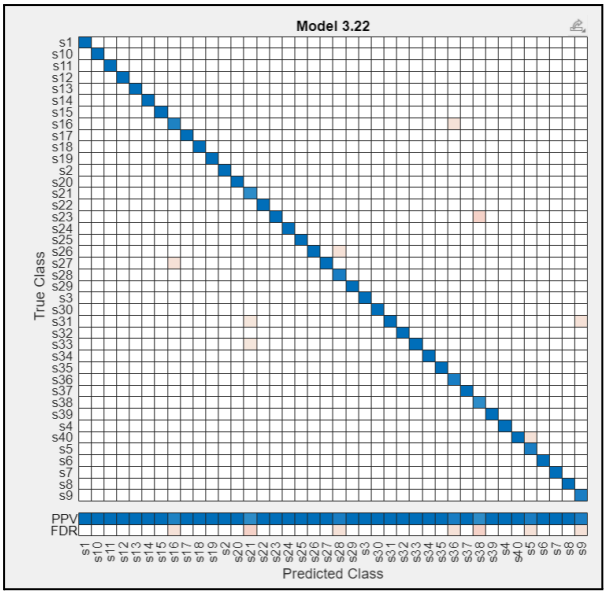
It refers to combining the predictions from two or more models. The goal of using ensemble methods is to improve the skill of predictions over that of any of the contributing members. Figure 3.15 shows scatter plots and confusion matrix for Ensemble Learning in EigenFace Approach



Before Training



After Training



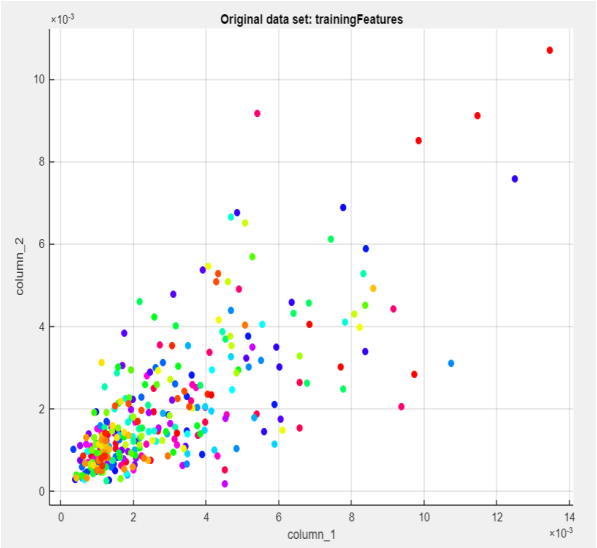
Confusion matrix

Figure 3.16 Scatter Plots and Confusion Matrix for Ensemble Learning in EigenFace Approach

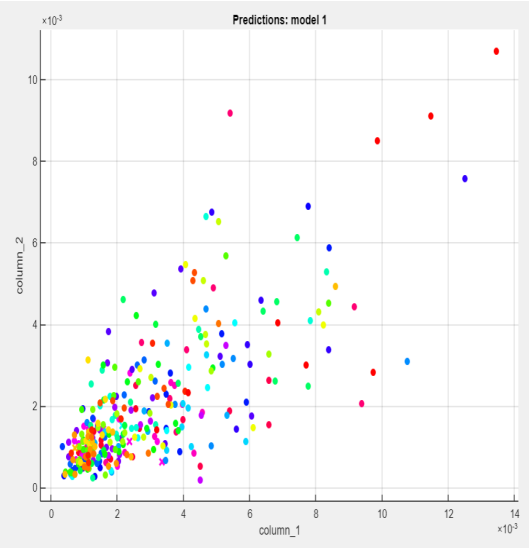
3.6.4 HOG

3.6.4.1 Linear Discriminant

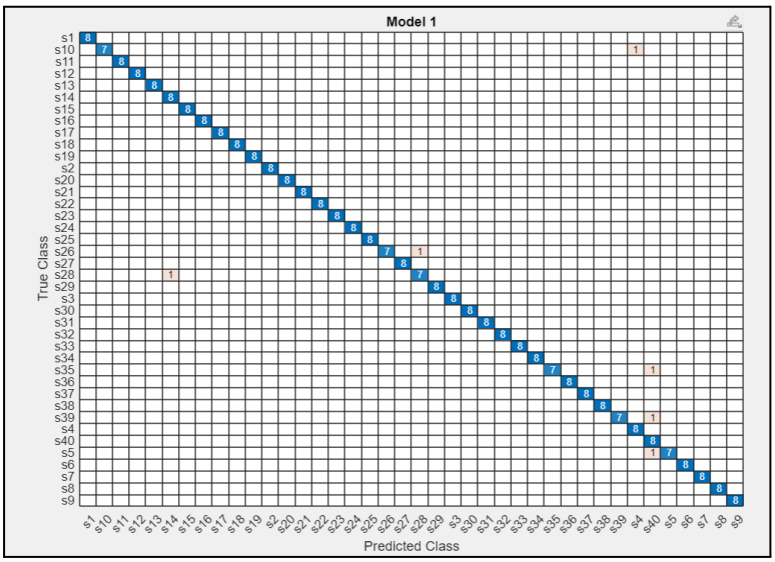
Figure 3.16 shows Scatter Plots and Confusion Matrix for Linear Discriminant model in HOG Approach



Before Training



After Training

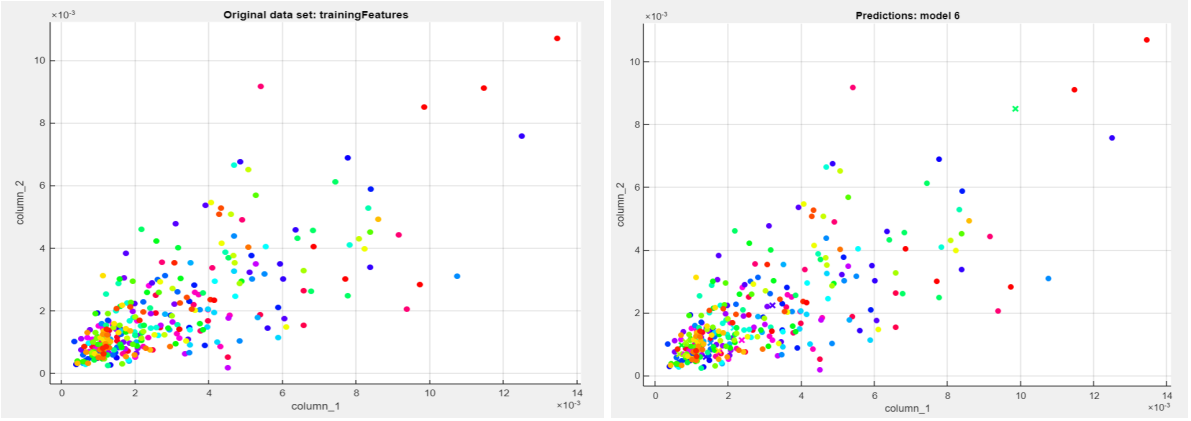


Confusion Matrix

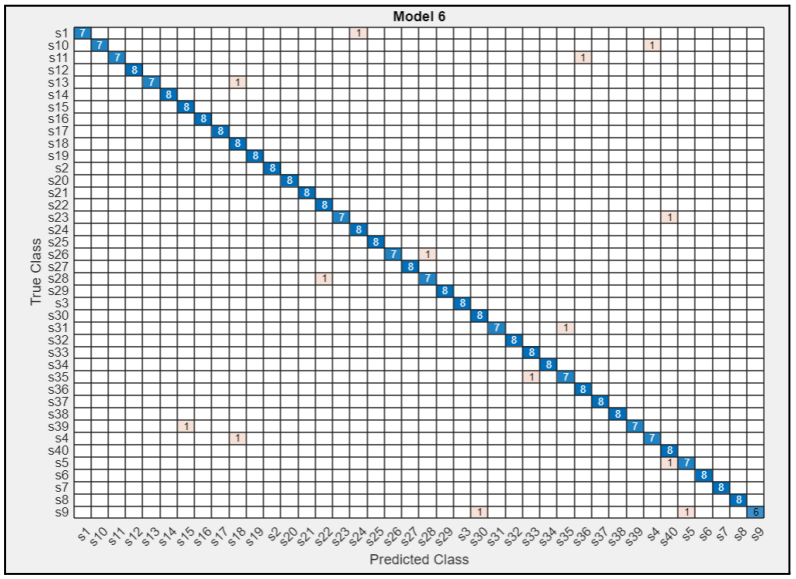
Figure 3.17 Scatter Plots and Confusion Matrix for Linear Discriminant model in HOG Approach

3.6.4.2 SVM

Figure 3.18 shows scatter plots and confusion matrix for SVM model in HOG Approach



Before Training **After training**

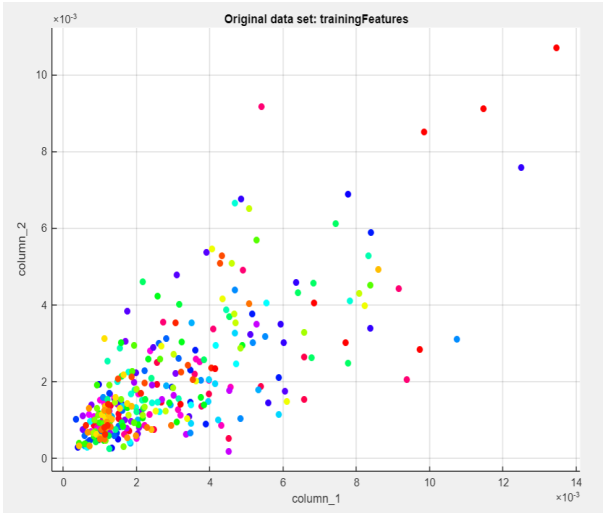


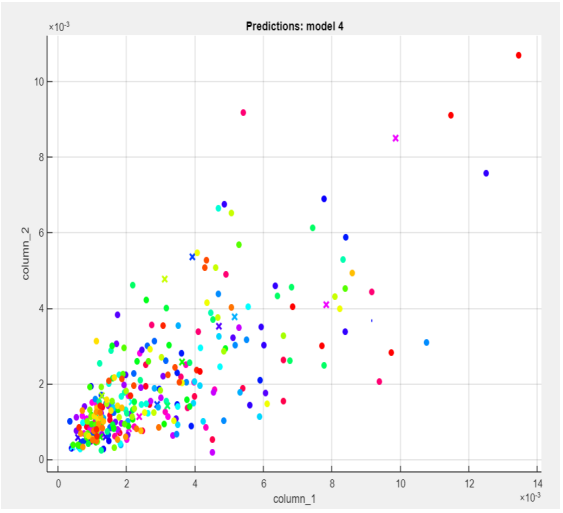
Confusion Matrix

Figure 3.18 Scatter Plots and Confusion Matrix for SVM model in HOG Approach

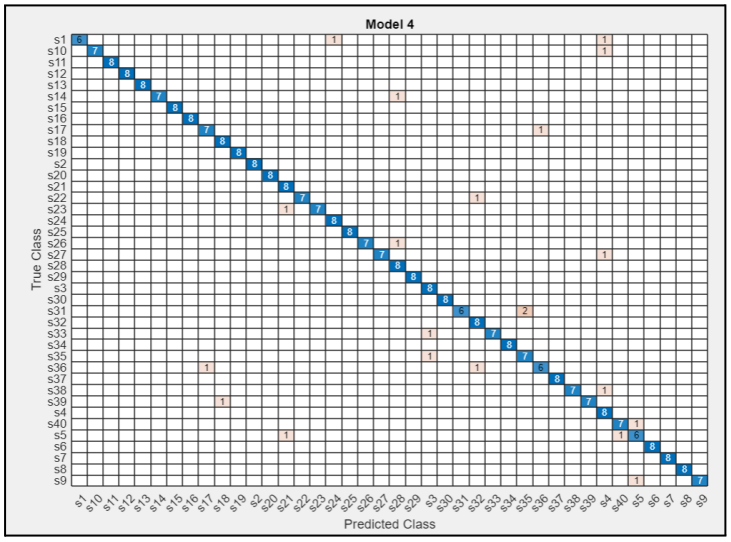
3.6.4.3 Ensemble Learning

Figure 3.19 shows scatter plots and confusion matrix for Ensemble Learning in HOG Approach.





Before Training **After Training**



Confusion Matrix

Figure 3.19 Scatter Plots and Confusion Matrix for Ensemble Learning in HOG Approach

CHAPTER 4: BLOCK DIAGRAM

A general block diagram of how the system works has been shown in the figure 4.1

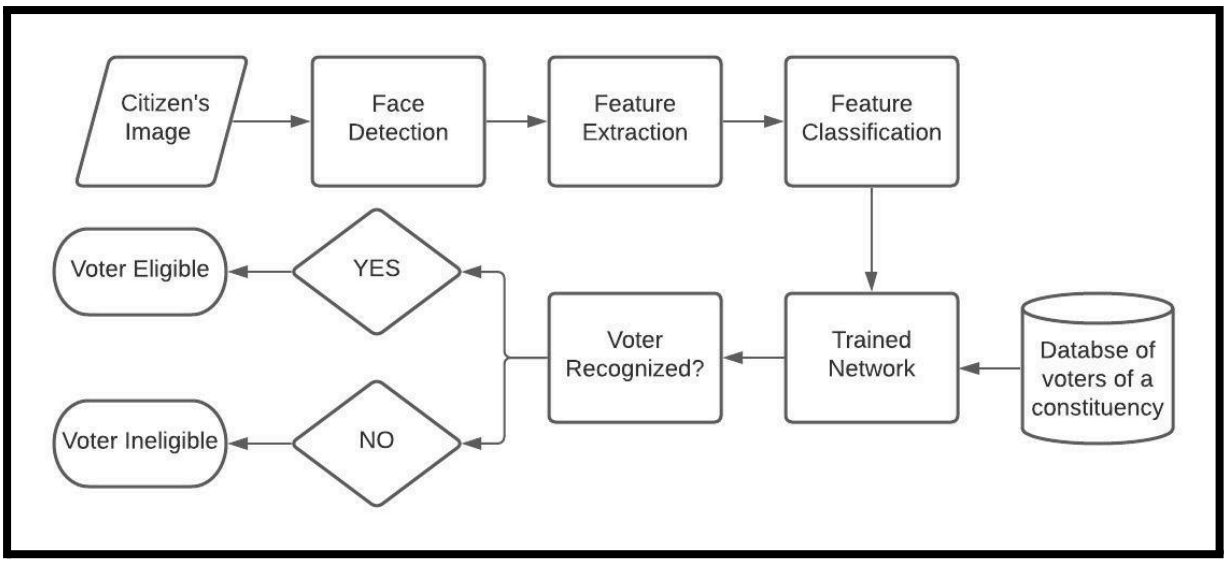


Figure 4.1 Block Diagram

The database consisting of the eligible voters are pre-processed. Pre-processing aims at improvement of image data by suppressing undesired distortions or enhancing some image features important for further preprocessing.The face is then detected using Viola Jones algorithm. The neural network will perform feature extraction and feature classification and training process takes place that is done using Eigenface and HOG transformation.Feature extraction is a process of dimensionality reduction by which an initial set of raw data is reduced to more manageable groups for processing. A characteristic of these large data sets is a large number of variables that require a lot of computing resources to process.Feature classification analyses various numerical properties of image features and organises data into categories. The Citizen's Image(Person X) is fed to the system through a webcam, is pre processed and then is sent for testing, where facial features are compared using PCA. If the Citizen's image is present in the database the system identifies it, thus allowing the person to cast a vote.

CHAPTER 5: SOFTWARE NEEDED

Our project involves use of MATLAB software.It is a programming platform designed specifically for engineers and scientists to analyze and design systems and products that transform our world. MATLAB, stands for Matrix Laboratory, a matrix-based language allowing the most natural expression of computational mathematics. An appropriate knowledge of MATLAB makes mathematical calculations much easier and helps professionals to solve numerical computations and perform Data Analysis and Visualization tasks.

It has tons and tons of toolboxes for us to use to simplify our code.A toolbox is a package of functions and/or classes. The Classification Learner app in the Statistics and Machine Learning toolbox lets us train classification models using supervised machine learning without writing any code.Classification Learner lets us perform common machine learning tasks, such as importing preprocessed data, exploring our data, quickly training multiple models, assessing model performance, and the iterative steps required to tune model performance, including optimizing hyperparameters, misclassification cost, and feature selection.

We are thinking of complementing our project with an app.GUIs helps a lot in interacting with one's project.

Graphical user interfaces (GUIs), also known as apps, provide point-and-click control of our software applications, eliminating the need for others to learn a language or type commands in order to run the application. We can share apps both for use within MATLAB and also as standalone desktop or web apps.

App designer is the recommended environment for building apps in MATLAB.App Designer lets us create professional apps without having to be a professional software developer.

Drag and drop visual components to lay out the design of our graphical user interface (GUI) and use the integrated editor to quickly program its behavior.

App Designer integrates the two primary tasks of app building – laying out the visual components of a graphical user interface (GUI) and programming app behavior.

CHAPTER 6 : RESULTS AND CONCLUSIONS

- The facedatabaseATT has 10 images each of 40 people in it. Each person has a separate folder for his image named from s1-s40. The montage of one image from each folder is as shown in figure 6.1.

Figure 6.1 Database Collection

- This below image shows a montage of all the images in a folder of one particular person. We have included 4 folders from s41-s44 including all our images which can be seen in figure 6.2

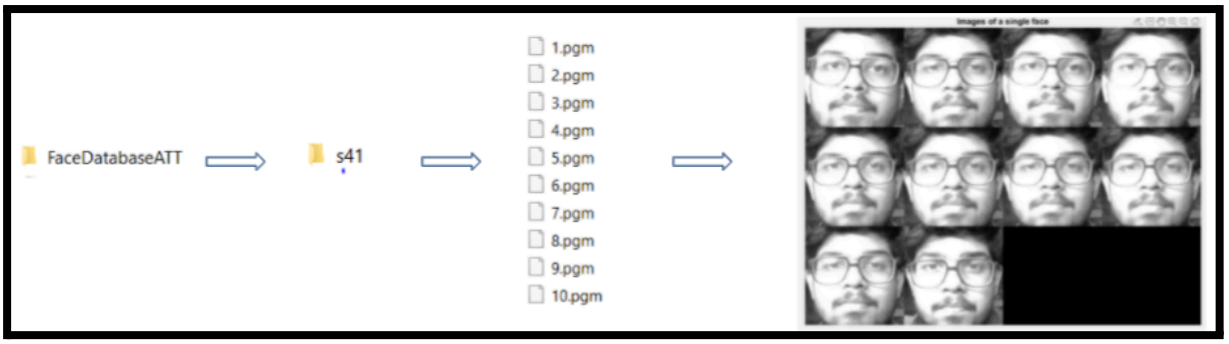


Figure 6.2 Database collection of team member

The below figure 6.3 is the comparison of the result obtained when a face is recognized using HOG features both by inbuilt function extractHOGFeatures and user defined function Hog_feature_vector.

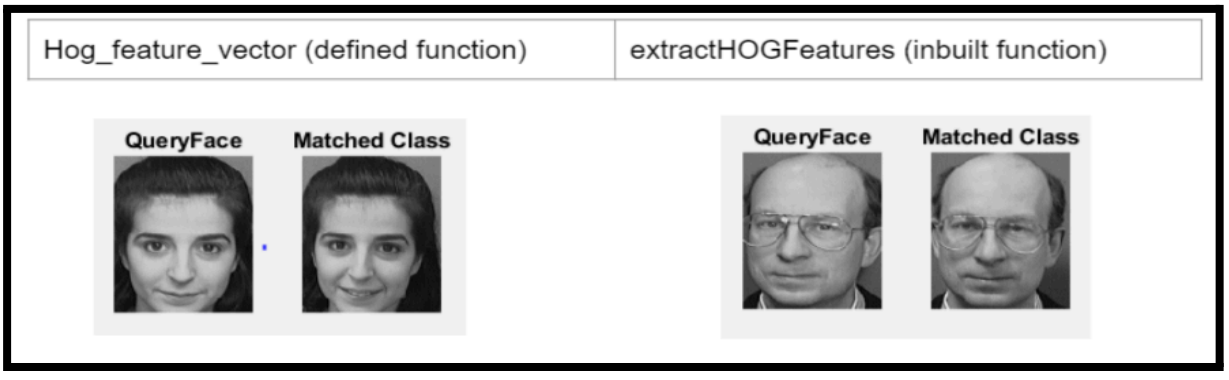


Figure 6.3 Results for HOG Approach

The below figure 6.4 is the comparison of the result obtained when a face is recognized using the Eigenface approach.

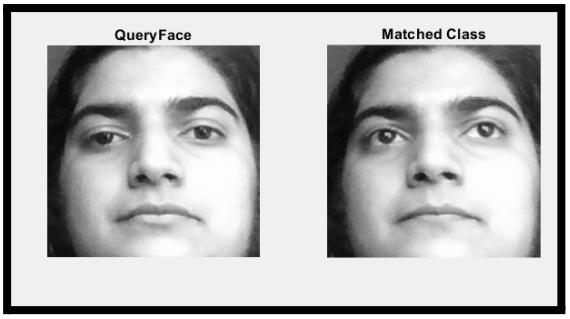


Figure 6.4 Results for EigenFace Approach

- The accuracy of each model (Linear Discriminant, SVM and Ensemble Learning) for both, Eigenfaces approach and Histogram of Oriented Gradients was tabulated as shown in tables 3.12.

  From the confusion matrix obtained for each of these, the classification error rate was calculated and tabulated as can be seen in the table 3.13.

  After analysis of the three models, we decided to go with the linear discriminant model with the HOG feature vectors approach as it was the best fit for the solution for the problem statement at hand with maximum accuracy of 98.1% and it also showed the least error rate of 0.014%

- The interface is in the process of being developed using the MATLAB GUI.Figure 6.7 and 6.8 are the GUI interface of Voter Recognition System

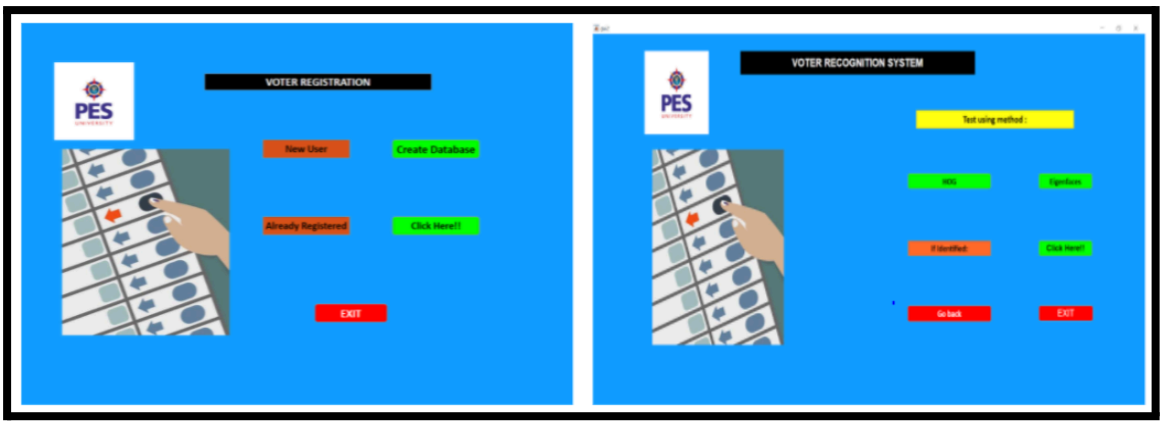


Figure 6.7 GUI interface of Voter Recognition System

Voter registration window helps us to add a new voter to our database. If already present he/she is taken to a next window, Voter recognition system, where the photo is tested and verified.Once verified the voter is sent to the window where he/she can cast their vote.If not verified there is a Go back option which takes us to the previous page. Once done, the Exit button is used to close all the running files.

Figure 6.8 GUI interface of Voter Recognition System

The above figure 7.8 is a demo electronic voting machine(EVM).The number of votes are counted for separate parties.Candidates can cast vote by pressing the party button.After all the votes has been voted.User can press the results button for viewing the the result of the election.In the window we can see the no of votes voted for a particular party,also the percentage of votes gained by the parties.Exit button closes all the current running windows.

REFERENCES

[1]M. Zulfiqar, F. Syed, M. J. Khan and K. Khurshid, "Deep Face Recognition for Biometric Authentication," *2019 International Conference on Electrical, Communication, and Computer Engineering (ICECCE)*, 2019, pp. 1-6, doi: 10.1109/ICECCE47252.2019.8940725.

[2]C. Molder and R. Oancea, "Appearance-based facial detection for recognition," *2012 9th International Conference on Communications (COMM)*, 2012, pp. 119-122, doi: 10.1109/ICComm.2012.6262545.

[3]J. Jung, S. Kim, C. Yoo, W. Park and S. Ko, "LBP-ferns-based feature extraction for robust facial recognition," in *IEEE Transactions on Consumer Electronics, vol. 62, no. 4, pp. 446-453, November 2016* , doi: 10.1109/TCE.2016.7838098.

[4]S. S. Thenuwara, C. Premachandra and S. Sumathipala, "Hybrid Approach to Face Recognition System using PCA & LDA in Border Control," *2019 National Information Technology Conference (NITC), 2019*, pp. 9-15, doi: 10.1109/NITC48475.2019.9114426.

[5]F. Perronnin and D. Larlus, "Fisher vectors meet Neural Networks: A hybrid classification architecture," *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015*, pp. 3743-3752, doi: 10.1109/CVPR.2015.7298998.

[6]H. S. Karthik and J. Manikandan, "Evaluation of relevance vector machine classifier for a real-time face recognition system," *2017 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia), 2017*, pp. 26-30, doi: 10.1109/ICCE-ASIA.2017.8307832.

https://towardsdatascience.com/the-intuition-behind-facial-detection-the-viola-jones-algorithm-29d9106b6999

https://www.analyticsvidhya.com/blog/2019/09/feature-engineering-images-introduction-hog-feature-descriptor/

geeksforgeeks.org/ml-face-recognition-using-eigenfaces-pca-algorithm/